Semantic labeling for quantitative data using Wikidata

Phuc Nguyen and Hideaki Takeda

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{phucnt, takeda}@nii.ac.jp

Abstract. Semantic labeling for quantitative data is a process of matching numeric columns in table data to a schema or an ontology structure. It is beneficial for table search, table extension or knowledge augmentation. There are several challenges of quantitative data matching, for example, variety of data ranges or distribution, and especially, different measurement units. Previous systems use several similarity metrics to determine column numeric values and corresponding semantic labels. However, lack of measurement units can lead to incorrect labeling. Moreover, the attribute columns of different tables could be measured by units differently. In this paper, we tackle the problem of semantic labeling in various measurement units and scales by using Wikidata background knowledge base (WBKB). We apply hierarchical clustering for building WBKB with numeric data taken from Wikidata. The structure of WBKB follows the nature taxonomy concept of Wikidata, and it also has richness information about units of measurement. We considered two transformation methods: z-score-tran based on standard normalization technique and unit-tran based on restricted measurement units for each semantic label of WBKB. We tested two transformation methods on six similarity metrics to find the most robust metric for Wikidata quantitative data. Our experiment results show that using unit-tran and ks-test metric can effectively find corresponding semantic labels even when numeric columns are expressed in different units.

Keywords: semantic labeling, quantity, unit of measurement, tabular data, LOD, Wikidata

In the era of Open Data, a large number of table data resources has been published on the Web or open data portals. The semistructured of tables make it easier for extracting and interpreting data in comparison with other unstructured data resources. For this reason, table data structure have been getting more attention to academia. Extracting and interlinking these table data resources will be beneficial for table search, table extension or knowledge base augmentation.

Semantic labeling for table data structure involves two main problems as schema matching and data matching [1]. In the schema matching task, the system performs matching a table column to knowledge base property and tables which have one or multiple columns to knowledge base classes. In the data matching task, each row in a table will be matched to a knowledge base instance. In this paper, we do not study the data matching problem; our evaluation focuses on schema matching.

The previous textual-based semantic matching system performs well on Web tables where textual information, i.e., columns header, cell labels, table captions, or surrounding text is available. However, tables in Open Data portals usually have a lot of numerical columns and lacking textual description, ambiguity headers or cell labels; it is challenging to perform correctly matching when we cannot use on any textual information.

Neumaier et al. [2] tackle the problem of semantic labeling for such tables with an assumption that the attribute column has the same unit with knowledge bases. However, numeric columns of tables are not necessarily represented by the same units in the real world setting. For instance, people in the US usually use foots to measure height while European usually use meters to measure the same thing. If measurement units are not considered, it could lead to incorrect matching when comparing numeric values only.

In this work, we study a problem of finding the corresponding quantity property, measurement unit and types for numerical columns. There are many previous systems perform matching to common knowledge bases, such as DBpedia, YAGO, Freebase. However, so far there is no work perform semantic labeling for Wikidata. Currently, Wikidata community is curated and maintained by thousands of users. Matching to Wikidata schema will be meaningful for benchmarking the previous methods.

Given a numerical column, the system will find a corresponding Wikidata quantity property, a Wikidata item which is a unit of measures, and list of Wikidata items which are types for such columns. Figure 1 show an example of our semantic labeling system. Suppose that we need to make labeling for a numerical column which has a list of numbers as 1.7, 1.8, 1.9, 2.1, the output of our system is quantity property P2048 (height). Unit of measurement is Q11573 (meter). Measurement objective is a set of Wikidata items from specific to abstract: Q5(human), Q215627(person).

Our contributions to this paper as follows:

1. We apply hierarchical clustering for building a background knowledge base (WBKB) with numeric data taken from Wikidata. The structure of WBKB follow the nature taxonomy concept of Wikidata, and it also has rich information about measurement unit of each numeric value.
2. We proposed two transformation methods: z-score-tran relied on standard normalization technique and unit-tran which based on restricted measurement units used for each semantic label of WBKB.
3. In the setting of unit-tran, we obtain measurement units of numerical columns. This method uses the restricted units for each node for finding the most relevant unit used for input data.
4. We test six similarity metrics for numerical data which are proposed in previous work on WBKB. After that, we test our transformation method with the best similarity metrics on several testing samples. The experiment

