Making Sense of Entities and Quantities in Web Tables

Yusra Ibrahim[1] Mirek Riedewald[2] Gerhard Weikum[1]
[1]Max Planck Institute for Informatics Saarbrücken, Germany
[2]Northeastern University Boston, USA
yibrahim@mpi-inf.mpg.de, mirek@ccs.neu.edu, weikum@mpi-inf.mpg.de

ABSTRACT

HTML tables and spreadsheets on the Internet or in enterprise intranets often contain valuable information, but are created ad-hoc. As a result, they usually lack systematic names for column headers and clear vocabulary for cell values. This limits the re-use of such tables and creates a huge heterogeneity problem when comparing or aggregating multiple tables.

This paper aims to overcome this problem by automatically canonicalizing header names and cell values onto concepts, classes, entities and uniquely represented quantities registered in a knowledge base. To this end, we devise a probabilistic graphical model that captures coherence dependencies between cells in tables and candidate items in the space of concepts, entities and quantities. We give specific consideration to quantities which are mapped into a <measure, value, unit> triple over a taxonomy of physical (e.g. power consumption), monetary (e.g. revenue), temporal (e.g. date) and dimensionless (e.g. counts) measures. Our experiments with Web tables from diverse domains demonstrate the viability of our method and its benefits over baselines.

Keywords

Information Extraction, Semantic Annotation, Web Tables

1. INTRODUCTION

Motivation: The Web contains a wealth of structured but schema-free data in the form of HTML tables. These are manually created by knowledgeable users who want to share information—about music, food, car companies, renewable energy, traffic statistics etc. The advent of cloud-based editing and publishing tools (e.g., Google Sheets and Fusion Tables, Microsoft Excel Online) makes it even easier for users to post such content on the Internet. Likewise, a huge amount of tabular data exists within enterprise intranets, typically created with spreadsheet software.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
 ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983772

There is a great opportunity in comparing and combining multiple tables, towards analytic insight. However, these tables are typically created in an ad-hoc manner, to be shared with human users. And the absence of schemas and, even more, the diversity and potential inconsistency of terminologies among different tables (by different users) makes such data fusion steps impossible—if desired to be automatic—or extremely tedious—if carried out manually. The vocabulary mismatch across tables has several dimensions:

- Names in table headers typically denote *classes* (e.g., car model) or general *concepts* (e.g., CO2 emission), but are chosen ad-hoc on a per-table basis.
- Names in cells of the table body often denote individual *entities* (e.g., Tesla—the car maker, Musk—its CEO, Model S—one of Tesla's models), but the entity names are highly ambiguous.
- Other cells contain *quantities* such as financial measures (e.g., revenue in USD), physical measures (e.g., power in kW or energy consumption per 100 km in kWh), or plain numbers denoting ratios, temporal changes, ratings, etc. The encodings of values and their units can vary heavily across tables (e.g., $1 bn vs. 1000m USD for revenue, MPG vs. l/100 km for fuel consumption).

Example: Table 1 shows a typical example of a Web table, about environment-friendly cars in the U.S. If we want to compare this data to a table about these (and other) cars in Europe, we face huge heterogeneity issues regarding headers (Manufacturer vs. Company), entities in cells (Toyota Prius Eco vs. Prius Model 2016) and quantities in cells (MPGe vs. kWh/100km). This table is taken from a Wikipedia article; tables "from the wild", appearing in user's homepages or posted to social media, are an even greater challenge for proper interpretation.

Problem: In order to make better sense and enable reuse of ad-hoc tables, we want to *canonicalize* their headers and cells: link classes and concepts to a taxonomic catalog or simply to Wikipedia articles, disambiguate entity names onto uniquely identified entities registered in a knowledge base (KB), and map quantities into a complete and normalized representation with easily interpretable value and unit. This paper addresses this very problem, with emphasis on making sense of entities and quantities.

Prior work and its limitations: While entity linking (and so-called Wikification) from text to knowledge bases has received wide attention (see [8, 29, 30] and references given there), there is fairly little work on semantic annotation and linkage of Web tables. The first work on lifting Web

Table 1: Example Table: Green Vehicles Comparison

Vehicle	Manu-facturer	Class	GHG emissions (1)	Tailpipe emissions (g/mi of CO2)	EPA Fuel Economy combined (MPG)	Annual Fuel Cost
Toyota Prius phev	Toyota	Hybrid electric	61 lb CO2	133	95 MPGe ([29kWh +0.2 gal]/100 mi)	$600.00
Toyota Prius Eco All years, gasoline fuel	Toyota	Hybrid electric	51 lb CO2	178	50 (21.25 km/li)	$600.00
BMW i3 All years, all fuels	BMW	Electric car	54 lb CO2	0	124 MPGe (27 kWH/100mi)	<550$
Tesla Model S (60/85 kWh battery) 2013 Award	Tesla	Electric car	54 lb CO2	0	95 MPGe(35 kWH/100mi)	$700.00
Chevrolet Volt 2011 Award	GM	Plug-in Hybrid	61 lb CO2	81	98 MPGe (35kWh/100 mi)	$800.00
Bolloré Bluecar	Cecomp	Electric car	15.2 kg/100km	0	NA	80 €/mo

(1) measured per 100 mile.

tables to "first-class citizens" for search engines, by Halevy et al. [4], solely aimed at indexing for searchability and did not pursue any form of canonicalization. The seminal work on semantic linking for table cells by Sarawagi et al. [17] devised a probabilistic graphical model to map classes, relations and entities to a knowledge base. The resulting accuracy was in the order of 80%, and the method has high computational complexity. The recent work of Bhagavatula et al. [1] improved accuracy above 95%. None of these prior works considered quantities in tables.

Sarawagi et al. [27] addressed quantities, but focused on the specific tasks of searching with numerical values and extracting numerical relations from text [18]. Fully canonicalizing tables so that they can be compared and joined has been out-of-scope. The work by Chakrabarti et al. [33] developed table-to-table matching methods, based on entity augmentation, for the purpose of searching Web tables. In their follow-up work [34] the matching problem for a table corpus is extended to consider also numeric attributes. Although this work supports some form of comparing and combining tables, full canonicalization where all cells are mapped to semantic items in a knowledge base has not been considered.

Our approach is inspired by this prior work, but goes beyond their settings in several ways:

- We completely canonicalize entities and quantities (as well as classes and concepts).
- We exploit the textual context that usually surrounds a table and jointly link names and values from both table and text. For example, Table 1 appears in a page with the text shown in Figure 1. This allows us to harness semantic redundancy and richer features.
- We devise an efficient algorithm for fast processing of input tables, with the goal of supporting analysts in a responsive manner.

