Semantic Web 1 (0) 1–5
IOS Press

Typology-based Semantic Labeling of Numeric Tabular Data

Ahmad Alobaid [a], Emilia Kacprzak [b] and Oscar Corcho [a]
[a] *Department of Artificial Intelligence, Universidad Politécnica de Madrid, 28223, Madrid, Spain*
E-mails: aalobaid@fi.upm.es, ocorcho@fi.upm.es
[b] *Department of Electronics and Computer Science, University of Southampton, UK*
E-mail: e.kacprzak@soton.ac.uk

Abstract. More than 150 Million tabular datasets can be found on the Google Crawl of the Web. Semantic labeling of these datasets may help in their understanding and exploration. However, many challenges need to be addressed to do this automatically. With numbers, it can be even harder due to the possible difference in measurement accuracy, rounding errors, and even the frequency of their appearance (if treated as literals).

Multiple approaches have been proposed in the literature to tackle the problem of semantic labeling of numeric values in existing tabular datasets, but they also suffer from several shortcomings: closely coupled with entity-linking, rely on table context, need to profile the knowledge graph and the prerequisite of manual training of the model. Above all, they all treat different kinds of numeric values evenly. In this paper, we tackle these problems and validate our hypothesis: whether treating different kinds of numeric columns differently yields a better solution.

Keywords: Semantic Labeling, Semantic Annotation, Levels of Measurements, Typology of Numbers, Fuzzy Clustering, Semantic Web

1. Introduction

The number of structured data published on the web is constantly growing. With initiatives such as the Open Data movement (e.g., data.gov[1] in the US or European Data Portal[2] in Europe), machine learning platforms (e.g., Kaggle[3]), platforms with scientific data (e.g. Figshare[4], Zenodo[5]) or social platforms (e.g., data.world[6]), the need and potential opportunities for approaches facilitating the population of these datasets is greater than ever.

One of the most fundamental problems is enabling the understanding of the dataset content which could be beneficial for different groups, for example, researchers working on the expansion of Knowledge Bases (KB), such as DBpedia or Wikidata [1], for a data scientist who found a dataset and wants to use it for his knowledge discovery task [2], or for expanding metadata describing a dataset which could be later further utilised for data search (e.g., Google Dataset Search Engine [3]).

There are a number of approaches focusing on the problem of understanding the semantic meaning of the content of datasets. They focus on detecting semantic labels for specific dataset cells (such as, [4]) or assigning semantic labels to the whole column (such as, [5, 6]). Recently, the problem of assigning semantic labels to numerical columns in a dataset started to get traction as many of the previous works focused mainly on textual columns with not much attention being drawn to numerical columns present in the dataset. In the work of Mitlöhnert et al. [7], it was pointed out that numerical columns are the most popular column type among datasets from a number of open data portals.

[1] https://www.data.gov/
[2] https://www.europeandataportal.eu/
[3] https://www.kaggle.com/
[4] https://figshare.com/
[5] https://data.mendeley.com/
[6] https://data.world/

1570-0844/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved

The approaches targeting the annotation of numerical columns need to tackle different issues than the ones developed for textual columns; it can be harder due to the possible difference in measurement accuracy, rounding errors, the frequency of their appearance (if treated as literals) or different values being reported by different resources (e.g., population data). Multiple approaches tried to solve this problem, but they also suffer from several shortcomings: closely coupled with entity-linking (each number in the numeric columns has to match a numeric property of an entity in the knowledge graph), rely on table context (e.g., the URL of the page, the caption of the table, the surrounding text), need to profile the knowledge graph (e.g., create a model of the whole knowledge graph, build inverted index), and the prerequisite of manual training of the model. Above all, they all treat different types of numeric data evenly.

In this work, we expand our recent work [8], where the semantic labeling of numerical values is performed using fuzzy clustering. Our previous work shows that it is possible to semantically annotate numerical columns in tabular datasets without applying object matching or entity linking techniques. It also shows a limitation: while it works well if the majority of the data points for a column/property are mostly near the center of the data (have a distribution close to normal), it does not perform well when the data points are scattered further away from the center. In this work, we expand on this work with an approach looking at different types of numerical column in a more tailored manner following the hypothesis: that treating types of numeric columns differently yields better results than treating them uniformly. We propose a typology of numbers based on the typology proposed by others [9–11]; taking into account the purpose of semantic labeling. We take different approaches to annotate each of them.

The contributions of this paper can be summarised in the following points:

- We introduce a typology of numerical values taking into account the semantic labeling task.
- We propose a way to detect the type given numerical values based solely on its content, without external resources or context.
- We propose a new approach to label numerical columns in tabular data taking into account the type of numerical values in a column.

The rest of the paper is organised as follows. We begin by reviewing the typology of numbers in Section 2. We review the space of semantic labeling and break down the problem in Section 3. We present the typology of numerical values and how we detect each type in Section 4. In Section 5 we show how we extract data from a knowledge graph and construct the model which we use for the semantic labeling in Section 6. We evaluate our approach in Section 7 and conclude the paper in Section 8.

2. Background: Types of Numbers

Mosteller and Tukey said: "Just writing in digits does not make a number"[10]. On the other hand, Stevens and Birkhoff [9] typology of numbers doesn't quite agree with this[7]. They propose four kinds of numbers: nominal, ordinal, interval, and ratio [8].

