# Assigning Semantic Labels to Data Sources

S.K. Ramnandan[1(✉)], Amol Mittal[2], Craig A. Knoblock[3], and Pedro Szekely[3]

[1] Indian Institute of Technology - Madras, Chennai, India
nandparikrish@gmail.com
[2] Indian Institute of Technology - Delhi, New Delhi, India
amolmittal.iitd@gmail.com
[3] University of Southern California, Los Angeles, USA
{knoblock,pszekely}@isi.edu

**Abstract.** There is a huge demand to be able to find and integrate heterogeneous data sources, which requires mapping the attributes of a source to the concepts and relationships defined in a domain ontology. In this paper, we present a new approach to find these mappings, which we call semantic labeling. Previous approaches map each data value individually, typically by learning a model based on features extracted from the data using supervised machine-learning techniques. Our approach differs from existing approaches in that we take a holistic view of the data values corresponding to a semantic label and use techniques that treat this data collectively, which makes it possible to capture characteristic properties of the values associated with a semantic label as a whole. Our approach supports both textual and numeric data and proposes the top $k$ semantic labels along with their associated confidence scores. Our experiments show that the approach has higher label prediction accuracy, has lower time complexity, and is more scalable than existing systems.

**Keywords:** Semantic labeling · Source modeling

## 1 Introduction

Semantic labeling of a data source involves assigning a class or property in an ontology to each attribute of a data source. When the source is a table, the objective is to assign to each column in the table a class or property that specifies the semantics of the column. When the source is more complex, such as an XML or JSON file, the objective is to map each attribute of the source to a class or property that specifies its semantics. The goal of our work is to learn a

This research is was supported in part by IARPA via AFRL contract number FA8650-10-C-7058 and in part by DARPA via AFRL contract number FA8750-14-C-0240. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DARPA, AFRL, or the U.S. Government.


F. Gandon et al. (Eds.): ESWC 2015, LNCS 9088, pp. 403–417, 2015.
DOI: 10.1007/978-3-319-18818-8_25

semantic labeling function from a set of sources that have been manually labeled. When presented with a new source, the learned semantic labeling function can automatically assign the semantic labels to each attribute of the new source.

We are interested in mapping diverse data sources with different schemas to a common ontology. Taheriyan et al. [14] explain that this involves two steps - assigning semantic labels (class or data property) from the ontology to each source attribute and determining the relationships between the labelled attributes using ontology properties. Our work focuses on the first step of learning the semantic labeling function from the data. To learn the mapping, we use the data rather than the attribute names, which can be quite cryptic as they are often abbreviated (e.g., fname rather than first-name). The challenge is that new sources rarely have the same set of values for an attribute as the sources that the system was trained on. Distinguishing numeric attributes is especially challenging. For example, *Humidity* and *ChanceOfSnow* are both percentages and are thus very similar.

The contribution of our work is a new algorithm for learning a semantic labeling function with the following properties:

- **Efficiency and Scalability**: evaluations show that our method is about 250 times faster than our previous method using Conditional Random Fields.
- **Coverage**: our method can effectively learn semantic labels for both text and numeric data and can handle noisy "mostly" numeric data where a fraction of values are not numbers.
- **Accuracy**: our comprehensive evaluation shows that our method improves the accuracy of competing approaches on a wide variety of sources.
- **Generality**: our method is ontology and schema agnostic and can learn a semantic labeling function with respect to any ontology or classification scheme that a user selects for their application.

We now formally define the problem of semantic labeling of data sources. A data source $s$ is defined as a collection of ordered pairs $< \{a\}, \{v_a\} >$ where $a$ denotes an attribute name (e.g. "Date of birth", "PIN Code" etc.) and $\{v_a\}$ denotes the set of data values corresponding to the attribute $a$ (e.g., if $a$ is "Date of birth", the set $\{v_a\}$ will have values like "02-10-1992", "Jan 1, 1950", etc.).

Input to our algorithm is a set of *labelled* data sources. Different data sources can have attributes that have different attribute names but map to the same semantic label. E.g., data source $s_1$ has an attribute "Population" and source $s_2$ has an attribute "Number of people" and both these attributes are assigned the same semantic label "populationTotal" from the given ontology. In our approach, the data values from these sources are normalized to a standard format. Multiple data sources are often mapped to the same ontology in many practical scenarios, e.g., museums map their data to a common cultural heritage ontology and universities map their data to a research networking ontology (e.g. vivoweb.org).

When we *combine* the labelled data sources, we get training data of the form $\{ (< \{a^1\}, \{v_i^1\} >, l^1), (< \{a^2\}, \{v_i^2\} >, l^2), \cdots, (< \{a^n\}, \{v_i^n\} >, l^n)\}$. Here, for each $j$, $\{a^j\}$ denotes the set of attribute names assigned to the semantic label $l^j$

and $\{v_i^j\}$ denotes union of the sets of corresponding data values. The goal is to learn the the *semantic labelling function* $\phi : <\{a\}, \{v_i\}> \rightarrow l$.