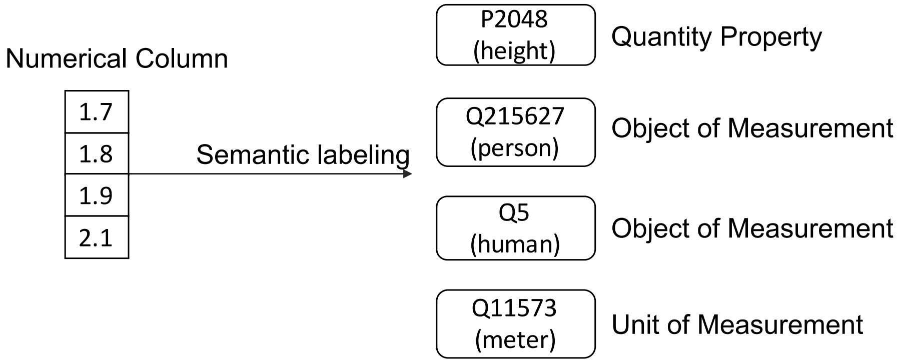

Fig. 1. An example of semantic labeling for a numerical column

runs on five-fold cross-validation over testing samples show that apply our transformation methods can help to improve semantic labeling performance.

0.1 Related Work

Regarding to units of measurement extraction, the previous system [3], [4], [5], [1], [6] use the textual description available for extracting quantitative data and units of measurement. Chaudron [3] is a system of extracting the quantitative data from Wikipedia InfoBox. Ibrahim et al. [4] proposed a system for canonicalizing entities and quantities. They use table header and surrounding text to find the unit information. Sarawagi and Chakrabarti [5] focus on the problem of table searching with numerical values. They use PCFG for parsing the units from header or cell labels. Ritze et al. [1] proposed a general framework for matching web table to DBpedia. These frameworks have also considered unit detection as a preliminary step of their framework. To detect unit of measurements they use textual information from table header and cell labels. InfoGather+ [6] develop a problem of table to table matching based on entity augmentation. Their method considers extracting the header table for finding unit information. Different from their approaches, we use only quantity values to determine unit information of numerical columns.

In some domains of knowledge, Buche et al. [7] provide a unit ontology for chemical risks. Hignette et al. [8] use a domain ontology to detect units in tables from the microbiology domain. Our system extract information of measurements from Wikidata which is a cross-domain knowledge base. From Wikidata query service, we get 195 distinct units cover many quantity properties and 246 unit conversion rules.

Ramnandan et al. [9] proposed semantic labeling for textual and numerical data. For numeric data, they use statistical hypothesis testing to find how similar of the empirical distribution of values and training data. Pham et al. [10] extended the work of Ramnandan et al. [9] using multiple similarity metrics as features. They create a classifier model build from metric learning for cross-domain prediction.

The most similar to our work is Neumaier et al. [2]. They study the problem of schema matching for numerical values to a hierarchical background knowledge graph. This graph is generated from numerical values taking from DBpedia. These numbers are grouped based on pairs of property and classes in DBpedia ontology. To provide the semantic label for numerical columns, the k-nearest neighbor's search is performed for finding the most similar nodes. However, limited support to handle measurement as one of limitation of DBpedia [3], their approaches do not consider the unit issues of numeric values. If the numerical column is expressed in different scale with the background knowledge graph, it is impossible to find correct answers. Then, semantic labeling for quantitative data must consider the unit issues in other to make properly matching.

1 Approach

1.1 Semantic Labeling for Quantitative Data

Notation:

- C: set of semantic labels of types and property-object pair nodes in WBKB
- U: set of units of measurement
- F: set of unit conversion rules. $F(u_{v_q}, u_{v_p})$ is the rules of convert from u_{v_q} to u_{v_p}.
- q: a query. $q \in C$
- p: semantic label of a node in WBKB. $q \in C$
- v_q: set of numbers of q. $v_q \in R$
- v_p: set of numbers of p. $v_p \in R$
- u_{v_q}: unit of v_q. $u_{v_q} \in U$
- U_p: set of units used in p. $U_p \in U$
- u_{v_p}: unit of v_p. $u_{v_p} \in U_p \in U$

Given q is a query of a numeric column with a list of numbers v_q, property label l_{v_q}, one or multiple context description c_{v_q}, and a unit of measurement u_{v_q}. Semantic labeling system perform K-nearest neighbor to find a corresponding node p in WBKB with property label l_{v_p}, one or multiple context description c_{v_p}, and a unit of measurement $u_{v_i} \in U_p$.