Nominal where numbers are just like text, so they don't have any significant meaning beside differentiating them from other numbers (or texts). An example of that is the numbers printed on the football player shirts: it only distinguishes the players, but it doesn't really have a meaning like bigger or smaller. Such numbers represent classes, and the number of occurrences (or members) of a given class is used. This is not the same as counting the occurrences of a given class.

Ordinal kind is when the order matters. It is similar to the nominal, but with the numbers having the order, i.e. a number X is less than a number Y with no regard to the actual value or the difference between the two. An example would be ordering the dishes in a menu from the most favorable to the least favorable, or differences in responses in satisfaction surveys between "very satisfied" and "somewhat satisfied" [12].

Interval scale is used to denote the increase or expansion in some way on a scale. A classical example presented by [9] is the use of two different scales to measure the temperature. We (humans) use Centigrade and Fahrenheit, and we have a way to transform the temperature from one scale to the other. The intervals are the same (e.g., the difference between 10 and 20 degrees Celsius is the same as the difference between 25 and 35 degrees Celsius) [12].

Ratio scales refer to what we use in our daily life: numbers like how many eggs in a carton or the height of a table. It has all the above properties. For example, we can transform the height of a table from meters to

[7] The paper was written by Stevens but Birkhoff helped complete the list of types.

[8] They are not mutually exclusive. As Stevens mentioned in the paper, each type include the following one [9]

centimeters by multiplying it by 100. The main difference is the existence of an absolute zero. For example, when we talk about the temperature, there can be an agreement on a zero value (For example 0 in Celsius and 32 in Fahrenheit for the freezing point of water at 1 atmosphere unit), but it is not a true zero point (it does not mean a complete absence of temperature). Such a point is referred to as the "starting" point [10, 12].

There are measures that are not considered by Stevens like cyclical measures [11]. An example of such cyclical measures is the angle; Chrisman said: "the direction 359° is as far from 0° as 1° is" [11]. The added levels presented by Chrisman are: Log-interval, extensive ratio, cyclic ratio, derived ratio, Counts, and Absolute. Log-interval is similar to the interval scale with the scaling happening on the exponent level. Extensive ratio is the same as the fundamental ratio explained by Stevens (e.g., the width of a book). Cyclic ratio is the same as the extensive ratio where it is bounded and repeat. Derived ratio is the mathematical function of the magnitudes of a simple ratio measure. Counts measure are also included within the ratio scale by Steven. It is to measure the how many occurrences or number of objects. Absolute is when a scale is predetermined and can't be transformed while preserving the meaning.

Mosteller and Tukey have a similar categorization: names (nominal), grades and ranks[9] (ordinal), amounts (fundamental ratio), balances[10] (derived ratio), counts, counted fractions[11] (absolute). But it lacks the followings: interval[12], log-interval, and cyclical ratio.

Types of numbers are also referred to as kinds of numbers, scales of measurement, and levels of measurement [9–12].

3. Understanding the Space

In this section, we review the state-of-the-art in assigning semantic labels to columns in tabular data followed by the used terms and formulate the problem statement.

[9] ranks are numbered while grades are labels (e.g. A, B, ...)

[10] It is similar to log-interval but also is a ratio so we consider it here a derived ratio

[11] It is not exactly equal as absolute; it is more general that counted fractions, but it is enough approximation for our purpose in this paper.

[12] Even though Mosteller and Tukey warns the reader when re-expressing zeros, they do not create a separate category for it

3.1. State-of-the-art

The topic of semantic annotation for tabular data has been of interest for search engine companies as well as for the semantic web community for several decades. Different techniques have been used to perform this task: graphical probabilistic models [13–16], linear regression [17, 18], decision trees [19], hierarchical clustering [6], and support vector machine (SVM) [14], among others.

These techniques need to be applied on computed information extracted from the data, which are referred to as "features" in machine learning. Cafarella et al. [17] use attribute correlation statistics computed from the crawled web documents and schema coherency score. Features like the number of hits on the tables [17], column types [13, 14], and text similarity [14, 18], and the relation between columns/entities [13–15, 20] are all used and shown to be the main drivers of high-quality annotation. Statistical tests to compare distributions from different samples have also been adopted, especially for dealing with numerical data [6, 18, 21, 22].

From our review of the state-of-the-art we can see that there are two kinds of learning sources (training sets). The first kind uses knowledge graphs such as YAGO [14] and DBpedia [5, 6] to create learning sets. Others are more focused on learning from scraped web pages which do not provide such ease to focus on a specific domain [13, 17, 19, 20]. Despite the fact that these approaches may be automatic or semi-automatic[13], some of them require manual actions (e.g., provide predefined conversion rules [5, 16], a blacklist of properties [6] to improve the accuracy and abbreviations resolution [5, 16] while others rely on experts to semantically annotate columns to semantic properties such as [23]. We also found that they treat all numerical columns and their values in the same way, not taking into account the type of numerical values that they contain.

In this work, we first detect the type of the numerical values as a way to improve the performance of semantic labeling without matching the exact numbers to a property of a matched entity, relying on an ontology, profiling knowledge graphs, or manual elimination of properties or tweaking of parameters (that is knowledge graph dependent).

