Automated Mapping for Semantic-based Conversion of Transportation Data Formats[⋆][†]

Marjan Hosseini, Safia Kalwar, Matteo Rossi, and Mersedeh Sadeghi

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
{firstname.lastname}@polimi.it

Abstract. This position paper outlines our proposed approach to automate the process of creating mappings between different data formats in the transportation domain. The approach exploits the *word2vec* model, in combination with graphs for finding meaningful equivalent relationships between concepts in different data formats.

Keywords: Machine Learning · Ontology · Data Mapping.

1 Introduction

The modern vision of transportation is that of ”mobility as a service”, in which users can seamlessly build door-to-door trips including several travel modes through a single entry point, with a unified interface and payment methods. To realize this vision, a wide range of diverse actors of the transportation ecosystem must communicate, interact, and cooperate with one another. Divergence of transportation standards and heterogeneity of data representations, formats and models are the main obstacles towards making such an interoperable system a reality. Hence, solutions are needed that bridge this fragmentation, hide the peculiarities of different standards and allow for the communication and exchange of data among heterogeneous, non-integrated systems.

In line with this objective, The SPRINT (Semantics for PerfoRmant and scalable INteroperability of multimodal Transport) project aims at developing tools and technologies that facilitate interoperability in the transport domain. The core idea underlying the project is to go beyond pure “syntactic” interoperability—where interested parties are forced to adopt a unified set of formats for data exchange—and instead leverage “semantic” interoperability, which enables different systems to communicate with each other through their native standards, by mapping their concepts to a common ontology, which provides an unambiguous and homogeneous view of data. One of the specific goals of the SPRINT project is to enhance and automate the conversion process realized

[⋆] This work was supported by Shift2Rail and the EU Horizon 2020 research and innovation programme under grant agreement No: 826172 (SPRINT).

by the ST4RT Converter [2], which is the main component for the realization of semantic interoperability among heterogeneous, legacy transport services.

A ST4RT converter (whose main principles and processes are depicted in Fig. 1) is a software artefact that acts as an adapter between two distinct formats. Given a suitable mapping between the source/target data and a reference formal ontology, a ST4RT converter first transforms data expressed in the source format into an intermediate representation based on the reference ontology. Then, following a similar procedure in the reverse direction, the converter translates the intermediate representation into the target data model. This approach has the notable advantage of exempting participating parties from harmonizing the syntax and structure of their data; a meaningful communication is achieved only if they agree on the concepts and semantics behind their terminology and syntax. As shown in Fig. 1(a), the ST4RT Converter relies on dual *lifting* and *lowering* processes mapping given standards to/from the reference ontology. The first step in whole approach is the annotation process, through which the source and target standards are semantically annotated to state the mappings between their data models and the reference ontology. Figure 1(b) shows the whole conversion workflows at design-time (left) and run-time (right). At design-time, given the structured data model of a standard, corresponding java classes are automatically generated, which are the basis for *annotation* process. At this stage, java classes, attributes and methods must be annotated to map each term to its equivalent concept in the reference ontology [2]. Using the annotated java classes as input resources, the conversion process happens at run-time when the system receives a message that is an instance of the source standard. The converter decomposes the source message into the concepts and terms according to its native standard and creates the instances of the corresponding java classes for each concept. Finally, it uses the defined mapping to lift such java instances to RDF triples conforming to the reference ontology. Following the inverse direction in the lowering stage, the converter first uses the defined mapping to translate the RDF triples into instances of suitable java classes representing concepts of the target standard, and it ultimately generates the converted message that is