To assign a semantic label to an attribute in a new data source, we take an ordered pair $<\{a\}, \{v_a\}>$ and use the semantic labelling function $\phi$ to predict its semantic label.

The rest of the paper is structured as follows: In Sect. 2, we describe our approach to semantic labelling. We describe how we handle textual and numeric data differently and how we combine the two to provide a robust technique capable of handling noise. In Sect. 3, we survey related work. In Sect. 4, we present the results of our experiments. Finally, in Sect. 5, we describe the future enhancements to our approach and conclude.

## 2 Approach

This section describes our approach for learning to label source attributes with semantic types using data sources that have already been aligned to an ontology. The training data consists of a set of semantic labels and each semantic label has a set of data values $v_i's$ and attribute names $a's$ associated with it. Our approach takes a holistic view by using techniques that capture characteristic properties associated with each semantic label as a whole rather than features from individual values. Given a new set of data values, the goal is to predict the top $k$ candidate semantic labels along with confidence scores.

### 2.1 Textual Data

We define a *textual semantic label* as a semantic label associated with textual data values (e.g. title of a painting, department name, etc.). In our approach, the set of data values associated with each textual semantic label $\{v_i\}$ in the training data is treated as a *document*. Similarly, at prediction time, the new set of data values is treated as a *query* document.

We index the training documents to improve query time efficiency. Data values are first tokenized by space and punctuation, then normalized and then indexed. Normalizations include removal of blank spaces, stemming, removal of common stop words, etc. Each document has a vector space model representation where each dimension corresponds to a unigram token from the vocabulary of tokens extracted. We used Apache Lucene[1] for indexing and searching of documents.

The weight assigned to a term in a document vector is the product of its *term frequency* (TF) and *inverse document frequency* (IDF), called TF-IDF. For each term $t$ in the document (or query) $x$, *term frequency* (TF) of $t$ in $x$ measures the number of occurrences of $t$ in $x$ and *inverse document frequency* (IDF) of $t$ measures the inverse of the number of documents containing term $t$.

Remember that each training document in the index corresponds to a distinct semantic label. In order to suggest the top $k$ candidate semantic labels for the set

[1] Apache Lucene: http://lucene.apache.org/core/.

of new data values at prediction time, we rank semantic labels in decreasing order of the cosine of the angle between the query document vector and each training document vector. The confidence score associated with a predicted semantic label is the corresponding *cosine similarity* between the documents' vectors.

The cosine similarity for a query document $q$ and a training document $d$ is

$$sim(q, d) = \frac{V(q) \times V(d)}{|V(q)| \times |V(d)|} \tag{1}$$

where $V(q)$ and $V(d)$ are the corresponding vector space model representations.

The idea behind using this approach stems from the fact that each semantic label has a characteristic set of tokens associated with it that can collectively help in identifying the correct semantic label. For example, if the data is about *dimensions of a painting*, data values typically look like "28 in. x 30 in." and hence, the presence of tokens like *x* and *in* strongly characterize this semantic label.

We call this approach the *TF-IDF-based cosine-similarity approach*. Though it seems quite simple, it results in higher prediction accuracy in terms of the mean reciprocal rank [3] and is extremely fast (low query time due to indexing) compared to existing approaches that extract features from each data value.

We also tried another similar approach in which the weight we assign to a term in a document vector is 1 if the term occurs in the document and 0 otherwise. Here, we rank semantic labels in decreasing order of the *Jaccard similarity* between the query document vector and the training document vector (corresponding to a semantic label). However, the TF-IDF cosine similarity approach proved to work better since the non-binary term weights are more informative and allows for a continuous degree of similarity between queries and documents.

### 2.2 Numeric Data

If the data values associated with a semantic label are numeric, instead of the TF-IDF-based approach, we analyse the distribution of numeric values corresponding to a semantic label. This arises from the simple intuition that the distribution of values in each semantic type is different. For example, the distribution of weights is likely to be different from the distribution of temperatures. In order to measure the similarity between distributions, we use *statistical hypothesis testing*.

The key output of statistical hypothesis testing used in our approach is the $p$-value. The $p$-value helps determine the statistical significance of the results of the hypothesis testing and is the probability of obtaining a test statistic at least as extreme as the one obtained using the sample data, assuming that the null hypothesis is true. Irrespective of the actual statistical hypothesis test used, the underlying idea is the same. The null hypothesis we are testing is that the two groups of data values are drawn from the same population (semantic label). A low $p$-value provides strong evidence against the null hypothesis while a large $p$-value provides weak evidence against the null hypothesis.