[13] We are not referring here to the gold standards that are built manually or the semantic models that are constructed by domain experts.

3.2. Problem Statement

We define a *dataset* as a collection of related data on a specific topic. In this work dataset, table and tabular data mean the same thing. Columns in a dataset may consist of numerical values, textual values or a combination of both. In this work, we focus on *numerical columns* and exclude any textual context, such as the header of the column or any text within column cells. This results into each numerical column being represented as a collection of numerical values. The **problem statement** of this work could be summarised as follows:

Given a collection of numerical values of specific type, a class describing the content of a dataset and a target knowledge base, return the list of properties in the knowledge base that most likely correspond to those numerical values, ordered by likelihood score.

Figure 1 outlines the inputs and outputs of the semantic labelling approach.

4. Typology of Numerical Columns

In this section, we present the typology of numerical columns that we use in out work. We discuss high-level types (*nominal*, *ordinal* and *ratio and interval*), and where applicable we further split those types to sub-types. We based our work on data types as discussed by Tukey [24] and Mosteller et al. [10] (as we mention in Section 2). We further discuss the applicability of those data types to our use-case of numerical columns within datasets published on the web and outline process of detection of each type.

4.1. Nominal

Nominal numbers are numbers which should not have any meaning beside identifying a group (or a class of some kind). This could mean that it is impossible to know whether the numbers in a given column is nominal or not without having extra context as they do not have a special meaning. But, understanding how these numbers are generated would give us some insights. Such numbers are used instead of names due to the uniqueness; multiple people can share the exact same name [25].

To ensure the uniqueness, some use a sequence of natural numbers (e.g., military units [26]). Some starts from large numbers and the sequence would be for example organised as (90000, 90001, 90002, ...). In sports, the numbers assigned to the players used to refer (some until today) to the position of the player in the field [25, 27, 28]. For example, in football (soccer) Goalkeepers tend to have squad number 1 [28]. In other cases, the number is a combination of segments, where each segment has a meaning [26, 29]. A segment of a number can mean the unit of the soldier (in the case of military personnel) [26], or the date of birth [29], sex, place of birth, and usually the last part of a such a nominal number ends with a random number or sequence [26, 29]. The third kind of nominal is the general case where a number represents a group, sometimes referred to as *categorical*. A common example is to have 1 to represent male and 2 to represent female in the sex column of tabular data about humans (scientists, athletes, artists, ...). The fourth kind of nominal is something that is either randomly generated (like automatically generated ids in software), or they only make sense for the platform like addresses in memory.

Here we identify four kinds (types) of nominal numbers: 1) sequential; 2) hierarchical; 3) categorical; 4) random. Below we explain how do we detect each of those nominal numbers. For nominal numbers in general, we expect all the numbers to be natural. Let X be the input collection of numbers,

$$\forall x_i \in X; x_i \in \mathbb{N}$$

Sequential nominal numbers can easily be detected if all the data exists. For example, if we have the sequence from 700 to 932, we can detect it by taking the minimum (700) and compare it with a generated sequence starting from that minimum value until the maximum (932). It becomes tricky when we have missing values due to the selected population having something in common (e.g., sequences of solider for a sub-unit). The intuition that we follow is that if more than the square root of the numbers in the generated sequence Y are also in the original collection of numbers, then we consider the collection of numbers as a sequential collection. So we consider a collection of numbers X sequential if the following equation is satisfied:

$$||X \cap Y|| > ||Y||$$

There is nothing magical about the square root; we can use a cube root or a simple percentage (e.g., 50%). At the end, it really depends on the kind of published

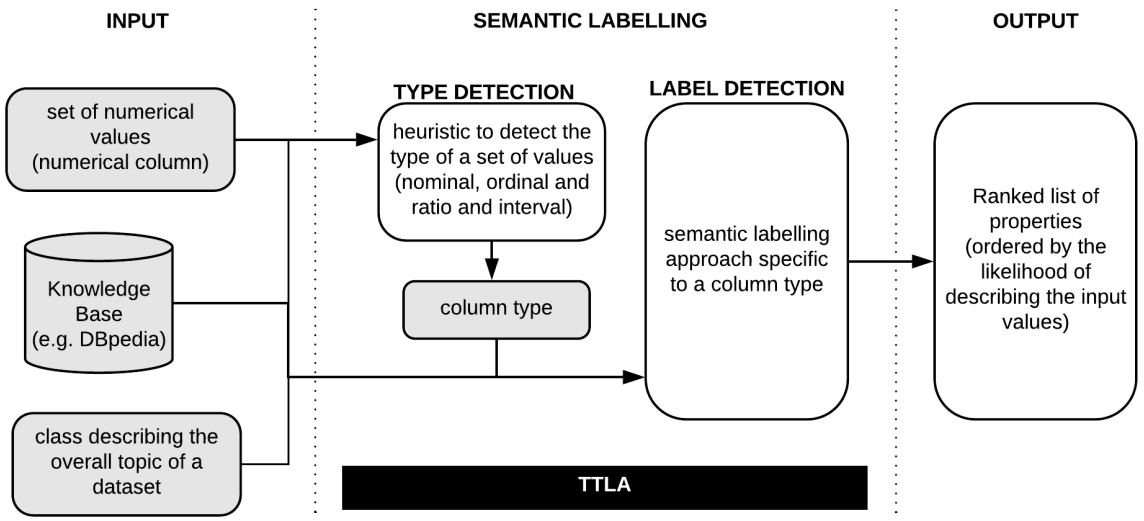

Fig. 1. Outline of inputs and outputs of the problem tackled in this work. Light gray highlights the inputs to the semantic labelling approach.

data that we are dealing with (e.g., how much noise in the data).