Our Approach: Probabilistic graphical models like Markov Random Fields (MRF's, aka. CRF's when inference is focused on conditional probabilities) are a most natural candidate for capturing the interdependencies in the potential linking targets of different entities, quantities, classes and concepts. Therefore, we conceptually start with a judiciously designed MRF model. To avoid the bottleneck of explicitly labeled training data, we employ distant supervision by drawing semantic relatedness weights from a knowledge base (with weights mined from Wikipedia links, unrelated to tables). We merely need a small set of annotated tables for tuning six hyper-parameters. To escape the high complexity of MRF/CRF inference (typically via MCMC sampling), we harness a theorem from [5] and construct a regular weighted graph from the MRF such that, under certain conditions, random walk (RW) algorithms closely approximate marginal probabilities for the MRF. Random walks can be implemented very efficiently. Working out the details of this MRF-to-RW reduction is one of the paper's key contributions. Our end-to-end solution for the table canonicalization problem is implemented in a system called *Equity* (Entity and quantity in tables).[1]

Contributions: Overall, this paper makes the following contributions:

- a comprehensive, distantly supervised MRF model for canonicalizing ad-hoc tables, handling classes, concepts, entities and quantities in both table cells and surrounding text;
- an efficient algorithm, based on random walks, for computing high-quality solutions;
- experimental results with a diverse set of Web tables that demonstrate the high accuracy of our method.

> The most efficient cars on the market are all electric cars. In fact, every electric car on the market is more efficient than even the most efficient conventional hybrid car (the Toyota Prius). Some of them are more than twice as efficient. As you scroll through the list below, note that the Prius has a MPG rating of 50 while Model S has a MPGe of 95. If you are not familiar with MPGe, it is a rating created by the EPA to determine the relative efficiency of an electric car compared to a gasoline car. MPGe is generally good for comparing electric cars to conventional gasmobiles and hybrids.

Figure 1: Text Snippet from context of Table 1

2. MODEL AND SYSTEM OVERVIEW

This section presents the formal problem definition and introduces important notation.

[1]More on this project, including experimental data, can be found at `www.mpi-inf.mpg.de/equity`.

2.1 Problem Input

The input to the Equity system is:

- A table T with $m+1$ rows, numbered $0 \ldots m$, and $n+1$ columns, numbered $0 \ldots n$; where row 0 is the *header row*. We use m_{ij} to refer to the mention in table cell (i, j), i.e., in row i and column j. The set of all mentions in table T is denoted as $\mathbf{M_T}$.
- A surrounding context with ν mentions m_k ($k = 1, \ldots, \nu$). The context is extracted from the web document's title and the table's surrounding text and caption. The set of all mentions in the context is denoted as $\mathbf{M_X}$.

We use $\mathbf{M} = \mathbf{M_T} \cup \mathbf{M_X}$ to refer to the set of all mentions in both table and context. We distinguish between two types of mentions. A *numerical mention* is a number, possibly accompanied by a unit. It represents a quantitative measure such as '27 kWh/mi'. All other mentions are referred to as *string mentions*. They are likely to refer to entities (e.g., GM or General Motors), classes (e.g., car manufacturers), or concepts (e.g., GHG emission).

Equity currently focuses on tables with the following common structural properties: The table header contains string mentions for classes and concepts. If the header is a class, then the non-header cells in the same column contain instances of the class, i.e., entities, as illustrated by the Manufacturer column in Table 1. If the header is a concept, then the non-header cells in the column contain quantitative measures, e.g., the Annual Fuel Cost in the example. As a result, Equity distinguishes between the following six *sorts* of mentions based on mention type and location: string/cell, string/header, string/context, numeric/cell, numeric/header, numeric/context.

Note that Equity can easily handle "transposed" tables where the header is not in row 0, but column 0, by working with the transpose of T.

2.2 Knowledge Base

The space of semantic targets to which we aim to map mentions in a table and its context is given by one or more knowledge bases (KB's). For individual *entities* and for *classes* (i.e., semantic types), we use Yago (`yago-knowledge.org`), which is one-to-one interlinked with Wikipedia for entities, and also connects Wikipedia categories with WordNet synsets for its extensive class hierarchy. For general *concepts*—abstractions that are neither classes nor entities (e.g., love, universe, number theory)—Yago is less suitable. In that case targets are Wikipedia articles (which do not have counterparts in Yago).

For quantities—the most challenging kind of targets—we constructed a new KB, called QKB, by importing information from `freebase.com` and restructuring it into the following organization. A *quantity* is a triple $\langle measure, value, unit \rangle$ where

- The *measure* is a name referring to a certain quantifiable aspect of an object or process (e.g., height of a building, power of a car's engine). We organize measures in a light-weight taxonomy (e.g., Revenue is placed under Monetary).
- The *value* is a numerical literal. Modifiers in mentions such as "M" for Mega or "mil" for Million, are factored into normalized values.
- The *unit* is a defined and widely used magnitude of a quantity, such as meter, kg, Watt, kWh, USD, EURO, etc.

For each quantity, QKB also keeps a set of alias names for the measure, and—when meaningful and possible—range bounds for the value, a regular expression for feasible surface forms of value and unit, and conversion rules for units. Our light-weight taxonomy covers physical, monetary and temporal measures and also unit-less numbers like ratios, rates, counts and scores.

2.3 Algorithm Objective and Output

We are interested in a (potentially partial) mapping Ψ from the set $\mathbf{M}$ of mentions to the set $\mathbf{S}$ of semantic items. Among all possible mappings, we aim to find one where (i) each mention is mapped to the "best" semantic target and (ii) the mapping is "consistent" with constraints implied by the table structure. These intuitive ideas will be formalized in the next sections. Clearly, there can be tension between the two goals that Equity has to address. For instance, given only the string "Tesla", the best match might be the KB entry for the person Nikola Tesla. However, in the context of Table 1, the location of this mention in a column of car manufacturers suggests a reference to the car maker.

Similar to the mentions, Equity also distinguishes between different *sorts* of semantic items: entities, classes, concepts and quantities.

Equity should produce the following mappings for sample mentions in Table 1:

- Mention m_{03}="GHG emission" is mapped to concept `Greenhouse_gas`.
- Mention m_{11}="Toyota" is mapped to entity `Toyota` (the company).
- Mention m_{25}="50" is mapped to the `physical` measure `EPA_Fuel_Economy` with value `50` and unit `Miles_Per_Gallon`.

