# Introduction

Over the past couple of decades there has been an exponential increase in the amount of data that is generated by the web and more recently the internet of things (IOT). This exponential increase has not always been accompanied by a similar increase in the extraction of information and knowledge from this data. The lag in information gain is mainly due to the large volume and unknown structure of the data that makes their analysis quite a difficult and complex task [1]. To overcome this issue, it is essential to develop the necessary tools that will help understand and manage the content of the data. Those tools will allow the correct classification of the data and will incorporate data transformation, cleansing and standardisation as part of pre-processing steps.

On top of data mining and knowledge retrieval there are many cases where different services offered by different applications need to be integrated or data need to be transferred from one system to another (e.g. system migrations). In those cases, it is important to have an interface agreement whereby data from interacting systems need to be well curated and mapped. Oftentimes this is a cumbersome and manual task especially when no additional documentation or system specs are provided.

Within the space described above, the research question we will try to answer as part of this project is the following:

*How can we enhance a set of data given as input (e.g. tabular data) with semantic meaning using existing knowledge graphs (e.g. DBpedia, WikiData) as reference?*

This project is inspired by the Sem Tab challenge that has been organised annually since 2019. The scope of the challenge is organised in 3 separate but overlapping tasks listed below:

* CTA Task: Assign a class from a KG to an entire column of a table
* CEA Task: Assign an individual entity of a KG to each specific cell
* CPA Task: Assign the relationship (i.e. object property) between 2 table columns

The above tasks are reflected in Figure 1

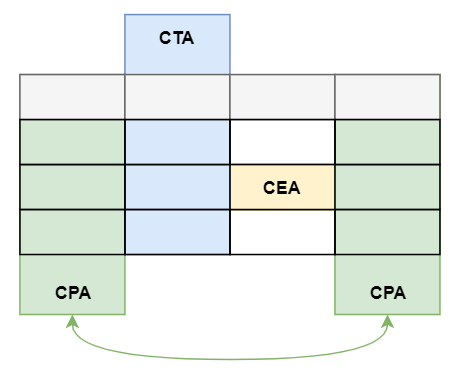


Figure 1. Matching tabular data to classes (CTA), entities (CEA) and properties (CPA)

# Critical Context

As mentioned in the previous section, the scope of this project has been inspired by the Semantic Web Challenge on Tabular Data to Knowledge Graph matching (SemTab) challenge that was first introduced in 2019 [2]. This challenge aims to create a framework that would systematically evaluate proposed matching systems by automatically generating datasets of increasing complexity and suggesting metrics to assess the accuracy of the submitted solutions.

To generate the benchmark dataset SemTab 2019 used DBpedia as the reference knowledge graph (and in fact its English annotations) however the approach is generic enough that the KG can be replaced by any other (e.g. Wikidata or other domain specific KG). The dataset creation methodology is summarised below:

* **Profiling**: Extracts a list of classes and properties of each class as well as a number of class instances that have the properties populated. The datatype and range is also extracted for the data and object properties respectively
* **Raw Table Generation**: Each identified class is extracted as a table with 3-7 columns randomly selected from the class properties. The process ignores tables with less than 5 rows. Moreover, to avoid large classes overwhelming the dataset there an upper limit (2000) rows that are randomly selected for each class. The process ensures that all values extracted are of the same data type and if multiple values exist of a property, only one of them is selected
* **Refinement**: With the dataset at hand this next step ensures that some noise is introduced to make the data more realistic and representative of a real life use case. This is done by identifying tables that more challenging to match and even within those identifies challenging rows and removes the rest from the dataset.
* **RDF Data**: Finally the process if converting the data to RDF format using column headers to identify class names and object properties while additionally the rdf:label predicate is used to keep the literal values of the cells in the dataset. As implied for the dataset to be converted to RDF it is necessary that tables have a header row.

To evaluate the proposed solutions the challenge used the traditional precision, recall and F1 scores for the CEA and CPA task whereas for the CTA task two different measures where used Average Hierarchical Score and Average Perfect Score in order to take into account the taxonomy of the classes.

In the following sections we present some of the proposed solutions for the SemTab challenge that will form the basis of our implementation

## MTab

MTab [2] is the top performing proposal out of all participants in the 2019 challenge across all three tasks.

Prior to dealing with the tasks as hand the proposed system is performing some preprocessing step to clean the data and extract some metadata for the given tables. In summary the preprocessing is attempting to rectify incorrect encoding predict the language of the table cell values as well as metadata on their data types and entity type from another KG (OntoNotes 5) which is also manually matched to DBpedia. Finally, the system performs a lookup directly to DBpedia or indirectly through redirected links from Wikidata to retrieve a list of candidate entities considering the language parameters identified in on of the previous steps.

Following the prepossessing there is a two phase approach whereby:

* in the first phase the system estimates the candidate according to relevance for each of the 3 tasks (entity, type and relation) and
* in the second phase those candidates are refined to come up with the final output of entity, property and class respectively. For entity re-estimation the algorithm is calculating the probability by combining the probability of the candidates from the first phase (entity, type and relation) with certain weights. Finally it using the entity with the highest probability to define the property and class.

MTab [24]. MTab is a system that can jointly deal with the three tasks CTA,

CEA and CPA. It is based on the joint probability distribution of multiple

table to KG matching, following the probabilistic graph model by [20]. However,

the team improves the matching by using multiple services including DBpedia

Lookup, DBpedia endpoint, Wikipedia and Wikidata, as well as a cross-lingual

matching strategy.