Hierarchical nominal numbers are more difficult to detect. They can be easily confused with another kind of nominal and even non-nominal numbers. They tend to include a sequence in their composition. They also have the same number of digits, but this is also the case for the sequential nominal numbers. In order to detect hierarchical data in a column, we first check the number of digits; if it is the same in all the cells, then we consider it to be hierarchical if it is not sequential. This is if the values are unique; having duplicate values is a strong evidence of non-hierarchical numbers. We have to admit that this detection method is not perfect, but it is an intuitive way of detecting hierarchical nominal numbers.

Categorical nominal numbers have some unique aspects compared to the other nominal numbers. They tend to have a large number of repetitions; the number of categories is an important signal to distinguish it from other categorical data. Another aspect of categorical numbers is that they are usually natural numbers. We consider a set of numbers X as categorical if they are nominal and the number of unique values U is much less << than the total population.

$$1 < ||U|| < \sqrt{||X||}$$

We outline the algorithm to detect nominal categorical data in Algorithm 1.

```
Function isCategorical(objects):
    counts = ∅
    foreach object do
        if object in counts then
            counts[object] += 1
        else
            counts[object] = 1
        end
    end
    if √objects.length > counts.keys.length > 1
     then
        True
    else
        False
    end
return
```

Algorithm 1: Detect Categorical Properties

A special case for having only one unique value is not considered categorical. We simply ignore that collection as extra knowledge would be needed to understand the meaning of this number.

Random nominal numbers are all of the remaining nominal numbers. But we have no way of detecting that, which makes sense because it is random. This

kind of numbers is usually manually removed as discussed by [6, 23].

4.2. Ordinal

Ordinal measure is one of the easiest cases to detect. Generally is it just a list of natural numbers starting from 1 until the last element of the list. An intuitive way to detect them is to see whether the set of numbers (what we want to examine) is equal to list of numbers 1 until the size of the list. Having a set of negative numbers or floats is a sign that the list of numbers we are examining is not ordinal.

4.3. Ratio and Interval

We put ratio and interval together because just looking at a zero; it is impossible to tell whether it is a "real" zero or just an agreement without extra information. Hence, we will treat them similarly, the same for cyclical, derived, and fundamental measures: as it is impossible without further evidence to distinguish between them. But we can distinguish simple counts from the rest of them. Counts should be positive by nature, and they do not have factions. Often have the difference between the square root for the maximum number and minimum number in the collection of numbers X is more than 1[14]. According to Mosteller and Tukey [10], this is the most commonly used "re-expression".

For numbers falling under the ratio-interval umbrella, we check whether the numbers are simple counts as follows:

$$1.5 * (Q_3 - Q_1) + Q_3 \leqslant P_{95} \tag{1}$$

$$\frac{(P_{95} - Q_2)}{Q_2} \geqslant 2 \tag{2}$$

P_{95} refers to the 95 percentile; Q_1,Q_2,Q_3 refers to the first, second, and third quartiles respectively. Equation 1 checks whether P_{95} is considered an outlier or not. Following the intuition that counts tends to increase a lot at the end (if ordered increasingly). We pick the P_{95} instead of the P_{100} to avoid possible noise or outliers (which can cause it to be falsely positive). Equation 2, follows the same intuition that counts tends to have a large increase at the top (large) percentiles and here we check if it doubles the numbers in the middle (knows as median or Q_2). Note that in some cases the number 2 can increase or decrease, but we found it to be a good balance in our preliminaries exploratory tests.

If Equations 1 and 2 are not satisfied, then we consider them to be anything else in the ratio and interval realm.

[14] Inspired by Tukey's way [24] for getting an intuition about what "re-expression" of data is better for getting insights from the data.

4.4. The Detection Order

We start by the ordinal ones. After that, we test for nominal-sequential and nominal-categorical. What is left is considered in ratio-and-interval group. A further distinction is made to check if it a simple count or other (which includes other kinds of ratio-and-interval like cyclical, balances, amounts, ..., etc.).

5. Model

In this section we describe how we extract numeric values from a given knowledge graph. We use the extracted data to build a model that we use afterwards to assign labels to numeric columns.

5.1. Data Extraction

We follow the method we published in our previous work [8]. In this step, we extract the numeric values from the knowledge graph.

To extract numeric properties from the knowledge graph, we use the following SPARQL query with a proper class URI. This will get all the properties for a given class.

```
SELECT distinct ?property WHERE {
?subject a <classURI>. ?subject ?property [].
} GROUP BY ?property
```

Then, we check whether more than 50% of the values (which are known as *objects*) are numeric. If so, then we consider this property as a numeric property which will be used with its values to build the model.