If a mention has no proper item in the KB, Equity should map it to Null.

2.4 System Architecture

Figure 2 depicts the major components of the Equity system. We employ standard preprocessing for extracting a table and its context from a web page, and for shallow NLP such as part-of-speech tagging and coarse-grained typing of names—both via the Stanford NLP tools based on trained CRF's [9]. Note that the typing by the Stanford NER tagger merely produces labels like Person, Location, Organization, Date, Money and Misc, for text spans that likely denote entities or values of these kinds.

Equity first *detects mentions* in both table cells and context. This was partly done by the Stanford NER tagger already, but we apply additional regular expressions over token types to detect more mentions. Especially for quantities this is often decisive to ensure high recall.

To identify *semantic item candidates*, we run a light-weight form of Named Entity Disambiguation. This is a specifically configured variant of AIDA [13] using a simple popularity-based prior only and giving it only the mention itself (without any context). As the mention boundaries from the previous stage are not necessarily correct, we re-run AIDA with different choices of mention substrings as input. From the output, Equity keeps the top candidate entities based on

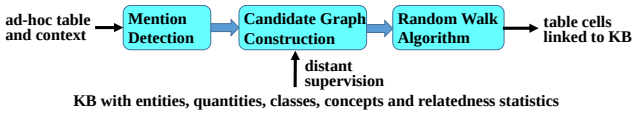

Figure 2: System Architecture of Equity

AIDA's confidence. For classes and concepts, which are not supported by AIDA, we perform simple string lookups against Yago and Wikipedia to generate candidates. For quantities, we match the input mentions against the alias names and, when applicable, regular expressions, for the measures in our QKB. The point of all this is to generate sufficiently many reasonable candidates. Hence this step does not have to be highly precise.

From the identified semantic items, we construct a *candidate graph*. This graph is constructed so as to approximate a full-fledged MRF with joint inference (see next section). Here we harness the KB for distant supervision, by using its precomputed relatedness scores as input for setting edge weights. These relatedness (aka. coherence) values are precomputed from Wikipedia links. We also apply some heuristic pruning when edge weights are negligibly small.

Finally, we perform random walks over the graph and identify, for each mention, the semantic item that has the highest stationary probability.

3. PROBABILISTIC GRAPHICAL MODEL

We start with a very natural approach and cast the problem of determining the semantic targets for the given table mentions into a probabilistic model with the following random variables:

- $X_{i,j}$: the observed surface form of a mention in table cell (i,j).
- X_k: the observed surface form of the k-th mention in the context of the table.
- $Y_{i,j}$: hidden variable for the semantic target corresponding to the mention in cell (i,j).
- Y_k: hidden variable for the semantic target corresponding to the k-th mention in the table context.

The X-variables range over the set of all possible strings, while the Y-variables range over the set of possible semantic targets S in the KB.

The desired mapping Ψ from the set of mentions $\mathbf{M}$ to the set of semantic items $\mathbf{S}$ is determined by inferring the Y-variables from the given X-variables. We propose to use Markov Random Fields (MRF), which have been successfully employed for identifying entities and types in tables [17] and for inference problems in image processing [2] due to their ability to efficiently represent spatial coherence relationships between pixels. Tables are similar to images in the sense that table structure implies implicit coherence relationships. For convenience, we introduce $\mathbf{H} = \langle X_{0,0}, \ldots, X_{m,n}, X_1, \ldots, X_\nu, Y_{0,0}, \ldots, Y_{m,n}, Y_1, \ldots, Y_\nu \rangle$ to refer to the vector of all random variables. Let H_i refer to the i-th entry in $\mathbf{H}$, $i \in \{0, 1, \ldots, 2(m+1)(n+1)+2\nu)\}$.

In addition to $\mathbf{H}$, the MRF is defined by a set of potential functions Φ, which capture relationships between the random variables. A pairwise relationship between H_u and H_v is modeled by function $\Phi_{u,v}$, which maps each pair of values from the domains of H_u and H_v, respectively, to a real number. For the sake of readability, we will also use $\Phi(H_u, H_v)$ to refer to $\Phi_{u,v}$. Similarly, the relationship between three random variables H_u, H_v, and H_w would be modeled by potential function $\Phi_{u,v,w}$, mapping a 3-tuple of values from the respective domains of the three random variables to a real number, and so on. Let E_Φ denote the set of sub-vectors of $\mathbf{H}$ over which Φ is defined. Then the MRF represents joint probability distribution

$$\Pr(\mathbf{H}) = \frac{1}{Z} \prod_{e \in E_\Phi} \Phi_e,$$

where Z serves to scale the values so that they are true probabilities. A crucial design decision for an MRF is to determine E_Φ, i.e., which random variables to connect through a potential function.

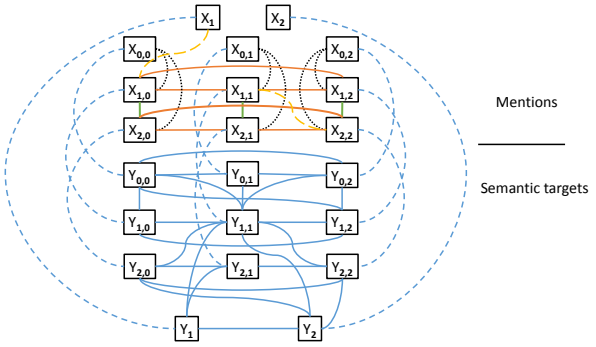

Figure 3: MRF model for a hypothetical table with 3 rows and 3 columns, whose context contains 2 mentions. For the Y-variables, not all edges of the fully connected subgraph are shown to reduce clutter.

3.1 Potential Functions

We limit all potential functions to be "pairwise". In addition to tractability of inference, this simplifies specifying the functions themselves. Intuitively, a pairwise potential function couples two random variables from $\mathbf{H}$ based on a relationship induced by table structure and content. The first family of potential functions captures the generic property that the surface form of the mention is closely related to the underlying semantic meaning.

Mention-target coupling: This dependency is represented by the blue dashed line in Figure 3. The corresponding family of potential functions is defined as $\phi_1(X_{i,j}, Y_{i,j})$, for all $0 \le i \le m$ and $0 \le j \le n$. Note that the individual functions in this family will differ depending on the sort of the mention and the semantic target (see Section 4.3). Similarly, the relationship between surface form and corresponding semantic target for the table context is captured by potential functions $\phi_2(X_k, Y_k)$, for $1 \le k \le \nu$. The next families of potential functions capture relationships induced by the table *structure*.