To provide the unit information for numerical columns, measurement unit information must be available in the background knowledge base. Neumaier et al. [2] use numeric values taking from DBpedia to build background knowledge base. However, DBpedia has very limited support to handle units of measurements [3]. To tackle the problem of unit issues, we use quantitative data of Wikidata as a knowledge background. Wikidata have well support for quantitative data, each numeric value comes along with its unit of measurement, and each quantity property has information about how many units used for this property. Similar to the work of Neumaier et al. [2], we build a background knowledge graph with numerical values and measurement units taking from Wikidata. Each node has the information about the canonical unit and other restrictedunits or scales. Suppose that we have a list of numbers taking from a table column, before using similarity metric for comparing the list of numbers and WBKB nodes, a transformation method is performed to find an appropriate scale of input numbers list with the scale of nodes restrictedunits. By this way, we can obtain correct nodes even if the input list of numbers is expressed in different of units or scales.

1.2 WBKB construction

The automatic approach to build NKB from Wikidata is modified from [2] having following steps:

Step 1: Query data from Wikidata.

1. Getting quantity properties.
We can get quantity properties directly from Wikidata SPARQL endpoint by the SPARQL Query 1.1:

Query 1.1. Getting Wikidata quantity properties query

```
SELECT DISTINCT ?property
WHERE {
    ?property wikibase:propertyType wikibase:Quantity.
}
```

Overall, we have 388 quantity properties. We sort and select only 50 most popular quantity properties for our experiments

2. Getting all restrictedunits for a property. Next, we continue using the SPARQL Query 1.2 for getting information about how many unit used by each property. In Wikidata, all properties have ID start with P character and after that a number. For example, P2049 is Wikidata ID of quantity property named width. Overall, we get 195 distinct units used for quantitative data and 246 unit conversion rules. The most popular unit of measurement is Q21027105 (Quantity property without units) which is used 32 times. Property P2043 (length) have the largest number of restrictedunits with 15 measurement units used for this property

Query 1.2. Getting restricted units for a quantity property

```
SELECT ?restricted_units{
    wd:PropertyID wdt:P2237 ?restricted_units.
}
```

3. Getting all subjects, values, and units of each property:
The SPARQL query for getting subjects, values, and units of values is shown in Query 1.3.

Query 1.3. Getting all subjects and values including with their unit of measurements each quantity property

```
SELECT ?subject ?value ?unit
WHERE {
    ?subject p:PropertyID/psv:PropertyID
    [wikibase:quantityAmount ?value;
     wikibase:quantityUnit ?unit].
}
```

4. Getting all types of a subjects:
Types of subjects are used to construct the NKB type layer. This includes the direct type and indirect types which is parent types of direct type. The SPARQL query for getting all types of a subject is shown in Query 1.4. P31 is the Wikidata ID of instance of property. P279 is the Wikidata ID of subclass of property. **SubjectID** denote for a entity on Wikidata. For example: Tokyo City have Wikidata **SubjectID** Q1490

Query 1.4. Getting all types of subjects

```
SELECT ?type
WHERE {
   {wd:SubjectID wdt:P31 ?t} UNION
   {?subject wdt:P31/wdt:P279* ?t}
}
```

5. Getting type hierarchy:
The property P279 (subclass of) is used to extract the type hierarchy from Wikidata. The query 1.5 shows how to get type hierarchy on Wikidata. TypeID is a Wikidata ID of type. For example: Q515 (City) is a subclass of Q486972 (human settlement).

Query 1.5. Getting all hierarchy types

```
SELECT ?parents_type
WHERE {
    wd:TypeID wdt:P279 ?parents_type.
}
```

Step 2: WBKB construction
Similar to Neumaier et al. [2], we have also build WBKB with two type of layers. The first layer is called as type hierarchy which represents the types of subjects. The second layer is called as a p-o hierarchy which is sub-nodes of type nodes. To

construct p-o hierarchy, subjects of type nodes are grouped as common property - object pair.

However, as a preprocessing step, all numeric values are converted to a canonical unit which is the most popular unit used by each property.