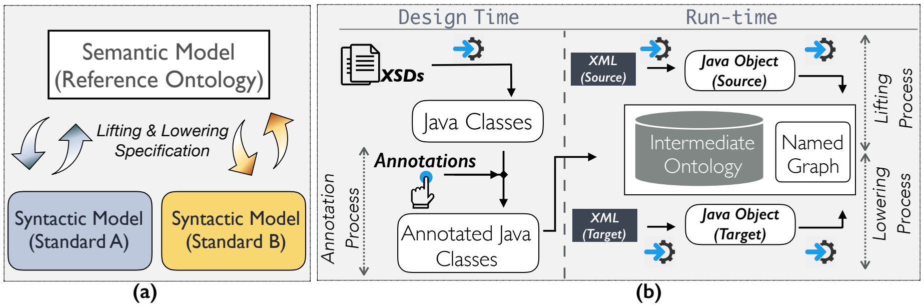

Fig. 1. a) ST4RT approach to semantic interoperability, b) Converter Workflow at design- and run-time, composed of annotation, lifting and lowering processes.

an instance of the target data model. Except for the mapping process in the annotation phase, all steps are accomplished in an automated manner.

This position paper presents our proposed approach to make the *annotation* phase of the conversion process more efficient, automated, and user-friendly. As mentioned above, so far the annotation step is carried out manually, which hampers the efficiency and overall performance of the process. Human users who are expert in both the reference ontology and the desired standards are required to establish the mappings with the concepts appearing in the source and target specifications, which is a time- and effort-consuming procedure. In the proposed approach, we aim at making the annotation-creation process more automated by taking advantage of machine-learning algorithms and methodologies.

The rest of this paper briefly describes related works (Sect. 2), then outlines the proposed method (Sect. 3), and concludes with a brief discussion (Sect. 4).

2 Related Works

The *word2vec* model [3] is a 2-layer neural network which can be trained using a sufficient amount of text corpus as the input, and which outputs the feature vectors of the words appearing in the input corpus so that the vectors of semantically similar words are mapped near one another in the vector space. These vectors can be employed to establish meaningful associations among the words (e.g., Milan is to Italy what Paris is to France). The produced vectors can also be used as the input to other machine learning techniques, such as clustering or extra deep neural networks. Another property of the *word2vec* model [4] is the capability of meaningfully combine words' vectors and represent longer texts by performing operations such as addition or subtraction. The *word2vec* model has already been employed in the medical domain for concept extraction [1].

3 Method

As explained in Section 1, in order to *lift* /*lower* a given standard to/from the reference semantic model we need to state the mapping between their concepts and structures in the *annotation* phase. This section describes the proposed method for the automatic generation of such mappings.

Definitions Let S and R be, respectively, the source standard and the reference semantic model. We indicate by $\mathcal{X}_S$ (resp., $\mathcal{X}_R$) the structure used by S (resp., R), and by $\mathcal{O}_S$ (resp., $\mathcal{O}_R$) the set of the vocabularies in $\mathcal{X}_S$ (resp., $\mathcal{X}_R$). For example, $\mathcal{X}_S$ could be defined through XSD or OWL. If $\mathcal{M}_S$ is an instance of $\mathcal{X}_S$ and $\mathcal{M}_R$ is an instance of $\mathcal{X}_R$, we say that $\mathcal{M}_S$ and $\mathcal{M}_R$ are equivalent if they are used for the same purpose in S and R—i.e., they are semantically equal.

We consider instances $\mathcal{M}_i$ (where $i \in \{S, R\}$) in which we can identify a *root concept*. Then, $\mathcal{M}_i$ is defined using a sub-tree $W_{\mathcal{X}_i}$ of $\mathcal{X}_i$, which is based on a vocabulary $V_{\mathcal{O}_i}$ that is a subset of $\mathcal{O}_i$ (i.e., $V_{\mathcal{O}_i} \subseteq \mathcal{O}_i$). We write

$$\mathcal{M}_i \in T(V_{\mathcal{O}_i}, W_{\mathcal{X}_i})$$

to indicate that $\mathcal{M}_i$ is built on sub-tree $W_{\mathcal{X}_i}$ using the terms of $V_{\mathcal{O}_i}$.