Header-cell coupling: This dependency is represented by the vertical black dotted line in Figure 3. It reflects that the header determines the information stored in a column. Equity captures this with a family of potential functions between the random variable for a header mention and the cell mentions in the same column: $\phi_3(X_{0,j}, X_{i,j})$, for $i > 0$.

Same-row coupling: This dependency is represented by

the horizontal orange solid lines in Figure 3. It models that the cells in a row contain data for a certain object represented by the row, hence are closely related. Formally this is encoded with potential functions for each pair of random variables for mentions in a row: $\phi_4(X_{i,j}, X_{i,k})$, for $i > 0$ and $j \neq k$.

Same-column coupling: This dependency is represented by the vertical green solid lines in Figure 3. Since all entries refer to the same "type" of information determined by the header, each cell's mention is closely related to the others in the same column. The corresponding family of potential functions is $\phi_5(X_{i,j}, X_{k,j})$, for $i > 0$, $k > 0$, $i \neq k$.

The last potential function families model global coherence properties.

Same-value coupling: This dependency is represented by the yellow long-dashed line in Figure 3. It captures the notion that given the specialized nature of a table and its context, occurrences of the same surface form are likely to refer to the same semantic target. This is modeled by potential functions that connect the random variables for all pairs of mentions that share the same surface form. One connects context mentions to table mentions: $\phi_6(X_{i,j}, X_k)$, for all $m_{ij} = m_k$. The other connects table mentions with each other: $\phi_7(X_{i,j}, X_{a,b})$, for all $m_{ij} = m_{ab}$. This could easily be relaxed to a coupling based on "similar", instead of identical, surface forms, e.g., to match 'MS Research' with 'Microsoft Research' or numbers such as '1.1 million' and '1,101,925'.

Candidate-candidate coupling: This dependency is represented by the blue solid line in Figure 3. It is motivated by the fact that all semantic targets for mentions in table and context should refer to a common topic, hence should be coherent. The corresponding families of potential functions are $\phi_8(Y_{i,j}, Y_{a,b})$, $\phi_9(Y_{i,j}, Y_k)$, and $\phi_{10}(Y_c, Y_k)$, for all $(i \neq a) \vee (j \neq b)$, $c \neq k$.

4. MRF AND RANDOM WALKS

Cohen [5] proved that marginal probabilities in an acyclic "pairwise"[2] MRF can be computed (almost) exactly through random walks followed by minimal post-processing. The proof includes the construction of an *ordinary-graph analog* of a given MRF, on which the random walks are performed. Even though the MRF for a table will usually contain cycles, Cohen's construction can still be applied to it. Cycles merely imply that equivalence between marginal probabilities in the MRF and the result of the random walk computation in the ordinary-graph analog might not hold any more. However, we argue—and confirm empirically—that the ordinary-graph analog still provides a good starting point for a random-walk based approach.

We now provide a summary of Cohen's approach, emphasizing intuition over detailed formalisms (for details, see [5]). Given an MRF, its ordinary-graph analog is constructed as follows:

- For each random variable V and each possible value $v \in V$, create a node n_v.
- Two nodes n_v and n_w, $v \in V$, $w \in W$, $V \neq W$, are connected by an undirected edge of weight $\phi_{V,W}(v, w)$, if and only if V and W are connected by an edge in the MRF.

[2]This is an MRF where all potential functions are defined over pairs of random variables.

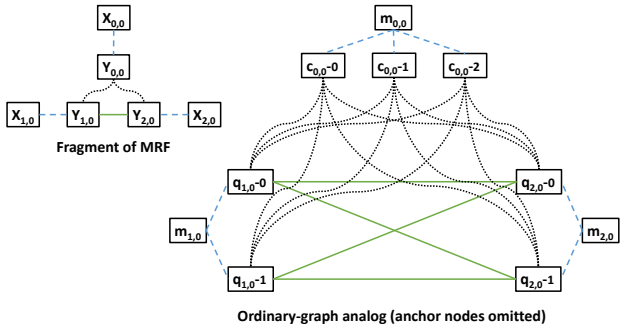

Figure 4: Ordinary-graph analog for a fragment of the MRF in Figure 3. For each connection of two random variables in the MRF, all values of the first are connected to all values of the second in the ordinary-graph analog. In the example, $Y_{0,0}$ ranges over three values; $Y_{1,0}$ and $Y_{2,0}$ each range over two possible values; and for the X-variables the only value is the mention given in the table.

- For each leaf variable L, i.e., variable that is connected to only one other variable in the MRF, there is an additional *anchor node* a_L. It is connected by an undirected edge of weight 1 to each node n_l, $l \in L$.
- There are no other nodes or edges.

Figure 4 illustrates this construction for a fragment of the MRF in Figure 3. To simplify notation, we will simply say "node v" to refer to "the node in the ordinary-graph analog that corresponds to value v".

In the case of an acyclic MRF, the marginal probability $\Pr(V = v)$ for random variable V can then be computed (almost) exactly using *Personalized PageRank* [12, 14]. This is a random walk algorithm with random restarts from a single designated start node. Let $\{v_1, v_2, \ldots, v_{|V|}\}$ be the set of possible values for random variable V. Cohen's proposed approach is to execute Personalized PageRank $|V|$ times, each time for a different $v_i \in V$ as the start node. Let α_s be the product of the PageRank values of all anchor nodes for the personalized PageRank execution with start node v_s. Then $\Pr(V = v_s)$ is obtained as $\frac{\alpha_s}{\sum_{i=1}^{|V|} \alpha_i}$.

4.1 Reduced Acyclic MRF

We create a reduced version of the MRF by removing edges until the remaining graphical model is acyclic. For clarity, we will refer to the MRF as defined in Section 3.1 as *full-MRF*; and to its acyclic version as *reduced-MRF*. More formally, reduced-MRF is the maximum spanning tree of the full-MRF, which Equity computes using Kruskal's algorithm. Ideally we would like to remove edges that have little impact on the marginal probabilities of the Y-variables. This impact is determined by the potential functions, which are difficult to learn due to lack of labeled training data. We therefore resort to a heuristic based on *priorities of edge types*.

Edge types are defined as in Section 3.1. Since surface form has a strong impact on the choice of semantic target, all *mention-target* edges have highest priority and will never be removed. For the other five edge types, Equity explores all $5! = 120$ possible sort orders of their priorities. Edges of the

same type are prioritized based on the sum of the individual weights of the corresponding edges in the ordinary-graph analog (see Section 4.3). The winner is selected based on performance on a small validation set of labeled tables.
This approach removes a large fraction of edges. We also explore an alternative that does not remove edges from full-MRF, but is computationally more expensive.