The training data consists of a set of numeric semantic labels and each semantic label has a sample of numeric data values. At prediction time, given a new set

of numeric data values (query sample), we perform statistical hypothesis tests between the query sample and each sample in the training data corresponding to a distinct semantic label. We rank the semantic labels in descending order of the $p$-values returned by the statistical hypothesis tests performed and suggest the top $k$ candidate semantic labels with the confidence scores as corresponding $p$-values.

We considered Welch's $t$-test [6] as our statistical hypothesis test. Given two samples of data, the $t$ statistic is defined by:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{2}$$

where $\bar{X}_i$, $s_i^2$ and $N_i$ are the sample mean, sample variance and sample size of the $i^{th}$ sample respectively. Welch's $t$-test does not assume that both samples of data have the same standard distribution. Once the $t$ statistic is calculated, it uses the $t$ distribution to test the null hypothesis that the two population means are equal (though the population variances may differ).

The problem with Welch's $t$-test is that it looks only at the *mean* of the population and not the complete distribution and hence does not match our need to test that the samples are drawn from the same distribution. Moreover, Welch's $t$-test expects the sample and population data to be approximately normal and expects the samples to have a similar number of data points. Most of the time, our problems fail to meet these expectations. To overcome this issue, we applied non-parametric tests to compare two samples of data.

We considered Mann-Whitney's U test [6], a non-parametric test of the null hypothesis that the two samples have the same distribution. It is more efficient than the $t$-test on non-normal distributions and does not expect the samples to have a similar number of data points. This test 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. If you assume that the two samples are drawn from distributions with the same shape, then it can be viewed as a comparison of the medians of the two samples.

We also considered the two-sample Kolmogorov - Smirnov (KS) Test [6], a non-parametric test that tests if the two samples are drawn from the same distribution by comparing the cumulative distribution functions (CDF) of the two samples. Similar to the Mann-Whitney test, it does not assume normal distributions of the population and works well on samples with unequal sizes.

The KS test computes the $D$ statistic which is the maximum vertical difference between the CDFs of the two samples and is given by

$$D_{N_1,N_2} = \sup_x |F_{1,N_1}(x) - F_{2,N_2}(x)| \tag{3}$$

where $F_{1,N_1}$ and $F_{2,N_2}$ are the cumulative distribution functions of sample 1 and sample 2 respectively. The $p$-value associated with the KS test determines the probability that the cumulative distribution functions of two samples that are randomly sampled from the same population are as far apart as observed with respect to the $D$ statistic.

The KS test is slightly more powerful than the Mann-Whitney's U test in the sense that it cares only about the relative distribution of the data and the result does not change due to transformations applied to the data. Also, the KS test is more sensitive to differences in the shape of the distribution, variance, and median, while the Mann-Whitney's U test is more sensitive to changes in the median. The non-parametric Wilcoxon signed-rank test is intended for paired variates and hence is not applicable in our case of independent attribute values. Our experiments on numeric data show that the Kolmogorov-Smirnov test achieves the highest label prediction accuracy of the various statistical hypothesis tests.

### 2.3 Overall Approach

We now present our overall approach (called *SemanticTyper*) combining the approaches to textual and numeric data. For textual data, we use the *TF-IDF*-based approach and for numeric data, we use the *Kolmogorov-Smirnov* (KS) statistical hypothesis test.

Data sources are often noisy and contain attributes with a mixture of numeric and text data. It is challenging to decide whether it is actually a numeric column and the text values are noise (e.g., years with noise such as "1999–2000") or it is a column of textual data (e.g., database identifiers). The challenge is to determine a threshold for the amount of noise allowed in a numeric column.

In order to resolve this, we adopted the rule that in the training data, if for a semantic label the fraction of pure numeric data values is below 60 %, it is trained as textual data (and hence indexed as document). If the fraction of numeric values is above 80 %, it is trained as purely numeric data (its distribution is extracted to be used in KS test) after discarding textual data values. In the other case (if the fraction is between 60 % and 80 %), the data is trained as both textual and numeric data (it is both indexed as a document and its distribution is extracted to be used in KS test).

At the time of prediction, given a new set of data values, we again calculate the fraction of numeric values. If it is greater than 70 %, it is tested as numeric data (textual data values are discarded). Else, it is tested as textual data. The above numbers (60 %,70 %,80 %) were arrived at empirically by running a coarse grid over these values by varying them in steps of 5 % and choosing the values that resulted in highest average label prediction accuracy.

During one of the experiments, we observed that while training, the fraction of numeric data values corresponding to the "Postal Code" semantic label was 71 % and hence it was trained as both textual and numeric data. During prediction, the fraction of numeric data values was 50 % and was hence was tested as textual data. The TF-IDF-based approach was hence used and was successful in predicting the correct semantic label as the first candidate suggestion. This clearly illustrates the strength of our approach in handling noisy data.