1.3 Similarity metrics

To distinguish two list of numbers, we consider using several similarity distances which was used in previous works as follows:

1. Range:
   The range similarity is really important to determine two list is measured in the same or different unit. We use the Jaccard similarity [10] for measure the similarity range. q is a label of query. p is a semantic label in NKB. v_p is all numerical values in p. v_q is all numerical value in q.
   $$s_{range}(v_q, v_p) = \frac{min(max(v_q), max(v_p)) - max(min(v_q), min(v_p))}{max(max(v_q), max(v_p)) - min(min(v_q), min(v_p))} \tag{1}$$
2. Statistic measurement:
   Welchs t-test is used for calculating the statistical hypothesis test. This similar metric used in [9]. Given two samples of data, the t statistic is defined by 2, and the similarity is measure by getting p_value of $t(v_q, v_p)$. X_{v_x}, s_x and N_i are the sample mean, sample variance and sample size of the q and p respectively. This measurement is sensitive with mean and variance of list of numbers:
   $$t(v_q, v_p) = \frac{\bar{X}_{v_q} - \bar{X}_{v_q}}{\sqrt{\frac{s_q^2}{N_q} + \frac{s_p^2}{N_p}}} \tag{2}$$
   $$s_{t\_test}(v_q, v_p) = p\_value(t(v_q, v_p)) \tag{3}$$
3. Cumulative Distribution with KS test:
   Kolmogorov - Smirnov (KS) Test similarly is use for comparing distribution functions of two samples. In the work of [9], and [10], $s_{cdf\_ks\_test\_p\_value}(v_q, v_p)$ is used as a similarity metric, while $s_{cdf\_ks\_test\_d}(v_q, v_p)$ is used in [2] work.
   $$s_{ks\_test\_d}(v_q, v_p) = \sup_x |F_{1,N_q}(x) - F_{2,N_p}(x)| \tag{4}$$
   $$s_{ks\_test\_p\_value}(v_q, v_p) = p\_value(s_{cdf\_ks\_test\_D}(v_q, v_p))) \tag{5}$$
4. Cumulative Distribution with KullbackLeibler divergence:
   I use KullbackLeibler divergence for measure how similarity CDF distribution of two list of number. The similarity distance is s_{cdf_kl}.
5. Histogram: Mann-Whitney test is a good technique to measure histogram of numeric values. This similarity it is used in work of [10]. The method tests ranks all values from the two samples from low to high and then computes a p_{value} that depends on the difference between the mean ranks of the two samples.
   $$s_{u\_test} = p\_value(u(v_q, v_p)) \tag{6}$$

1.4 Transformation method

The idea of transformation method is making numeric values of columns to the same scale with each node in WBKB. To do it, we consider two transformation methods as follows

1. Z-score Transformation:
   We use this transformation method to normalize v_q and v_p to normal distribution. It help to make sure that v_q have the same scale with v_p. The z-score transformation is calculated by following equation

$$tran_{z\_score}(v_i) = \frac{v_i - \bar{v_i}}{\sigma(v_i)} \tag{7}$$

2. Unit Transformation:
   Using the restricted units of each node in WNKB, the unit tranformation method find a approriated scale with unit conversion based on closest median of $tran_{unit}(v_q)$ and median v_p.

$$tran_{unit}(v_q) = f_{convert}(v_q, F(u_{v_q}, u_{v_p})) \tag{8}$$

   Almost the rule for conversion is multiplication. However, in the temperate conversion, the formula of conversion are performed. In order to get u_{v_q} we have to calculate:

$$u_{v_q} = \underset{u_p^i}{\operatorname{argmin}}(|median(v_p) - f_{convert}(median(v_q), F(u_p^i, u_{v_p})|) \tag{9}$$

2 Experiments

2.1 Training and Testing

We use the same setting with Neumaier et al. [2]. We use five-fold cross-validation technique for spiting quantitative values of each numerical property to two part. 80% values and 20% values are assigned for building WBKB training and WBKB testing respectively.

Testing samples are extracted from leaves nodes of WBKB testing. Since the complexity of Wikidata schema, the type nodes structure of WBKB testing is not necessarily similar to WBKB training. It is one of difference between Wikidata schema and DBpedia Ontology. In DBpedia, data extract from template matching from Wikipedia InfoBox. Then numerical values in DBpedia share the same schema structure. Meanwhile, data on Wikidata can be edited by everyone; a concept-value can be linked to many concepts.