4.2 Modified Construction for Full-MRF

Depending on table structure and content, full-MRF might not have any leaf variables. This in turn implies that the corresponding ordinary-graph analog might have no anchor nodes, and therefore the computation using PageRank values of anchor nodes would be undefined. (Cohen did not encounter this problem as he only considered acyclic MRF, which are guaranteed to have leaf variables.) Even if there are leaf variables, as in the case of reduced-MRF, the meaning of the product of the PageRank values of the anchor nodes is not clear. Hence we have to re-think (1) the choice of start nodes for personalized PageRank and (2) how to use the PageRank values to select the best semantic target for each Y-variable.
Due to their unclear role for MRF with cycles, Equity works with a slightly modified ordinary-graph analog where *all anchor nodes and their adjacent edges are removed*. On the resulting graph, the best semantic target for a random variable $Y_{i,j}$ with candidate set $\{y_1, y_2, \ldots, y_{|Y_{i,j}|}\}$ is determined by executing personalized PageRank with start node m_{ij}, i.e., the node for the mention in table cell (i, j). Let β_k refer to the PageRank value of node y_k. Equity returns the candidate y_w with the largest β-value and estimates its probability of being the right answer as $\frac{\beta_w}{\sum_{k=1}^{|Y_{i,j}|} \beta_k}$. In general, the semantic candidate for table cell (i, j) is determined by (1) running personalized PageRank with starting node m_{ij} and (2) selecting that node $y_{i,j} \in Y_{i,j}$ with the highest PageRank among all semantic target candidates for cell (i, j).
The approach is motivated by the following intuition. Since mention node m_{ij} is directly connected to all semantic candidates for $Y_{i,j}$, starting there corresponds to a *prior*: greater edge weight results in correspondingly greater PageRank mass. The remainder of the graph then accounts for the effect of the table context. As closely related values of connected random variables will have edges of greater weight, the candidate that is well-connected to, and hence more coherent with, this context receives a greater PageRank value from those other edges.

4.3 Edge Weights

So far we have only specified the graph structure for personalized PageRank computation. Now we turn our attention to the edge weights. Instead of attempting to first learn the potential functions and then convert them to edge weights, we apply distant supervision using the KB and co-occurrence patterns in Wikipedia to determine those weights directly. Edge weights are defined by edge type. Each is the product of a type-specific weight vector and a feature vector, i.e., for an edge of type i connecting values $u \in U$ and $v \in V$ of random variables U and V, it is defined as

$$\mathbf{w}_i^T \mathbf{f}_i(u, v).$$

Due to the small number of labeled training cases, the number of parameters learned from these data has to be small. Hence for most edge types, the vectors are one-dimensional. We constrain all multi-dimensional weight vectors to only contain equal values. As a result, we only have a single *hyper-parameter* for each edge type. The hyper-parameters, each with a value between 0 and 1, are learned from a separately withheld and randomly selected validation set of labeled training tables. Equity performs a grid search to find the parameter combination with the best performance on the validation data. In the following, we introduce the edge weight features.
Mention-target edges connect a surface form to a semantic candidate item. For string mentions, we build on previous work and use features based on string similarity [17] and popularity statistics from Wikipedia links [13]. However, no previous work considered the relationship between surface form and semantic target for quantities. Depending on the sort of mention and semantic candidate, we use the following 1-dimensional feature vectors:

- m_{ij} is a string mention; $Y_{i,j} = c$, where c is a concept or class: Based on the intuition that surface form and semantic target are often textually similar for concepts and classes, we use the Jaro-Winkler distance between m_{ij} and c.
- m_{ij} is a string mention; $Y_{i,j} = e$, where e is an entity: We use the *popularity-based prior* that was found most effective for named entity disambiguation by Hoffart et al. [13]. For string mention m_{ij} and candidate entity e, it is defined as the number of Wikipedia links with anchor text m_{ij} that refer to e, divided by the total number of Wikipedia links with this anchor text.
- m_{ij} is a numerical mention; $Y_{i,j} = q$, where q is a quantity: We propose a new feature based on links in Wikipedia tables that refer to Wikipedia articles about units of measurement. Let m'_{ij} be the *unit component* of m_{ij}, i.e., the leftover after removing the magnitude. Then the feature is defined as the number of links in Wikipedia tables that have anchor text m'_{ij} and refer to a unit that is associated with quantity q, divided by the total number of links in Wikipedia tables with this anchor text and referring to any unit of measurement.

Header-cell edges: Given header and cell mentions m_{0j} and m_{ij}, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in header and non-header cell, respectively, of a column.
Same-row edges: Given same-row mentions m_{ij} and m_{ik}, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in any row.
Same-column edges: Given same-column mentions m_{ij} and m_{kj}, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in any column (excluding the header).
Same-value edges: We use a 1-dimensional feature vector with value equal to the Jaro-Winkler distance between the two surface forms.
Candidate-candidate edges: Equity uses a *relatedness* feature based on Wikipedia link co-occurrences. The relatedness of two semantic items is computed as the number of Wikipedia pages in which they co-occur, normalized so that the maximum value is equal to 1. In case of edges connecting a class and entity semantic targets in the same column, the weight of the edge is updated by the relation between the class and the entity's classes. That is, the edge

is weighted using a mixture of the candidates' relatedness and the classes overlap measures.

5. IMPLEMENTATION

5.1 Mention Recognition

For detecting mentions in tables and their contexts, we use the state-of-the-art Stanford NER tagger [9]. However, this tool was designed for natural-language sentences as input and shows low recall on tables. Hence we developed an extended mention recognition system as part of the Equity system. Our tool is centered on a rule-based classifier that uses regular expressions to detect occurrences of classes, concepts, entities and quantities in tables and their surrounding text. The major steps are as follows.

Classify columns: A column can be classified as numerical, textual or mixed. We run our regular expression classifier on each cell of the column, and then use majority voting.

Detect concepts and classes: We annotate the headers of numerical columns as mentions of concepts, and the headers of textual columns as mentions of classes. For mixed columns we base the decision on the majority of their cells.

Detect quantities: We use regular expressions to identify mentions of quantities, and to decompose them into value and unit.

Detect entities: We use the Stanford NLP parser to detect all possible noun phrases in a textual cell and mark them as entity mentions.