## 3 Related Work

Goel et al. [5] describe an approach that uses a supervised machine learning technique based on Conditional Random Fields (CRF) for semantic labelling of data sources. They extract features from the data values after tokenizing and building a CRF graphical model to represent the latent structure of the data sources, such as the dependency between field labels and their token labels, dependency between neighboring tokens within a field, and dependency between labels of neighboring fields. They assign semantic labels to all fields in a tuple (corresponding to a row in the data source) and then combine the labels of the fields in a particular source attribute to assign a label to the attribute. However, there is a tradeoff between the amount of latent structure exploited and corresponding training time to generate the CRF models.

Limaye et al. [8] work on the problem of annotating tables on the Web with entity, type, and relationship labels. They propose a probabilistic graphical model to label table cells with *entities*, table columns with *types*, and pairs of table columns with *binary relations* simultaneously rather than making the labelling decisions separately for each. The task of assigning semantic labels to columns is achieved using two feature functions (among 5 in total) - one that looks at the dependency between the type of column and the entity of entries in that column and the other that looks at the dependency between the type of column and the column header text using textual similarity measures. Mulwad et al. [9] assigns candidate labels for each cell value using Wikitology, similar to Limaye's work in using a probabilistic graphical model to assign labels to individual cells.

The approaches described above rely on training a probabilistic graphical model to annotate columns with semantic types. They analyze entries in the column separately and do not use any statistical measures to extract characteristic properties of the column data as a whole. Further, training probabilistic graphical models is not scalable as the number of semantic labels in the ontology increases due to explosion of the search space. Unlike in a named entity extraction setting, dependency between labels of adjacent source attributes (used in [5]) is not of use in semantic labeling of data sources since the order of attributes in a data source is not consistent enough to improve the accuracy of the labelling.

Venetis et al. [15] present an approach to annotate tables on the Web by leveraging resources already on the Web. They extract an *isA database* from the Web that is of the form (instance, class) and subsequently, label a particular column with a particular class label if a substantial fraction of the cells in that column are labelled with that class label in the *isA database*. They look for explicit matches for cell contents from a column in the *isA database* to assign labels to the table cells individually and then use a maximum likelihood approach to predict a semantic label for the column.

Syed et al. [13] exploit a web of semantic data for interpreting tables. They use the table headings (whenever available) and the values stored in the table cells to infer a semantic model that can be further used to generate linked data. This is achieved through the development of Wikitology - a hybrid knowledge

base of information extracted from Wikipedia and RDF data from DBpedia and other Linked Data sources.

An important aspect of the work by both Venetis et al. and Syed et al. is that they exploit a huge amount of data extracted from various sources. While having more data can be useful, it also restricts the approach to only those domains and ontologies where there is a large amount of extracted data. If we have a user defined ontology, it can be difficult to use the models from a general source, such as DBpedia. This is taken care in our approach where we learn the semantic labelling function from sources previously labeled using a given ontology. Sequeda et al. [11] address the problem of mapping relational tables to RDF, but generate IRIs based on predefined rules and do not learn mappings to labels in an existing ontology as we do.

A lot of work has been done in the related areas of schema and ontology matching [2,4,7,10]. Schema matching takes two schemas as input and produces a mapping between semantically identical attributes. Schema and ontology matching can be viewed as the combination of semantic typing and relationship mapping and this paper focuses on the former. Stonebraker et al. [12] developed an approach to schema matching that uses a collection of four different *experts* whose results are combined to generate mappings between attributes. One of their experts uses TF-IDF based cosine similarity to compare columns of textual data and another uses the Welch's $t$-test to compare columns of numeric data. Our work, which draws on some of these ideas, formulated an overall combined approach which is highly scalable, applied it to the problem of semantic typing, performed detailed experiments and analysis to come up with a better performing statistical test (Kolmogorov-Smirnov), and demonstrated the effectiveness of the approach on a diverse range of datasets.

## 4 Evaluation

For our experiments, we used datasets from multiple domains: *museum*, *city*, *weather*, *phone directory* and *flight status*. There are three types of experiments based on the nature of semantic labels to be assigned in the data sources: purely textual, purely numeric, and mixture of textual and numeric labels. The datasets and code used in our experiments have been published online[2].

### 4.1 Data Sets

For evaluating our approach on *purely textual* labels, we used data from the *museum* domain consisting of 29 data sources in diverse formats from various art museums in the U.S. Semantic labels were assigned to the attributes in these data sources manually to the Europeana Data Model, an ontology of cultural heritage data.[3]

[2] https://github.com/usc-isi-i2/eswc-2015-semantic-typing.git.
[3] https://joinup.ec.europa.eu/catalogue/distribution/europeana-data-model-primer.