# Approaches: Methods & Tools for Design, Analysis & Evaluation

# Work Plan

# Risks

# Ref

[1] A. Lausch, A. Schmidt, L. Tischendorf, “Data mining and linked open data – New perspectives for data analysis in environmental research”, Ecological Modelling 295, 2015, p. 5-17 [https://doi.org/10.1016/j.ecolmodel.2014.09.018](https://0-doi-org.wam.city.ac.uk/10.1016/j.ecolmodel.2014.09.018)  
  
[2] Jimenez-Ruiz, E. , Hassanzadeh, O., Efthymiou, V., Chen, J. and Srinivas, K. (2020). SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In: The Semantic Web. ESWC 2020. Lecture Notes in Computer Science. (pp. 514-530).

[3] Nguyen, P., Kertkeidkachorn, N., Ichise, R., Takeda, H.: MTab: Matching Tabular Data to Knowledge Graph using Probability Models. SemTab, ISWC Challenge (2019)

F. Kalloubi, E. H Nfaoui, O. El Beqqali, “Micro blog semantic context retrieval system based on linked open data and graph-based theory”, Expert Systems With applications 53, 2016, p. 138-148

[10.1109/SITA.2014.6847286](https://0-doi-org.wam.city.ac.uk/10.1109/SITA.2014.6847286)

T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, E. Di Sciascio, R. Mirizzi, C. Bartolini, “Building a relatedness graph from Linked Open Data: A case study in the IT domain”, Expert Systems With Applications 44, 2016, p. 354-366

[https://doi.org/10.1016/j.eswa.2015.08.038](https://0-doi-org.wam.city.ac.uk/10.1016/j.eswa.2015.08.038)

The diagram in Figure 1 illustrates the steps taken for the first 3 tasks of the coursework (i.e. to create an ontology based of the give data, load the data to a KG and extract useful insights).

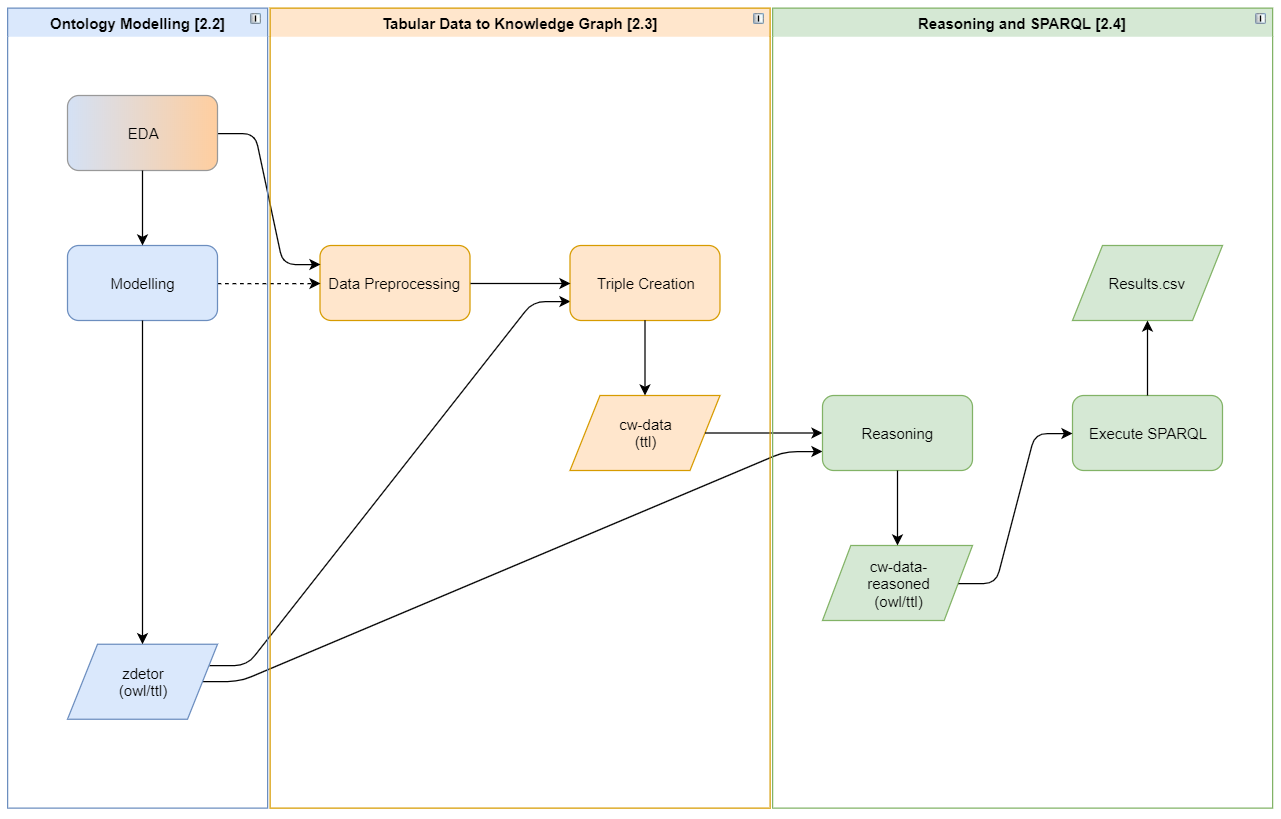


Figure 1. Process of modelling and converting tabular data to a KG