5.2. Model Construction

Before we build the model, we first need to detect the kinds of numbers for each property for the corre-

Table 1
Typology Detection and Labeling

Type	Sub-type	Detect	Label
Nominal	Sequential	yes	yes
Nominal	Hierarchical	yes	no
Nominal	Categorical	yes	yes
Nominal	Random	no	no
Ordinal	-	yes	no
Ratio-Interval	Counts	yes	yes
Ratio-Interval	Other	yes	yes

Table 2
Model Features

Property URI	Numeric Type/Sub-type	Features
.../military-service-number	sequential	95000, ...
.../height	other	192.4, ...
...	...	...

sponding class. The model for each class will be in the format shown in Table 2.

Below we show the different model construction methods for each numeric kind.

5.2.1. Nominal

The features used to build the model depend on whether the numbers are sequential, hierarchical, categorical, or random. Below we show how we compute features depending on different kinds of nominal numbers.

For the sequential kind, we use the trimean which is more resistant to outliers [24]. We show the formula here:

$$trimean = \frac{Q1 + 2 * Q2 + Q3}{4} \tag{3}$$

We also use the standard deviation with trimean instead of the mean. We refer to the standard deviation with trimean as *tstd*.

For the categorical kind, we use the unique number of numbers (the number of categories) followed by the percentages for each category ordered ascendingly. Note that in the feature column, a cell can have multiple features (comma separated).

Matching hierarchical with a numeric property from the knowledge graph is complex. If we know the different sections and which digits represent them, we would be able to do more[15], but this is not the case, and hence we won't be able to semantically annotate them.

For the nominal-random kind we ignore it because it is impossible to annotate such kind without extra evidence or elimination of other kinds, this is because it is random by definition.

5.2.2. Ordinal

This kind of data is common in tabular data, but we found that is very limited in knowledge graphs, which makes sense as such ordering usually done with some filtering (e.g, heights in zone A or among group B, ..., etc); hence, ordinal numbers will be ignored.

5.2.3. Ratio and Interval

These are the kinds of data that are more interesting to annotate. Comparing floating numbers can be a difficult or even expensive O(||X||*||Y||) to check if the collection of numbers *Y* are in the numeric property *X* with a slight measurement or accuracy difference (e.g., 0.1). For example, if we have a number in Y 4.1 which represents a height of something and the numeric property for the same entity in the knowledge graph is measured with higher accuracy like 4.100391; these two measurements won't match.

We can distinguish simple counts from the rest as it has different aspects that can be exploited to annotate the collection of numbers in a numeric column.

Counts Counts usually vary and using the raw values as-is is typically hard to understand and analyze [10, 24]. Annotating such data by machines is also difficult as the probability distribution or the numbers (counts) alone doesn't provide enough evidence to distinguish the data [10, 24]. Exact match won't be a solution as a single difference in counts or the data for the same events but after a single year could be very close but not exactly the same.

[15] like treating each section differently or treating each as a dimension

Nonetheless, exploratory data analysis techniques may help us here. Although they are generally meant to be used by humans to explore the data, we intend to automate this process here. So, we transform the data to help us in understanding them; explanatory data analysis experts use square root ($\sqrt{X}$) and logs ($\log X$); such transformation of data is referred to as "re-expression" of the data [10, 24]. Logs generally make the results too close while square roots tend to be a good balance between the logs and the raw values [24]; hence, we transform the raw data using the square root. After that, we use the trimean Equation 3 and the trimean-standard-deviation (tstd).

Others For the rest, we propose a way to overcome the need to perform exact match and the use of interval to make up for measurement accuracy differences. We use fuzzy c-means to perform automatic semantic labeling similar to our previous work [8]. We will be using the trimean [24] instead of the cluster centers as in Equation 3.

This reduces the effects of long-tailed distributions in the input data or the values of the numeric properties in the knowledge graph. We also use standard deviation and use trimean to compute it instead of the mean.

6. Semantic Labelling

In this section, we perform semantic labeling for the different types of data where we treat each data type differently to maximize accuracy. Since this difference was in computing the features, we can use the same algorithm in labeling the numeric columns.

For the labeling task, we use the fuzzy c-means clustering technique [30]. We can divide it into two steps: the first one is computing the centers of the clusters; the second step is the classification (labeling in our case). Note that in the classical fuzzy c-means, it computes the cluster centers based on the data points minimizing the total error. In our case, we compute the clusters from the points we extract from the knowledge graph, and we compute the features which will be the centers of the clusters. After that, we use the input data points of the numeric column and compute the features, which will be used to look for the closest cluster. As this is fuzzy clustering, it will belong to multiple clusters with a percentage of the belonging for each. Below we explain the steps further.

Table 3
Membership Vector Example

Cluster	Membership
B	0.8
A	0.15
C	0.5

6.1. Centroids

Differently from classical fuzzy c-means [30], we compute the clusters centers - which are also known as centroids - using the different features we present in Section 5. The reason is that in our case we already know that the numbers of a given numeric property belong to the same cluster while in the classical fuzzy c-means, the goal is to find clusters given a collection of data points.

6.2. Classification

Given a collection of numbers as an input, we compute the features as in Section 5. The computed features are then used as new data points. The classical fuzzy c-means formula can be used here. For each data point - which are the features computed for a given numeric column - the membership vector is computed. This membership vector indicates the percentage of belonging of a given column to each of the clusters in the model. For example, if we have three clusters: A, B, and C, a membership vector for a numeric column can be something like this $< 0.15, 0.80, 0.05 >$. This means that the given column belongs 15% to cluster A, 80% to cluster B, and 5% to cluster C. An output of the classification is the membership vector. For practical reasons, we order the results in descending order with the corresponding cluster and the membership vector as in Table 3. Note that the sum of the membership values for a given data point should be 1.