For evaluating our approach on collection of *purely numeric* labels, we identified 30 numeric data properties from the *City* class in DBpedia and extracted these properties for various cities in the world. Most of the data properties possess more than 17,000 data values. We split the data associated with each semantic label into 10 partitions and manually synthesized 10 data sources by combining one partition from each semantic label to create one data source.

For evaluating our overall approach on a mixture of textual and numeric labels, we used 52 data properties from the *City* class from DBpedia, 30 of which are the ones used in the numeric approach and the remaining 22 data properties contain textual data values. The interesting aspect of the data collected from DBpedia is that it is noisy in the sense that even semantic labels, which are supposed to contain numeric data values, often contain textual values since the data is often authored on Wikipedia by a diverse group of people. This is where our overall approach is effective in handling noise.

We also evaluated our overall approach on the *weather*, *phone directory*, and *flight status* domains, which contain closely related data extracted from separate Web sites and consist of a diverse mixture of textual and numeric semantic labels. The datasets corresponding to the above domains were used in the experiments of Ambite et al. [1].

### 4.2 Experimental Setup

As already explained, we are not only interested in the top-1 prediction but in the top-$k$ predictions due to inherent similarity in many semantic labels. In our experiments, we took the value of $k$ to be 4 since experiments showed that the correct prediction was included 97 % of the time using our approach. In each experiment, the evaluation metrics of interest are mean reciprocal rank (MRR) [3] and average training time. MRR is useful because we are interested in the rank at which the correct semantic label is predicted among the 4 predictions provided by the system. It helps analyse the ranking of predictions made by any semantic labeling approach using a single measure rather than having to analyse top-1 to top-4 prediction accuracies separately, which is a cumbersome task.

Suppose the data set consists of $n$ sources $\{s_0, s_1, s_2..., s_{n-1}\}$. We perform $n$ runs and average the results of these $n$ runs to prevent cases in which the test data source is skewed in favor of our approach. In the $i^{th}$ run, we test our approach in labelling data source $s_i$. In order to understand how the number of labelled data sources in the training data affects our performance, in the $i^{th}$ run, we perform $n-1$ experiments. In the $j^{th}$ experiment ($j$ running from 1 to $n-1$) in the $i^{th}$ run, we train on $j$ data sources, specifically the $j$ subsequent data sources starting from $s_{i+1}$ (wrapping around 1 in a cyclical fashion), and test our approach on data source $s_i$. We obtain the MRR and training times for each experiment separately and average them over the $n$ runs. Thus, we essentially perform $n(n-1)$ experiments.

For example in the museum dataset containing 29 data sources, in the $1^{st}$ run, we test our approach on data source $s_1$ by performing 28 experiments. We train

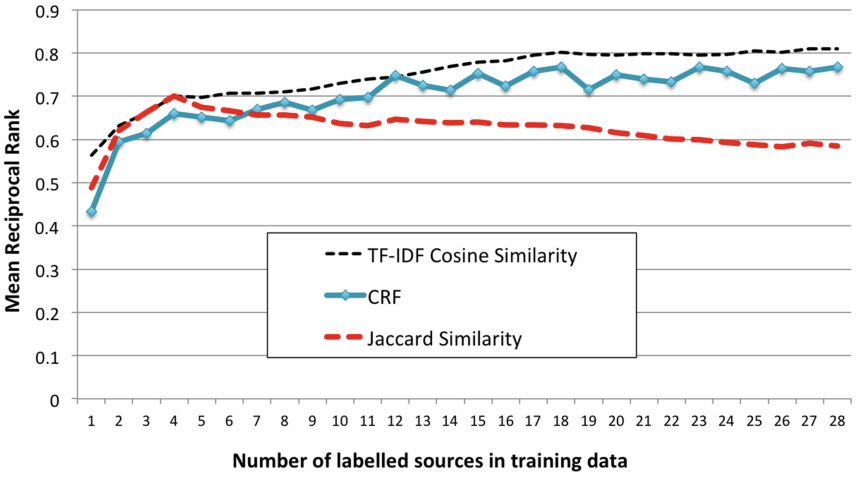


**Fig. 1.** Textual data from the museum domain

using only data source $s_2$ in experiment 1, data sources $\{s_2, s_3\}$ in experiment 2, $\cdots$ and data sources $\{s_2, s_3, \cdots, s_{29}\}$ in experiment 28.

There can be cases where a semantic label is absent in the training set but is present in the test set. In such a case, an ideal system is expected to identify this case and report that the semantic label in the test set is *absent* in the training set. If this is correctly identified, we assign a reciprocal rank of 1. Unlike previous approaches, the TF-IDF-based approach has the potential to identify this case if there is limited or no overlap in tokens between the test and training document. The KS-test gives a low $p$-value in such cases but identifying a suitable threshold for the KS-test will be addressed in future work.