Enrich mentions: We further augment mentions in a cell with all sub-strings of the detected noun phrases. We repeatedly call the text-based entity-linking tool AIDA [13] with each sub-string as the sole input, to determine candidate entities. Then we filter the mention candidates, to select the maximum-length non-overlapping mentions with non-null candidates.

5.2 Candidate Search

Quantity Candidates: We start by finding candidates from our QKB for the unit part of the quantity mention. However, the unit is not always included in the cell. Therefore, we perform an expansion search for quantity candidates. We look for possible units, first in the cell, then in the column header, and eventually in the table context. Moreover, for quantities that do not have units, such as votes or scores, we use the column header to identify the measure.

Entity Candidates: We use the AIDA web service[3], to retrieve a set of candidate entities for each mention. The input is a set of possible mentions, and the output is a set of top-k candidate entities based on a simple popularity prior.

Class Candidates: We use Locality Sensitive Hashing (LSH) to retrieve candidate classes for mentions; then we filter them based on Jaro-Winkler distance between class name and mention.

Concept Candidates: Similar to the previous case, we use LSH followed by a filtering step using Jaro-Winkler distance. Furthermore, we add candidate measures from the QKB as candidates, as some column headers have labels like frequency, width or height. We ensure that the candidate units for the column cells are compatible with the candidate measure for the column header when we perform the final inference over the graph.

[3]https://gate.d5.mpi-inf.mpg.de/aida/service/disambiguate

5.3 Random Walk Algorithm

As explained in section 4, we construct 2 types of graphs: one for the full-MRF and one for the reduced-MRF. We rescale all edge weights by multiplying them with the hyper-parameters for the respective edge type. We use the power-iteration technique to compute the stationary vector of random walks with restart on the graph as described in Section 4. We check convergence based on the relative ordering of the semantic items, following [12].

6. EXPERIMENTAL RESULTS

We evaluated the effectiveness of the Equity system on a systematically sampled and fully annotated collection of 69 Web tables (with context). Equity is also compared to previous work on three larger collections with up to 6,085 tables.

6.1 Setup

Dataset: We build a corpus of web tables from two different sources: the Google Tables API[4] and the Wikipedia tables corpus from [1][5]. Note that fully annotating a table with the ground-truth for all mentions is a labor-intensive task requiring specialist knowledge. Hence we opted for annotators from our lab and aimed for a sampled and relatively small, but fully annotated collection. We wanted to cover a variety of domains: environment, finance, sport, health and politics. To get this diversity, we used a handful of keywords per domain to search for tables from the two sources and then randomly sampled medium-sized tables from the search results. In total, we obtained 69 tables this way: 63 from Wikipedia articles and 6 from various web sites. Table 2 shows statistics about the test dataset. We will make this annotated collection publicly available for other researchers.

Table 2: Statistics for Test Data Collection

Average per Table	Various Websites	Wikpedia
# rows	13.57	10.86
# columns	5.00	6.00
# numerical columns	1.57	3.02
# entity mentions	17.38	30.28
# quantity mentions	23.34	29.29
# class & concept mentions	2.28	4.29

Hyper-Parameter Tuning: We used a withheld set of 7 tables (disjoint from the test data) from a variety of domains (health, finance, etc.), in order to tune the hyper-parameters of Equity: six weights for different kinds of edges (see Section 4.3). We performed a grid search over 1000 combinations to obtain the best hyper-parameters for the full-MRF model and, separately, for the reduced-MRF.

6.2 Results

We report and discuss the effectiveness of Equity for mention detection and for mention linking. Our performance measure is precision, micro-averaged over all mentions of all 62 tables of the test collection. The total number of mentions evaluated is 5,581.

[4]https://research.google.com/tables

[5]http://websail-fe.cs.northwestern.edu/TabEL/

Table 3: Micro-averaged Precision of Mention Detection

	# Mentions			Micro-avg. Precision %		
Type	Table	Cxt.	All	Table	Cxt.	All
class	109	0	109	62.4%	-	62.4
concept	284	0	284	70.1	-	70.1
date	160	165	325	100.0	97.0	98.5
entity	1628	0	1628	49.7	-	49.7
loc	221	188	409	98.2	94.7	96.6
money	0	7	7	-	100.0	100.0
org	116	225	341	85.3	60.9	69.2
percent	19	35	54	100.0	100.0	100.0
person	86	55	141	89.5	56.4	76.6
quantity	2011	272	2283	82.7	58.1	79.8
Total	4634	947	5581	71.5	74.6	72.0

6.2.1 Mention Detection

Table 3 shows the precision of the mention recognition stage, broken down into mentions in tables and mentions in the contexts (Cxt.) and the total over both.

The Stanford NER Tagger alone was able to detect 1,277 mentions (out of which 1,120 are correct mentions) of the following types: date, location, money, organization, percent and person—mainly in the context. The Equity mention detector additionally identified 4,304 mentions (out of which 2,898 are correct mentions) of the following types: class, concept, entity, and quantity. In total, our method discovered 5,581 mentions in tables and their contexts. The micro-averaged precision is about 72%. Table 3 breaks this down onto the different kinds of mentions. The weak points are mentions of classes and mentions of entities other than location, organization and person (i.e., the row "entity" in the table)—mostly products or other artifacts (e.g., movies). On the other hand, we achieve almost 80% precision for quantities, which is the main target of this paper.

6.2.2 Mention Linking

Table 4 gives the micro-averaged precision that Equity (in its reduced-MRF configuration) achieves for mapping mentions to semantic items in the KB. We consider only correctly recognized mentions here, as the errors from the previous stage of mention detection would lead to trivial follow-up errors. In total, we evaluated 4,018 mentions at this mention linking stage. Overall, we obtain around 92% precision in linking quantities, entities, classes and concepts. For entities alone we achieved 88%, and for all kinds of quantities 93%. Table 4 shows the break-down for the different kinds of input mentions. Here, the row "quantity" refers to all numeric mentions excluding those of type date and money. The latter two are mostly detected by the Stanford NER tagger, whereas most of the remaining quantity mentions are only detected by our method. The numbers show that the quantities detected by the Equity-specific method exhibit even higher precision for mention linking, around 94%. As dealing with quantities has been the main target of this work, we consider the observed performance as very good. For linking entity mentions, the precision is well above 90% for location, organization and person. Similar to the mention detection stage, the remaining kinds of entities—for example, products such as car models—are a somewhat weaker point. Precision for these is around 81% ("entity" row in Table 4).