7. Evaluation

In this section, we evaluate our hypothesis that detecting and treating numeric values differently depending on the kind of numeric values would results to higher precision than using a general technique.

.We define the set of experiments to evaluate the hypothesis; we describe the data we use, report the results of the experiments, and we finish this section with the discussion of the results.

7.1. Experiment

As we are dealing with different kinds of numbers, we divide the input data for each kind: nominal-sequential, nominal-hierarchical, nominal-categorical, ordinal, ratio-and-interval-counts, ratio-and-interval-others. We do this manually, and then we apply the detection methods we reported earlier in this paper. This is done to measure the performance of our detection methods. Then, taking into account the knowledge that we have about the kinds of numbers, we apply the labeling methods and report the precision, recall, and F1 scores. We do this for each dataset. The results of the detection is a kind (e.g., nominal) and sub-kind (e.g., sequential). For the labeling, the output is the URI of the property of the numeric column that our algorithm determines to be more probable. The source code of the experiment and the data are published [31] (also available on GitHub [16])

7.2. Data

We use T2Dv2 dataset [32]: the only benchmark that we found the in the literature that has different types of numeric columns. We use 124 numerical columns from that dataset that we were able to understand and for which we find a corresponding property in DBpedia. We show the typology found in the dataset in Table 4. We can see that the most prominent lies under the ratio-interval type. The count type takes close to 0.4 of the total number of numerical columns. This has been reported in our previous paper [8] when talking about long-tailed distributions which often caused by the data being condensed in one of the sides far from the mean. This was the primary cause of incorrect labeling in that work. We also notice here that we have a type called *Year*, which is not mentioned in the background nor in the typology that we adopt. This is due to its unique properties. The type Year is confusing as it can be thought of as a simple count from the birth of Jesus until a given moment. This could also be thought of as a nominal since it is often used to tag and name things produced, created, or occurred in a year without any regard of the counting aspect. This can also be argued for the other types, but it is more common with the years. Since it can be compared, some might argue that it is not nominal (nominal does not encompass the greater or less than operator [9]). Due to that, we will be ignoring it in our experiments.

[16] https://github.com/oeg-upm/ttla

Table 4
Typology in T2Dv2 dataset

Numeric Type	Sub-type	Percentage
Nominal	Sequential	0.008
Nominal	Hierarchical	0.0
Nominal	Categorical	0.0
Nominal	Random	0.048
Nominal	combined	0.056
Ordinal	-	0.04
Ratio-Interval	Count	0.387
Ratio-Interval	Other	0.234
Ratio-Interval	combined	0.621
Year	-	0.282

7.3. Results and discussion

We first report the scores for the detection of the typology for each of the numeric columns in Table 5. For the sequential, we found that there is only one column and we got it wrong. The reason is the noise; it is actually not a single table in the file, instead, there are multiple tables with headers put in a sequence, which doesn't make them look like an actual sequence. From the benchmark T2Dv2, we did not find any hierarchical or categorical columns hence the N/A in the table. For the ordinal, we got high precision and recall applying a simple function. For the count, we reached a precision score of approximately 0.8 and a similar recall. We inspected the wrongly detected ones and we see that they do not look like counts. Counts tends to have a high variance as the number increases which was not the case in a couple of the columns that were wrongly detected. This is highly inspired by the work of Tukey [24]; even though his proposed work did not work for us as it was meant for humans to get insights from data and not for machines to distinguish simple counts from the other types. For the last type, it is a failback when the data do not match any of the above as mentioned earlier in the paper, so there is no specific detection algorithm per se for it.

For the labeling part, we report the precision, recall, and the F1 score. We do that for different values of k, taking the top k properties that are the most probable. For $k = 1$ - taking into account only the top suggested property - the resulted recall is high ($\geqslant 0.8$) while the precision is not (> 0.4). Looking closely to the type with the lowest precision, we found that it is ratio-interval-other; we were not expecting the precision of it to be low.

Table 5
Typology Detection Scores

Numeric Type	Sub-type	Precision	Recall	F1
Nominal	Sequential	0.0	0.0	N/A
Nominal	Hierarchical	N/A	N/A	N/A
Nominal	Categorical	N/A	N/A	N/A
Nominal	Random	N/A	N/A	N/A
Ordinal	-	0.8	1.0	0.889
Ratio-Interval	Count	0.792	0.809	0.8
Ratio-Interval	Other	0.552	0.516	0.533

We inspected the wrongly labeled properties and found that some are due to being labeled to knowledge graph specific/internal properties (e.g., *dbo:wikiPageID*) or wrong type detection. For example, the type of *areaOfCatchment* is wrongly detected as of type ratio-interval-count when building the model (while it should have been ratio-interval-other). Also, a couple of them wrongly point to properties related to years (e.g., *dbp:yearLeader*, *dbo:eruptionYear*[17], ..., etc.). We found that years are one of the most difficult to work with relying only on the values [8, 15]. This could also be improved by having a better type detection, but may not prevent wrongly assigning year-related properties if the used typology does not have a (sub-)type *year*.