Given $\mathcal{M}_s$, $V_{\mathcal{O}_S}$, $W_{\mathcal{X}_S}$, $\mathcal{O}_R$ and $\mathcal{X}_R$, we aim to define a *Map* method that maps the concepts appearing in $\mathcal{M}_S$ to concepts of $\mathcal{X}_R$, thus building an instance $\mathcal{M}_R$ of R that is equivalent to $\mathcal{M}_S$:

$$\mathcal{M}_R = \mathit{Map}(\mathcal{M}_S, V_{\mathcal{O}_S}, W_{\mathcal{X}_S}, \mathcal{O}_R, \mathcal{X}_R)$$

Assumptions For applying the proposed method, we assume that the following four premises are true. Although some of them might not be true in general, our aim is to automate the mapping process in most cases, sacrificing completeness to obtain efficiency, thus we deem these simplifications to be acceptable and general enough. We briefly discuss in Sect. 4 how the last one can be relaxed.

Assumption 1 The language in both sides of the mapping is English.

Assumption 2 Given that our method targets mappings between standards in the same domain and both involved systems cover the concepts of the domain, we assume that for each concept in the source system, we have at least one corresponding concept in the reference system.

Assumption 3 The corresponding instances in the source (i.e., the given standard) and target (i.e., the reference semantic model) formats include the same equivalent concepts; that is, for each concept in the source instance, there exists exactly one concept in target format (one to one relationship between concepts).

Assumption 4 All concepts exist in the *word2vec* model.

Procedure Figure 2 depicts an overall workflow of the proposed procedure. In order to map the source data to the reference data format, the first step is to decompose the source data to its components: $V_{\mathcal{O}_S}$, which is the set of terms that exist in $\mathcal{M}_S$, and the tree representation $W_{\mathcal{X}_S}$ of the given structured data. Then, the semantic equivalent of the *main concept* in the source data should be determined, where the main concept is the root of the tree. According to Assumption 2, there should be at least one concept in the reference data model corresponding to the source main concept. To detect it, the extracted root of the tree structure in the source data ($W_{\mathcal{X}_S}$), would be embedded to its 300-dimensional vector space employing the *word2vec* model. If the main concept is a phrase, its atomic parts should be embedded individually and then averaged. Since the *word2vec* model identifies semantically close concepts based on their relative distances in the 300-dimensional vector space, we search the space for the vector that is nearest to the one of the source main concept and tag it as the equivalent concept in the reference system.

After determining the equivalent main concepts, the structures corresponding to that particular main concept in the source and reference systems ($W_{\mathcal{X}_S}$ and $W_{\mathcal{X}_R}$) are retrieved. Then, inside the tree structures of each data format, all the