Table 4: Micro-averaged Precision of Mention Linking, Considering all Mentions (Entities, Concepts, Classes, Quantities) in Table and Context

	# Mentions			Micro-avg. Precision %		
Type	Table	Cxt.	All	Table	Cxt.	All
class	68	0	68	82.4	-	82.4
concept	199	0	199	84.9	-	84.9
date	160	160	320	100.0	100.0	100.0
entity	809	0	809	81.0	-	81.0
loc	217	178	395	94.9	96.6	95.7
money	0	7	7	-	100.0	100.0
org	99	137	236	89.9	95.6	93.2
percent	19	35	54	100.0	100.0	100.0
person	77	31	108	98.7	96.8	98.1
quantity	1664	158	1822	97.5	60.8	94.3
Total	3312	706	4018	92.2	89.4	91.7

Table 5: Entity Linking for Different Datasets, Considering only Entity Mentions in Tables, but not in the Context.

	Micro-avg. Precision %			
Data Set	Equity full-MRF	Equity red-MRF	Limaye et al. [17]	TabEL [1]
web_manual	86.11	85.11	81.37	89.41
wiki_links	96.39	96.24	84.28	97.16
wiki_random	83.04	82.98	–	96.17
Equity corpus	84.11	85.36	–	–

6.2.3 Comparison with Other Systems

Although the specific focus of our work is on quantities in tables, we also performed comparisons to prior work on entity linking in tables, using various annotated datasets from these works. We compare two configurations of our Equity system against the systems proposed in [17] and [1], restricting all inputs to entity mentions in table cells (i.e., no context, no quantities).

Table 5 shows the results, for the following datasets, with results for baselines as reported in the literature:

- web_manual [17], a set of 371 web tables with a total of 9,239 mentions,
- wiki_links [17] with 6,085 Wikipedia tables containing a total of 131,807 mentions, and
- wiki_random [1] with 3,000 randomly selected Wikipedia tables and about 40,000 mentions.

Equity outperforms [17] on all datasets. In comparison to TabEL [1], Equity performs nearly as well on the larger wiki_links collection, which has many tables from prominent Wikipedia articles. On wiki_random, on the other hand, Equity is substantially outperformed by TabEL.

The reason is that this dataset contains many tables from the long tail of Wikipedia with lower curation quality. In particular, these tables contain a substantial fraction of misleading anchor texts. For example, the mention 'Oslo' appears with a link to 'Bislett_Stadium', and 'BMW' is linked to 'BMW_in_Formula_One'. A supervised learning method like TabEL can handle such peculiar instances better. Recall that Equity is designed for coping with quantities and entities together, as opposed to focusing on entities alone.

Table 6: Ablation Study on Mention Linking, Considering only Entity Mentions in Table and Context

	Micro-avg. Precision %		
	web_ manual	wiki_ links	Equity corpus
full-MRF	86.11	96.39	86.69
red-MRF	85.11	96.24	87.79
full-MRF w/o cand-cand	84.81	96.17	86.63
red-MRF w/o cand-cand	84.81	96.17	87.92
full-MRF w/o table-struct.	84.92	96.22	86.37
red-MRF w/o table-struct.	85.09	96.25	87.86
full-MRF w/o same-value	86.11	96.39	86.69
red-MRF w/o same-value	85.11	96.24	87.79

6.3 Ablation Study

To study the importance of the different edge types in the graph models, we performed an ablation study where we selectively disabled some of them in both full-MRF and reduced-MRF. Table 6 shows the results on mention linking, limited to entities because quantities are only annotated in the Equity corpus.

We observe that the reduced-MRF and the full-MRF have almost the same precision in all configurations. However, the reduced-MRF variant of Equity is much faster (see below). The results on leaving out specific types of edges show that our methods are robust. Missing certain cues affects the output quality only slightly. On the other hand, this also shows that the wiki_links corpus, the by far largest of the datasets, is a fairly easy test case. The other two corpora are rather small; hence there is no final conclusion yet on the importance of edge types.

6.4 Error Analysis

Many of the linking errors we observed are due to the absence of specific measures or units in our QKB, or caused by very ambiguous column headers. Examples for the latter are "$\eta(Observed)$" for measuring the thermal efficiency of a heat engine and "Nat." referring to nationalities with abbreviations of countries such as "GRE" (for Greece, presumably). We also observed cases where the column header gives misleading information such as "Density ($area/km^2$)" while the values in that column indicate population densities in $people/km^2$. Also, Equity sometimes misclassifies a column as numeric; an example is the column "Pollutant" with values like "CO2", "PM10" etc. Conversely, we occasionally miss out on a numeric column; an example is "Govt." with numbers referring to a country's governments at different periods.

6.5 Run-Time Analysis

We implemented the Equity system in Java using a Postgres database as a KB repository, and measured its run-time on a server with 4×4 Intel Xeon CPU E5-2667 v3 @ 3.20GHz cores, setting the maximum memory allocation pool for Equity to 40GB. The run-times for the reduced-MRF variant of Equity are 15 times faster than those for the full-MRF. Further analyzing the time spent in different components shows that the dominant factors are (i) SQL calls to fetch candidates and associated statistics from the KB and (ii) web service calls to obtain auxiliary information from AIDA. Discounting these components, which could be re-implemented in a much more light-weight manner, the time to process one table is about 2 seconds on average for the reduced-MRF variant of Equity. With some code tuning, this could be further optimized.

7. RELATED WORK

Entity Linking: There is ample work on detecting and disambiguating entities that appear in text documents; [29] is a recent survey on this topic. Some of the prominent approaches map to Wikipedia (e.g., [21, 24]), thus covering also classes and concepts, whereas others strictly focus on individual entities with DBpedia, Yago or Freebase as their point of reference (e.g., [13, 20]). The best-performing methods typically combine a variety of signals and techniques like pair-wise relatedness of entities [6, 23], refined context models [16], graph algorithms [13], and random walks [11]. The Equity system adopts some of these techniques, embedding them into its generalized framework for linking mentions in tables to both entities and quantities.

Quantity Extraction: Numeric attribute values and numeric expressions in natural-language text have been considered by work on information extraction and knowledge fusion [7, 18, 26, 28]. This line of research is related to our stage of quantity mention detection, but does not address the issue of canonicalizing quantities.

Table Search and Matching: Starting with the seminal work of [3, 4, 19], there is growing research on Web tables and spreadsheets, with the goal of searching table contents, matching tables against each other and inferring table header semantics [27, 31, 33, 34]. Linking table cells to a KB is of no or minor concern in these works. [27] and [34] deal with quantities in tables, using computational expensive techniques like MCMC inference for probabilistic graphical models.