Not only ratio-interval-other has wrongly labeled properties, but also other types have mislabeled properties. We scrutinize and found that there are properties that are simply just wrong. For example, the *dbp:iosNumber* of a *dbo:Currency* is labeled as *dbp:width*; we looked it up and found that there is nothing called the width of a currency. Looking at the values, they look similar to the values of dbp:iosNumber but not exactly the same, so it makes sense why it was confused with it.

As the k increases and takes more properties into account, we notice the significant score increase especially for the precision of ratio-interval-other; it rises from 0.486 to 0.914. This means those correct properties are still in the top suggested ones for the most part and the precision increase as we increase k.

We have carefully checked the result for recall for the nominal-sequential. This may be misleading, because there was only one column with that type and the algorithm got it correctly; so the two possible outcomes were 0 or 1. For the other types that have N/A, this is due to the lack of these types in either the input dataset or the knowledge graph. An exception to that is nominal-random; the reason is that we do not have a way to detect a such (sub-)type and will failback. As a result, the failback (sub-)type is ratio-interval-other.

[17]Note that "/ontology" here should be "/property", but our approach does not rely on the ontology behind

Table 7
Compare Labeling Scores

k	Approach	Precision	Recall	F1
1	TTLA	**0.687**	0.892	0.776
	FCM	0.34	-	-
	Random	0.0004	-	-
3	TTLA	**0.94**	0.892	0.915
	FCM	0.55	-	-
	Random	0.0012	-	-
5	TTLA	**0.976**	0.892	0.932
	FCM	0.83	-	-
	Random	0.002	-	-
10	TTLA	**1.0**	0.892	0.943
	FCM	0.91	-	-
	Random	0.004	-	-

We compare our approach with our previous work [8] as we did not find any other approach reporting the classification of numeric columns separately for the only two publicly available tabular datasets with numeric columns. We denote our approach in Table 7 with TTLA (which stands for Tabular Typology-based LAbeling) and the other approach with FCM as denoted in the other paper. We also include the scores of getting a correct property randomly from [8]. The scores of our approach (TTLA) for $k = 1$ achieve twice the precision as the previous work (FCM). TTLA continue to out-perform FCM as we increase k. We can see a clear significant improvement of precision. The recall is not reported in the previous work, so we put the sign $-$ in the corresponding cell, and we report the recall to allow comparison in future work.

8. Conclusion and Future Work

In this paper, we show that applying different methods for different types of data taking into account the typology yields better results in labelling numerical columns. We show a typology taking into account the task of semantic labeling. We found that some kinds of numerical properties do not exist in the current benchmarks, they are under-represented.

Table 6
Typology Labeling Scores

k	Numeric Type	Sub-type	Precision	Recall	F1
1	Nominal	Sequential	1.0	1.0	1.0
	Nominal	Hierarchical	N/A	N/A	N/A
	Nominal	Categorical	N/A	N/A	N/A
	Nominal	Random*	N/A	N/A	N/A
	Ordinal	-	N/A	N/A	N/A
	Ratio-Interval	Count	0.83	0.957	0.889
	Ratio-Interval	Other†	0.486	0.8	0.604
	All	-	0.687	0.892	0.776
3	Nominal	Sequential	1.0	1.0	1.0
	Nominal	Hierarchical	N/A	N/A	N/A
	Nominal	Categorical	N/A	N/A	N/A
	Nominal	Random*	N/A	N/A	N/A
	Ordinal	-	N/A	N/A	N/A
	Ratio-Interval	Count	0.957	0.957	0.957
	Ratio-Interval	Other†	0.914	0.8	0.853
	All	-	0.94	0.892	0.915
5	Nominal	Sequential	1.0	1.0	1.0
	Nominal	Hierarchical	N/A	N/A	N/A
	Nominal	Categorical	N/A	N/A	N/A
	Nominal	Random*	N/A	N/A	N/A
	Ordinal	-	N/A	N/A	N/A
	Ratio-Interval	Count	1.0	0.957	0.978
	Ratio-Interval	Other†	0.943	0.8	0.866
	All	-	0.976	0.892	0.932
10	Nominal	Sequential	1.0	1.0	1.0
	Nominal	Hierarchical	N/A	N/A	N/A
	Nominal	Categorical	N/A	N/A	N/A
	Nominal	Random*	N/A	N/A	N/A
	Ordinal	-	N/A	N/A	N/A
	Ratio-Interval	Count	1.0	0.957	0.978
	Ratio-Interval	Other†	1.0	0.8	0.889
	All	-	1.0	0.892	0.943

* Unable to detect and fail back
† Include failed back (sub-)types

Although this work demonstrates how taking into account the typology of numbers can improve the performance of semantic labeling of numeric columns, yet it is just the tip of the iceberg. Further in-depth inspection and even simulation can be employed to extend this work and more tests with upcoming benchmarks.

References

[1] Y.A. Sekhavat, F.D. Paolo, D. Barbosa and P. Merialdo, Knowledge Base Augmentation using Tabular Data, in: *LDOW*, 2014.

[2] E. Kacprzak, L. Koesten, L.-D. Ibáñez, T. Blount, J. Tennison and E. Simperl, Characterising dataset searchâĂŤAn analysis of search logs and data requests, *Journal of Web Semantics* (2018).