We select the most of 50 properties for building WBKB. The number of types in Wikidata much larger comparing to DBpedia. As 50 most quantity properties of Wikidata, we get 1896 distinct type, while only 198 distinct types extracted from the work of Neumaier et al. [2]. Additionally, Wikidata has much more amount of numeric data and also ranges of data more larger in comparison with

DBpedia [2]. The left figure 2 depicts the 5% to 95% inter-quartile ranges of 50 property in logarithmic 10. Property P1090 (redshift) has the shortest range, i.e., 0.035, while the largest range is 6.36E+28 of property P2067 (mass). The right figure 2 shows total numeric values per each property. The largest total number is property P2044 (elevation above sea level) with around 13M numeric value, while the smallest total number is property P1697 (total valid votes).

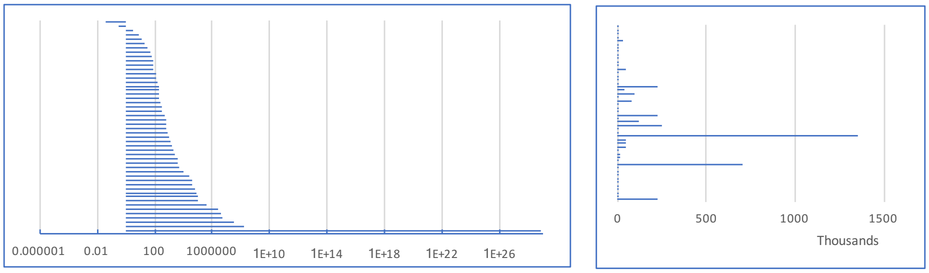

Fig. 2. 5%-95% inter-quantile ranges and number of values of WBKB

We consider generating three testing samples sets, i.e., sam-set, dif-set, and all-set. In the sam-set, all the samples are expressed in the same unit with the training WBKB. To get the sam-set samples, we shuffle random select maximum 50 leave nodes in WBKB testing. In total, we get average 1038 samples for five cross-validations. In the dif-set, the samples are expressed in a different unit with training WBKB. We use the restrictedunit for each node for converting the samples in sam-set to different scales. We shuffle random select 1038 samples from converting samples as dif-set. The all-set is the combination of all samples in sam-set and dif-set.

I experiment with three sets of samples. The sam set uses the same unit with WBKB, the dif set uses different units with WBKB, and all set is the combination of sam set and dif set with ration is 50% for each set. The evaluation is an aggregation of top k (in this experiment, we set k value is 50 which is similar to the best settings of [2]).

The 50 properties, training, and testing samples are available at [1].

2.2 Experiment setup

We set up three experiments as follows:

1. Experiment 1: What is the most effective similarity metric?
   In this experiment, we use the sam-set to test how the performance of six similarity metrics on WBKB. The most robustness similarity metric will be used in experiment 2 and experiment 3.

[1] https://github.com/PhucntNII/wbkb

2. Experiment 2: How well the transformation method improving the semantic labeling?
   We test the performance of the best similarity metric and Two transformation method on the dif set.
3. Experiment 3: How the performance of transformation method on the mixture all-set?
   Similar to the experiment 2, but we test on the all-set. In this experiment, we also test how well performances of measurement unit labeling perform?

2.3 Evaluation metric

To measure the accuracy of the top-k neighbors, we use prop and type measures of [2]. The prop measure the top-k neighbors contains the correct property label. We modify the type measure to the top k neighbors contain the correct type path which is the hierarchy structure of semantic context in WBKB. Similar to [2] We also show the accuracy in top one, five, and ten accuracies.

2.4 Results

Experiment 1 Table 1 depict the result of experiment 1. Applying the metric ks_test_d provide the best result. This result similar to the system of [2] where ks_test_d perform as the best similarity metric for DBpedia data. The ks_test_p_value metric have also provided a comparable result with the ks_test_d. [9] test this similarity metric on domain data provide the best performance. Overall, from previous work and the result of experiment 1, the ks-test method is suitable for numerical data. Because using metric ks_test_d will gave the best result, we only use the ks_test_d metric for evaluation in the next experiment.