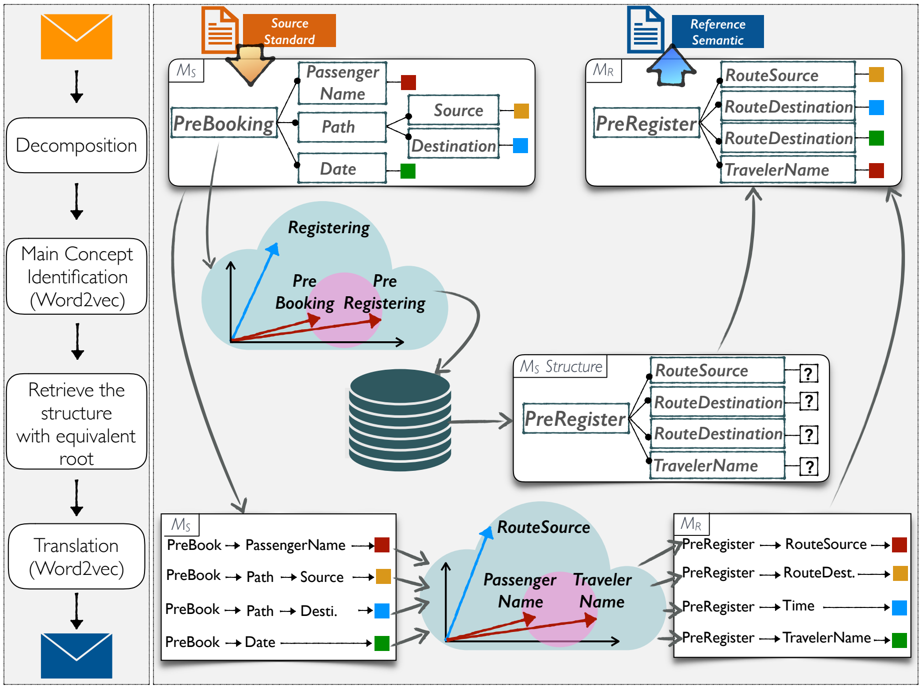

Fig. 2. Proposed mapping procedure from source to reference data format.

possible routes from the root to the leaf nodes are extracted. To reach a single leaf node, there is only one path from the root. Each extracted route consists of the set of all the terms from the root to that particular leaf node. According to Assumption 3,[1] the structures of the equivalent main concepts in the source and reference data contain the same number of attributes, hence the number of leaf nodes in both data formats are equal, and so is the number of routes.

For each route in the tree structure of the source data, its corresponding route in the reference data structure should be identified. To do so, a vector for each of the words in the current route is obtained using the *word2vec* model, and then the average of such vectors must be calculated. We expect that the vectors that are the result of the averaging operation in the corresponding routes in the source and reference data fall closer to each other in the vector space. This is possible due to the properties of the *word2vec* model, in which combinations of words can be meaningfully represented through vector addition. Similarly, the average operation of the vectors preserves their semantics, since it just scales the vector magnitude by a positive number, leaving the direction unchanged. We assign the route from the source data structure to the nearest average of the existing routes in the reference data structure. Then, the attribute name of the

[1] As mentioned in Sect 3, we trade-off generality for efficiency. In future works we will look to relax the assumptions underlying the approach.

leaf node from the source data is mapped to the corresponding leaf node in the reference data format and their attribute values are transferred accordingly.

4 Discussion

This section outlines some of the possible challenges that might arise while applying the proposed method. The first one is related to the extracted concepts and occurs because the individual concepts in the structure are typically a combination of two or more words, for example *pre_booking* or *PreBooking*. These kinds of compound words usually do not exist in the *word2vec* model as a single word. To address this issue, it might be necessary to perform a pre-processing step, which could possibly be splitting the compound words to their atomic stems and then computing their average.

Another challenge could be due to the absence of some terms of the source or reference ontologies, which prevents the averaging in the route matching step. To tackle this problem, one possible approach is further training the existing *word2vec* model, using the transfer learning technique. To this end, a sufficient number of unstructured texts containing the missing words are necessary. Although, instead of unstructured text, it might be possible to perform transfer learning using instances of structured data formats containing the missing words, either by flattening the structured text to make it unstructured, or by adding extra layers to the *word2vec* model.

Finally, to validate the method, a possible approach could consist in preparing a dataset containing a set of pairs $(\mathcal{M}_1, \mathcal{M}_2)$ of equivalent instances in different data formats (hence, with the same main concepts). Then, the *Map* method should be applied to one element from each pair (say, $\mathcal{M}_1$) and the result should be compared to the true data structure and the terms of the other element of the pair (i.e., $\mathcal{M}_2$). Subsequently, the direction of the *Map* method should be reversed (i.e., it should be applied to $\mathcal{M}_2$) and the same process should be repeated. The method is validated if both mapped data are equivalent to the corresponding true data formats.

References

1. Andrew L Beam, Benjamin Kompa, Inbar Fried, Nathan P Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. *arXiv preprint arXiv:1804.01486*, 2018.
2. Alessio Carenini, Ugo DellArciprete, Stefanos Gogos, Mohammad Mehdi Purhashem Khallehbasti, Matteo Rossi, and Riccardo Santoro. ST4RT – semantic transformations for rail transportation. In *Transport Research Arena (TRA)*, pages 1–10, 2018.
3. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
4. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.