### 4.3 Results: Textual Data

We used the 29 data sources from the museum domain to test our approach on textual data. Figure 1 shows the variation of MRR against the number of labelled data sources used in training for the three approaches on textual data: TF-IDF-based cosine similarity, a Jaccard-similarity-based approach (as explained in Sect. 3.1) and the Conditional Random Field (CRF)-based learning technique, which extracts features from data values individually [5].

As evident from Fig. 1, the TF-IDF-based cosine-similarity approach achieves higher MRR regardless of the number of labelled sources in the training data compared to the other two approaches. It reaches a maximum MRR of 0.81 when trained with 28 labelled data sources. It achieves an MRR of 0.56 when trained with 1 labelled data source, indicating that on the average, it predicts the correct semantic label in the second rank. The MRR steadily increases with the number of labelled data sources, attaining an MRR of 0.78 when trained with 16 labelled data sources itself. Beyond 16 data sources, we observe gradual increase

in the MRR for the TF-IDF-based approach. When trained with 16 labelled data sources, the CRF-based approach and Jaccard similarity reach MRRs of 0.72 and 0.63 respectively.

Each point on the x-axis corresponds to the number of labelled training sources and the corresponding ordinate value is the average of the MRRs obtained in $n$ experiments (each experiment corresponding to a distinct test data source). In order to ensure that the results we observed based on the average MRR are *statistically significant*, we ran a one-sided paired two-sample $t$-test between the TF-IDF-based approach and the other two approaches for the number of labelled training sources ranging from 1 to 28. We observe that for all points on the x-axis, we favour the alternative hypothesis that the population mean MRR for the TF-IDF-based approach is greater than that of either of the other two approaches with a 95 % confidence.

An interesting observation is that the Jaccard-similarity approach achieves an MRR comparable to the TF-IDF-based approach when the number of training data sources is less than 5, beyond which the performance of the Jaccard similarity approach starts declining monotonically and performs worse than the CRF-based technique thereafter. A possible explanation for this observation is that in the Jaccard similarity approach, the weights of tokens in the vector representation of documents representing semantic labels is binary indicating presence of terms. Hence, as the number of training data sources increases, a larger fraction of tokens in the vocabulary are present in each document and the binary weights are not informative enough resulting in the vector models of most documents giving close Jaccard similarities. Thus, the Jaccard similarity approach finds it more difficult to predict the correct semantic label at a higher rank as the number of training data sources increases.

### 4.4 Results: Numeric Data

We used the numeric data properties of the *City* class from DBpedia (divided into 10 data sources) to test our approach on numeric data. Figure 2 shows the variation of MRR against the number of training data sources used for approaches proposed by us in in Sect. 2.2, namely the Welch's $t$-test, the Mann-Whitney U test, and the Kolmogorov-Smirnov test. In addition to these three approaches, we also tested the TF-IDF-based approach (used for textual data) on this numeric data and compared the results with the existing CRF-based semantic labelling technique [5].

Figure 2 clearly shows that the Kolmogorov-Smirnov (KS)-test-based approach achieves much higher MRR than the other 4 approaches for all number of labelled data sources used in training. It reaches a maximum MRR of 0.879 when trained with 6 data sources and then saturates, retaining almost the same MRR for higher number of training data sources used. The maximum MRR scores achieved by other approaches is as follows: the Mann-Whitney U-test-based approach is 0.779, the $t$-test-based approach is 0.608, the TF-IDF-based approach is 0.715, and the CRF-based approach is 0.729.

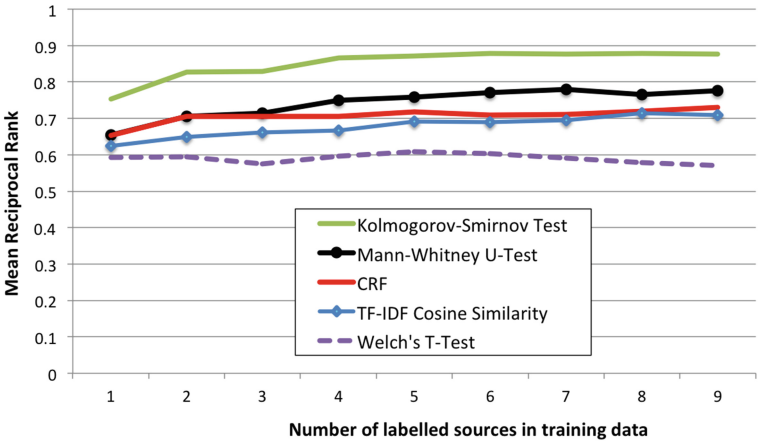


**Fig. 2.** Numeric data from DBPedia on the city domain

The interesting observation is that the Welch's $t$-test-based approach, which theoretically should perform better than the TF-IDF-based approach and the CRF-based approach on numeric data, actually does not perform better. This is possibly because the assumptions of the $t$-test that the distribution of the underlying population be Gaussian and that the two samples being compared have similar number of data points is violated. The curve for the $t$-test approach is decreasing with an increase in the number of training sources since the assumption of equal number of data points is violated to a greater extent as more data sources are included in the training.