Table 1. Similarity metric comparison on the sam-set

	prop			type		
Top	1	5	10	1	5	10
ks_test_d	**0.3888**	**0.7711**	**0.9193**	**0.1295**	**0.2568**	**0.3566**
ks_test_p_value	0.3526	0.7114	0.874	0.0798	0.2268	0.3171
kl	0.0468	0.258	0.4842	0.005	0.0116	0.0195
t_test	0.0692	0.3204	0.5079	0.0137	0.0331	0.0507
jaccard	0.2328	0.543	0.706	0.0422	0.0927	0.1776
u_test	0.1139	0.4102	0.6258	0.0108	0.0744	0.117

Experiment 2 Table 2 illustrate the experiment 2 results. It clear that using the sample which is different scale with WNKB is really hard for making the correct labeling. Apply transformation method before calculate similarity metric will improve the accuracy of semantic labeling.

Table 2. Transformation method evaluation on the dif-set

	prop			type		
Top	1	5	10	1	5	10
ks_test_d	0.0428	0.2114	0.3091	0.0029	0.0121	0.0245
z_score_ks_test_d	0.0297	0.1647	**0.5428**	0.0021	0.0324	0.068
unit_tran_ks_test_d	**0.1272**	**0.3854**	0.4565	**0.0191**	**0.0609**	**0.11**

Experiment 3 Table 3 illustrate the experiment 3 results. Using the unit transformation with ks_test_d metric will give the best result on the mixture unit sample set.

Table 3. Comparision of tranformation method on the all-set and unit measurment labeling

	prop			type			unit		
Top	1	5	10	1	5	10	1	5	10
ks_test_d	0.2158	0.4912	0.6142	0.0662	0.1345	0.1906	0.1944	0.3855	0.4596
z_score_ks_test_d	0.0288	0.1457	0.4329	0.002	0.0255	0.0498	0.0122	0.074	0.1634
unit_tran_ks_test_d	**0.2487**	**0.5711**	**0.6834**	**0.0684**	**0.1442**	**0.2198**	**0.2158**	**0.4529**	**0.5536**

2.5 Conclusion and Future Work

In this paper, we perform semantic labeling for numerical columns to WNKB with quantitative data taken from Wikidata. We benchmark six similarity metrics on numerical data of Wikidata that obtain that KS Test provides the best results. To tackle the problem of the different measurement unit in numerical columns, we proposed two transformation method to convert a list of numbers to the same scales with each node in WBKB. Finally, if the samples are expressed in multiple units, using the unit transformation and ks-test metric, our system can get a top 10 accuracy as 0.6834 accuracies for property labeling, 0.2158 for type labeling, and 0.5536 for unit labeling. In the future, we plan to expand this work for semantic labeling Open Data portal tables where it combination method for dealing with textual and numerical data is considered. Another direction for this paper is using the result for building table extending such as generating table caption or description.

References

1. Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, page 10. ACM, 2015.

2. Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. Multi-level semantic labelling of numerical values. In *International Semantic Web Conference*, pages 428–445. Springer, 2016.
3. Julien Subercaze. Chaudron: Extending dbpedia with measurement. In *European Semantic Web Conference*, pages 434–448. Springer, 2017.
4. Yusra Ibrahim, Mirek Riedewald, and Gerhard Weikum. Making sense of entities and quantities in web tables. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1703–1712. ACM, 2016.
5. Sunita Sarawagi and Soumen Chakrabarti. Open-domain quantity queries on web tables: Annotation, response, and consensus models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 711–720. ACM, 2014.
6. Meihui Zhang and Kaushik Chakrabarti. InfoGather+. *Proceedings of the 2013 international conference on Management of data - SIGMOD '13*, page 145, 2013.
7. Patrice Buche, Juliette Dibie-Barthelemy, Liliana Ibanescu, and Lydie Soler. Fuzzy web data tables integration guided by an ontological and terminological resource. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):805–819, 2013.
8. Gaëlle Hignette, Patrice Buche, Juliette Dibie-Barthélemy, and Ollivier Haemmerlé. An ontology-driven annotation of data tables. In *International Conference on Web Information Systems Engineering*, pages 29–40. Springer, 2007.
9. S Krishnamurthy Ramnandan, Amol Mittal, Craig A Knoblock, and Pedro Szekely. Assigning semantic labels to data sources. In *European Semantic Web Conference*, pages 403–417. Springer, 2015.
10. Minh Pham, Suresh Alse, Craig Knoblock, and Pedro Szekely. Semantic labeling: A domain-independent approach. In *ISWC 2016 - 15th International Semantic Web Conference*, 2016.