[3] N. Noy, M. Burgess and D. Brickley, Google Dataset Search: Building a search engine for datasets in an open Web ecosystem, in: *28th Web Conference (WebConf 2019)*, 2019.

[4] V. Efthymiou, O. Hassanzadeh, M. Rodriguez-Muro and V. Christophides, Matching web tables with knowledge base entities: from entity lookups to entity embeddings, in: *International Semantic Web Conference*, Springer, 2017, pp. 260–277.

[5] D. Ritze, O. Lehmberg and C. Bizer, Matching html tables to dbpedia, in: *Proceedings of the 5th International Confer-*

ence on Web Intelligence, Mining and Semantics, ACM, 2015, p. 10.

[6] S. Neumaier, J. Umbrich, J.X. Parreira and A. Polleres, Multi-level semantic labelling of numerical values, in: *International Semantic Web Conference*, Springer, 2016, pp. 428–445.

[7] J. Mitlöhnert, S. Neumaier, J. Umbrich and A. Polleres, Characteristics of Open Data CSV Files, in: *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 72–79. doi:10.1109/OBD.2016.18.

[8] A. Alobaid and O. Corcho, Fuzzy Semantic Labeling of Semi-structured Numerical Datasets, in: *European Knowledge Acquisition Workshop*, Springer, 2018, pp. 19–33.

[9] S.S. Stevens et al., On the theory of scales of measurement (1946).

[10] F. Mosteller and J.W. Tukey, Data analysis and regression: a second course in statistics., *Addison-Wesley Series in Behavioral Science: Quantitative Methods* (1977).

[11] N.R. Chrisman, Rethinking levels of measurement for cartography, *Cartography and Geographic Information Systems* **25**(4) (1998), 231–242.

[12] D. Lane, J. Lu, C. Peres, E. Zitek et al., Online statistics: An interactive multimedia course of study, *Retrieved January* **29** (2008), 2009.

[13] A. Goel, C.A. Knoblock and K. Lerman, Exploiting structure within data for accurate labeling using conditional random fields, in: *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 1.

[14] G. Limaye, S. Sarawagi and S. Chakrabarti, Annotating and searching web tables using entities, types and relationships, *Proceedings of the VLDB Endowment* **3**(1–2) (2010), 1338–1347.

[15] M. Taheriyan, C.A. Knoblock, P. Szekely and J.L. Ambite, Learning the semantics of structured data sources, *Web Semantics: Science, Services and Agents on the World Wide Web* **37** (2016), 152–169.

[16] M. Zhang and K. Chakrabarti, Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables, in: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ACM, 2013, pp. 145–156.

[17] M.J. Cafarella, A. Halevy, D.Z. Wang, E. Wu and Y. Zhang, Webtables: exploring the power of tables on the web, *Proceedings of the VLDB Endowment* **1**(1) (2008), 538–549.

[18] M. Pham, S. Alse, C.A. Knoblock and P. Szekely, Semantic labeling: a domain-independent approach, in: *International Semantic Web Conference*, Springer, 2016, pp. 446–462.

[19] Z. Syed, T. Finin, V. Mulwad and A. Joshi, Exploiting a web of semantic data for interpreting tables, in: *Proceedings of the Second Web Science Conference*, Vol. 5, 2010.

[20] P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao and C. Wu, Recovering semantics of tables on the web, *Proceedings of the VLDB Endowment* **4**(9) (2011), 528–538.

[21] S. Ramnandan, A. Mittal, C.A. Knoblock and P. Szekely, Assigning semantic labels to data sources, in: *European Semantic Web Conference*, Springer, 2015, pp. 403–417.

[22] E. Kacprzak, J.M. Giménez-García, A. Piscopo, L. Koesten, L.-D. Ibáñez, J. Tennison and E. Simperl, Making sense of numerical data-semantic labelling of web tables, in: *European Knowledge Acquisition Workshop*, Springer, 2018, pp. 163–178.

[23] A. Merono Penuela, Refining Statistical Data on the Web (2016).

[24] J.W. Tukey, Exploratory data analysis. 1977, *Massachusetts: Addison-Wesley* (1976).

[25] Uniform number (Major League Baseball), https://en.wikipedia.org/wiki/Uniform_number_(Major_League_Baseball) [Online; accessed 14-March-2019].

[26] Service number, https://en.wikipedia.org/wiki/Service_number[Online; accessed 14-March-2019].

[27] Baseball positions, https://en.wikipedia.org/wiki/Baseball_positions [Online; accessed 14-March-2019].

[28] Squad number, https://en.wikipedia.org/wiki/Squad_number_(association_football) [Online; accessed 14-March-2019].

[29] A.S. Lunde, The person-number systems of Sweden, Norway, Denmark, and Israel (1980).

[30] J.C. Bezdek, R. Ehrlich and W. Full, FCM: The fuzzy c-means clustering algorithm, *Computers & Geosciences* **10**(2–3) (1984), 191–203.

[31] A. Alobaid, E. Kacprzak and O. Corcho, TTLA, 2019. doi:10.5281/zenodo.2619307.

[32] D. Ritze, O. Lehmberg and C. Bizer, T2Dv2 Gold Standard for Matching Web Tables to DBpedia, 2015, [Online; accessed 26-June-2018].