We observe that the TF-IDF-based approach performs almost as well as the CRF-based technique, and that the KS-test and the Mann-Whitney-test-based approaches are clearly better suited to tackle numeric data with the KS-test-based approach achieved the highest MRR.

We ran a one-sided paired two-sample $t$-tests between the KS test and each of the other approaches to ensure the results are statistically significant. For each point on the x axis, we observed that we favour the alternative hypothesis that the population mean MRR for the KS test is greater than that of the other approaches with 95 % confidence.

### 4.5 Results: Overall Approach

First, we used the data extracted from DBpedia consisting of the 52 numeric & textual data properties of the *City* class to test our proposed overall approach (*SemanticTyper*). Figure 3(a) shows the variation of MRR with the number of training data sources. We compare our proposed overall approach against the CRF-based semantic labelling technique [5] and the TF-IDF-based approach.

As can be seen from the graph, *SemanticTyper* achieves an average increase of 0.09 and 0.12 in MRR compared to the CRF-based labelling technique and

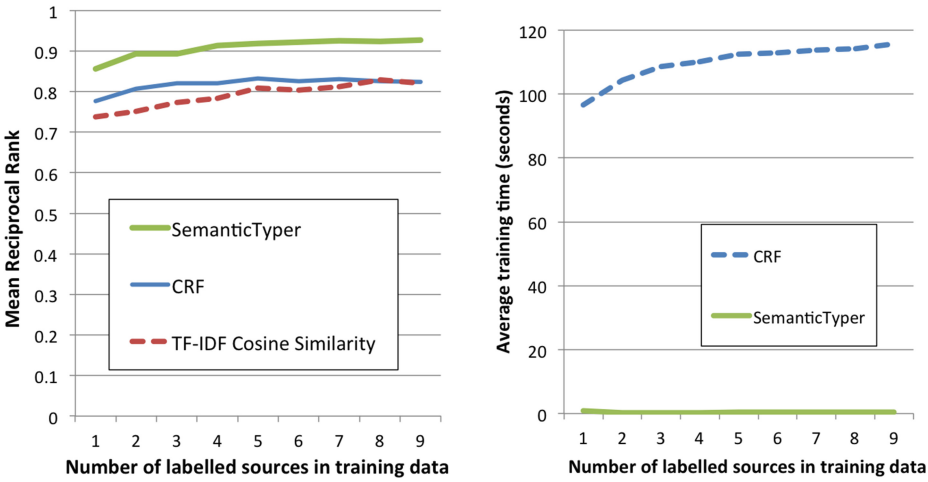


**Fig. 3.** (a) Mean reciprocal rank (b) Average training time for a mixture of textual and numeric data from DBPedia on the city domain

TF-IDF-based approach respectively. The maximum MRR achieved by *SemanticTyper* is 0.926, CRF-based technique is 0.823 and TF-IDF-based approach is 0.821.

For each point on the x-axis, we ran a one-sided two-sample $t$-tests. We reject the null hypothesis in favour of the alternative hypothesis that the population mean MRR achieved by *SemanticTyper* is greater than that of either of the other 2 approaches with 95 % confidence, showing that the differences are statistically significant.

We also compared our overall approach, (*SemanticTyper*), against the CRF-based approach and TF-IDF-based approach on the datasets from *weather*, *phone directory* and *flight status* domains [1]. In each of the 3 domains, *SemanticTyper* consistently achieved higher MRR as compared to CRF and TF-IDF-based approaches as we increased the number of labelled training sources (since the *phone directory* domain consists of mainly textual data, *SemanticTyper* reflects the TF-IDF-based approach). We present the maximum MRR achieved by the approaches in each domain in Table 1(we observe it occurs when training on all labelled data sources apart from the test source).

### 4.6 Training Time

For evaluation of the training time, we ran the CRF-based labelling technique [5] on the complete city dataset from DBpedia. As shown in Fig. 3(b), the training time increased linearly with the number of sources in the training data, starting from 96.6 s for 1 training data source to 115.6 s for 9 training data sources. The average training time was found to be 109.9 s.