Table Canonicalization: Closest to our work is the prior research of [17, 27, 22, 1]. [17], [22] and [1] pursued the same goal as our work, but did not consider quantities at all. Also, these methods use expensive inference algorithms and partly rely on extensive training data; our work avoids both of these potential bottlenecks. [25] addresses the linking of table headers and rows to concepts and entities in DBpedia, but focuses on small and narrow HTML tables.

[27] specifically addressed quantities in tables. It developed a grammar-based technique for column annotation and a supervised classifier for inferring units of columns with numeric values. However, this was driven by the task of searching a heterogeneous table corpus, without resolving the heterogeneity—thus leaving out the task of linkage to a comprehensive KB.

MRF/CRF and Random Walks: Our approach builds on insights from the work of W. Cohen [5] about the connections between MRF inference and random walk algorithms. Cohen has developed this further into a general framework for reasoning with random walks [32].

8. CONCLUSION AND FUTURE WORK

This paper addressed the task of fully canonicalizing mentions in ad-hoc tables and their surrounding contexts, by linking mentions of entities, classes, concepts and quantities to a knowledge base. To this end, we devised an MRF model, distantly supervised by relatedness measures from a KB, then derived a reduced acyclic MRF, and finally cast the

inference over this light-weight model into an efficient algorithm based on random walks over normal weighted graphs. Our experiments with a collection of Web and Wikipedia tables demonstrate that particularly the detection and linking of quantities—our main target—works very well. The reduced-MRF method achieves an overall linking precision of about 92%, and even 93% for quantity mentions. The Equity system developed in this paper is a first building block in our longer-term research towards making sense of Web tables and spreadsheets in enterprises. Future work includes joining, comparing and aggregating data over multiple tables, harnessing their canonicalization by Equity.

Acknowledgement

We would like to thank Chandra Bhagavatula and Doug Downey for making their data available to us, and the anonymous reviewers for their constructive feedback.

9. REFERENCES

[1] C. S. Bhagavatula, T. Noraset, D. Downey: TabEL: Entity Linking in Web Tables. ISWC 2015

[2] A. Blake, P. Kohli, C. Rother: Markov Random Fields for Vision and Image Processing. MIT Press 2011

[3] M. J. Cafarella, A. Y. Halevy, D. Z. Wang, E. Wu, Y. Zhang: WebTables: exploring the power of tables on the web. PVLDB 2008

[4] M. J. Cafarella, A. Y. Halevy, J. Madhavan: Structured data on the web. Commun. ACM 54(2): 72-79 (2011)

[5] W. W. Cohen; Graph Walks and Graphical Models. Technical Report CMU-ML-10-102, Carnegie-Mellon University 2010

[6] S. Cucerzan: Name entities made obvious: the participation in the ERD 2014 evaluation. ERD 2014

[7] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. KDD 2014

[8] ERD 2014: ACM International Workshop on Entity Recognition & Disambiguation, co-located with SIGIR 2014

[9] J. R. Finkel, T. Grenager, C. D. Manning: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. ACL 2005

[10] L. Galarraga, G. Heitz, K. Murphy, F. M. Suchanek: Canonicalizing Open Knowledge Bases. CIKM 2014

[11] Z. Guo, D. Barbosa: Robust Entity Linking via Random Walks. CIKM 2014

[12] T. H. Haveliwala: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Trans. Knowl. Data Eng. 2003

[13] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum: Robust Disambiguation of Named Entities in Text. EMNLP 2011

[14] A. N. Langville, C. D. Meyer: Google's PageRank and beyond - the science of search engine rankings. Princeton University Press 2006

[15] N. Lao, E. Minkov, W. W. Cohen: Learning Relational Features with Backward Random Walks. ACL 2015

[16] N. Lazic, A. Subramanya, M. Ringgaard, F. Pereira: Plato: A Selective Context Model for Entity Resolution. TACL 2015

[17] G. Limaye, S. Sarawagi, S. Chakrabarti: Annotating and Searching Web Tables Using Entities, Types and Relationships. PVLDB 2010

[18] A. Madaan, A. Mittal, Mausam, G. Ramakrishnan, S. Sarawagi: Numerical Relation Extraction with Minimal Supervision. AAAI 2016

[19] J. Madhavan, L. Afanasiev, L. Antova, A. Y. Halevy: Harnessing the Deep Web: Present and Future. CIDR 2009

[20] P. N. Mendes, M. Jakob, A. Garcua-Silva, C. Bizer: DBpedia spotlight: shedding light on the web of documents. I-SEMANTICS 2011

[21] D. N. Milne, I. H. Witten: Learning to link with wikipedia. CIKM 2008

[22] V. Mulwad, T. Finin, A. Joshi.: Semantic message passing for generating linked data from tables.: ISWC 2013.

[23] F. Piccinno, P. Ferragina: From TagME to WAT: a new entity annotator. ERD 2014

[24] L-A Ratinov, D. Roth, D. Downey, M. Anderson: Local and Global Algorithms for Disambiguation to Wikipedia. ACL 2011

[25] D. Ritze, O. Lehmberg, C. Bizer.: Matching html tables to dbpedia.: WIMS 2015

[26] S. Roy, T. Vieira, D. Roth: Reasoning about Quantities in Natural Language. TACL 2015

[27] S. Sarawagi, S. Chakrabarti: Open-domain quantity queries on web tables: annotation, response, and consensus models. KDD 2014

[28] T. Sellam, O. Alonso: Raimond: Quantitative Data Extraction from Twitter to Describe Events. ICWE 2015

[29] W. Shen, J. Wang, J. Han: Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. IEEE Trans. Knowl. Data Eng. 27(2): 443-460 2015

[30] R. Usbeck et al.: GERBIL: General Entity Annotator Benchmarking Framework. WWW 2015

[31] P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, C. Wu: Recovering Semantics of Tables on the Web. PVLDB 2011

[32] W. Y. Wang, K. Mazaitis, N. Lao, W. W. Cohen: Efficient inference and learning in a large knowledge base - Reasoning with extracted information using a locally groundable first-order probabilistic logic. Machine Learning 100(1): 101-126 (2015)

[33] M. Yakout, K. Ganjam, K. Chakrabarti, S. Chaudhuri: InfoGather: entity augmentation and attribute discovery by holistic matching with web tables. SIGMOD 2012

[34] M. Zhang, K. Chakrabarti: InfoGather+: semantic matching and annotation of numeric and time-varying attributes in web tables. SIGMOD 2013