**Table 1.** Maximum mean reciprocal rank on a mixture of textual and numeric data from the weather, flight status, and phone directory domains

| Domain | No.of sources | No.of textual labels/source | No.of numeric labels/source | Max. MRR | | |
|---|---|---|---|---|---|---|
| | | | | CRF | TF-IDF | SemTyper |
| Weather | 4 | 7 | 4 | 0.875 | 0.943 | 0.955 |
| Flight Status | 2 | 6 | 3 | 0.421 | 0.590 | 0.646 |
| Phone Directory | 3 | 8 | 1 | 0.704 | 0.831 | 0.831 |

On the other hand, for our proposed approach, the training time corresponds only to the time spent in indexing textual semantic labels using Apache Lucene and extracting the distribution from numeric semantic labels. Recall that for noisy semantic labels, we perform both of the above operations. The average training time using our approach is 0.45 s. Also, the training time remains almost constant even as more data sources are used for training. We do notice that there is a fixed header cost in training time in our approach due to connection establishment, I/O operations in indexing using Apache Lucene, though this is on the order of a tenth of a second.

Thus, we observe that the average training time of the CRF-based approach compared to our approach is about 250 times slower. This drastic drop in training time for our approach is possible because unlike the CRF-based approach, we are operating on the set of data values of a semantic label as a whole.

## 5 Conclusion and Future Work

This paper presents an integrated approach to the problem of mapping attributes of a data source to data properties defined in a domain ontology. Automating the semantic labeling process is crucial in constructing semantic descriptions of heterogeneous data sources prior to integrating them. Our approach called *SemanticTyper* is significantly different from approaches in past work in that we attempt to capture the distribution and hence characteristic properties of the data corresponding to a semantic label as a whole rather than extracting features from individual data values. It is evident from experimental results that our approach has much higher label prediction accuracy and is much more scalable in terms of training time than existing systems. Our approach makes no restrictions on the ontology from which data properties are to be assigned.

We plan to explore several directions in future work. First, the schema of a data source often contains metadata about attributes, such as attribute name, that can be helpful in assigning a semantic label to an attribute. For example, consider two semantic labels - *BirthDate* and *DeathDate*. The values of both semantic labels look very similar making it difficult to predict the correct semantic label as the first suggestion. But we can leverage the attribute name to differentiate between the two. Thus, we want to extend our approach to exploit

the information contained in attribute names to improve the labelling. Second, in case of numeric data, many times instead of continuous real valued attributes (like rainfall or elevation), we have attributes that take only a set of discrete values (like age in years, number of states, etc.). So, the performance can be enhanced further by identifying these cases and then using more suitable statistical tests (e.g., the Mann-Whitney test). Third, we plan to explore alternative tokenization and word n-gram representations as well.

## References


1. Ambite, J.L., Darbha, S., Goel, A., Knoblock, C.A., Lerman, K., Parundekar, R., Russ, T.: Automatically Constructing Semantic Web Services from Online Sources. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 17–32. Springer, Heidelberg (2009)
2. Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webtables: Exploring the power of tables on the web. Proc. VLDB Endow. **1**(1), 538–549 (2008)
3. Craswell, N.: Mean reciprocal rank. In: Liu, L., Zsu, M. (eds.) Encyclopedia of Database Systems, p. 1703. Springer, New York (2009)
4. Doan, A., Domingos, P., Halevy, A.: Learning to match schemas of data sources: a multistrategy approach. Mach. Learn. **50**(3), 279–301 (2003)
5. Goel, A., Knoblock, C.A., Lerman, K.: Exploiting structure within data for accurate labeling using conditional random fields. In: Proceedings of the 14th International Conference on Artificial Intelligence (ICAI) (2012)
6. Lehmann, E., Romano, J.: Testing Statistical Hypotheses. Springer Texts in Statistics. Springer, New York (2005)
7. Li, W.S., Clifton, C.: Semantic integration in heterogeneous databases using neural networks. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB). pp. 1–12 (1994)
8. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. PVLDB **3**(1), 1338–1347 (2010)
9. Mulwad, V., Finin, T., Joshi, A.: Semantic message passing for generating linked data from tables. In: Alani, H., et al. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 363–378. Springer, Heidelberg (2013)
10. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. SIGMOD Rec. **33**(4), 65–70 (2004)
11. Sequeda, J., Arenas, M., Miranker, D.P.: On directly mapping relational databases to RDF and OWL (extended version). CoRR abs/1202.3667 (2012)
12. Stonebraker, M., Bruckner, D., Ilyas, I., Beskales, G., Cherniack, M., Zdonik, S., Pagan, A., Xu, S.: Data curation at scale: the data tamer system. In: Proceedings of CIDR 2013 (2013)
13. Syed, Z., Finin, T., Mulwad, V., Joshi, A.: Exploiting a web of semantic data for interpreting tables. In: Proceedings of the Second Web Science Conference (2010)
14. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: A Scalable Approach to Learn Semantic Models of Structured Sources. In: Proceedings of the 8th IEEE International Conference on Semantic Computing (ICSC 2014) (2014)
15. Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. Proc. VLDB Endow. **4**(9), 528–538 (2011)