Profiling the Potential of Web Tables for Augmenting Cross-domain Knowledge Bases

Dominique Ritze, Oliver Lehmberg, Yaser Oulabi, Christian Bizer
Data and Web Science Group, University of Mannheim,
B6, 26 68159 Mannheim, Germany
{dominique,oli,yaser,chris}@informatik.uni-mannheim.de

ABSTRACT

Cross-domain knowledge bases such as DBpedia, YAGO, or the Google Knowledge Graph have gained increasing attention over the last years and are starting to be deployed within various use cases. However, the content of such knowledge bases is far from being complete, far from always being correct, and suffers from deprecation (i.e. population numbers become outdated after some time). Hence, there are efforts to leverage various types of Web data to complement, update and extend such knowledge bases. A source of Web data that potentially provides a very wide coverage are millions of relational HTML tables that are found on the Web. The existing work on using data from Web tables to augment cross-domain knowledge bases reports only aggregated performance numbers. The actual content of the Web tables and the topical areas of the knowledge bases that can be complemented using the tables remain unclear. In this paper, we match a large, publicly available Web table corpus to the DBpedia knowledge base. Based on the matching results, we profile the potential of Web tables for augmenting different parts of cross-domain knowledge bases and report detailed statistics about classes, properties, and instances for which missing values can be filled using Web table data as evidence. In order to estimate the potential quality of the new values, we empirically examine the Local Closed World Assumption and use it to determine the maximal number of correct facts that an ideal data fusion strategy could generate. Using this as ground truth, we compare three data fusion strategies and conclude that knowledge-based trust outperforms PageRank- and voting-based fusion.

Categories and Subject Descriptors

[**Information systems**]: Data management systems—*Information integration*

Keywords

web tables; data profiling; knowledge base augmentation; slot filling; schema and data matching; data fusion

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883017.

1. INTRODUCTION

Cross-domain knowledge bases such as DBpedia, YAGO, or the Google Knowledge Graph are employed as background knowledge within a wide range of different applications including Web search, question answering, data integration, and entity linking. For these scenarios, the content of the knowledge bases should be as complete, correct, and up-to-date as possible. In order to fulfill such requirements, cross-domain knowledge bases should continuously be updated and extended using high-quality data from external sources.

Relational Web tables, i.e. data tables extracted from HTML pages [5], are an interesting source of external data for extending cross-domain knowledge bases as they cover a very wide range of topics and as there is potentially a large overlap in the table data that is published by different websites.

There is already a decent body of research on using Web tables [15, 28, 11] as well as other Web data sources [9, 14] to extend existing knowledge bases [15, 28, 9, 11] or user provided tables [25, 6, 14]. The problem with the existing approaches is that they are either evaluated on very small and thus not representative Web corpora or that they are evaluated on large Web corpora owned by search engine companies, which do not allow information about the content and coverage of their crawls to be published. This makes it impossible to generalize and scientifically verify the research results. Further, none of the existing publications answers the question which topical areas of the knowledge bases can be complemented using Web table data. For these reasons, we believe that a large publicly available corpus, such as the WDC Web Tables corpus[1], along with an in-depth profiling of its contents can greatly benefit the research community by serving as a common ground for the evaluation of knowledge base augmentation methods.

This paper reports about the results of matching 33 million Web tables from the WDC Web Tables Corpus to the DBpedia knowledge base [13]. Based on the matching results, we profile the potential of Web tables for augmenting different parts of the knowledge base and report detailed statistics about classes, properties, and instances where missing values can be filled using Web table data as evidence. In order to explore the degree of overlap between the Web tables, we group the matched facts by the described instance and property and are thus able to report the size distribution of the resulting groups of alternative values from different sources for specific facts. Recent work [9] proposes to apply

[1] `http://webdatacommons.org/webtables/`

a *Local Closed World Assumption* (LCWA) and use the content of an existing knowledge base to evaluate the accuracy of data values that are chosen by a data fusion heuristic. We empirically verify the Local Closed World Assumption and show that it is transferable to values that are missing in the target knowledge base. In order to establish a basis for comparing different data fusion strategies, we apply the Local Closed World Assumption to determine the maximal number of correct facts that an ideal data fusion strategy could generate. Using this as ground truth, we then compare three data fusion strategies and verify the claim from Dong et al. [10] that knowledge-based trust outperforms PageRank- and voting-based data fusion strategies. The contributions of this paper are:

1. An in-depth profiling of a publicly available Web tables corpus, providing insights into its topical contents.

2. The confirmation of the validity of the LCWA and an evaluation indicating that evaluation results obtained using the LCWA are transferable to data with no corresponding values.

3. A verification that knowledge-based trust outperforms PageRank- and voting-based data fusion strategies.

The paper is organized as follows: First, we introduce the WDC Web Tables Corpus in Section 2 and our reference knowledge base DBpedia in Section 3. Section 4 describes the matching techniques that are used and discusses statistics about class, property and instance correspondences that are created during the matching. Afterward, we analyze the overlap of the triples generated using these correspondences in Section 5. Section 6 verifies the Local Closed World Assumption, compares the different data fusion strategies, and provides statistics about the fused values. Section 7 discusses our findings in relation to existing work. Our conclusion are drawn in Section 8.

2. WEB TABLES

The WDC Web Tables Corpus is a large, public corpus of relational HTML tables. It has been extracted from the 2012 version of the CommonCrawl Web Corpus[2] which consists of 3.5 billion HTML pages originating from 43M different websites. Altogether, the 2012 version of the crawl contains over 11 billion HTML tables.

For building the WDC Web Tables Corpus, several heuristics were applied to distinguish between relational and layout tables: First, all tables containing other tables were excluded. Afterward, a classifier using layout and context features (similar to the features proposed by Wang et al. [22]) was used. As result of both steps, 147.6 million tables were classified as relational tables. This corresponds to 1.3% of all HTML tables in the crawl which is in line with the results of Cafarella et al. [7] who found 1.1% of all HTML tables in a Google crawl to contain relational data.

For our experiments, we only consider Web tables from the mostly English language top-level domains (TLDs) *com*, *org*, *net*, *eu*, and *uk*. The majority (83%) of these tables originate from *com*-domains (see Table 1). We further exclude all tables without an entity label column (see below) and tables with less than five rows or three columns. The resulting subset of the corpus consists of 33 403 411 tables.

We define the entity label column as the column that contains the names of the entities that are described in the table. Without such a column, we cannot determine the topical content of a table. To detect the entity label column, we apply the following heuristic: The entity label column must be of data type *string*, contain at least four characters and have the highest number of unique values in the table (in case of a tie, the left-most column is used). A detailed evaluation of this heuristic is provided by Ritze et al. [18].

Besides the data type *string*, we further detect the column data types *numeric* and *date* by applying about 100 manually defined regular expressions to all of a column's values. The final data type for the column is then decided by a majority vote.

Table 1 shows statistics about the number of columns, rows and values in total and per data type. Columns without obvious data types are excluded in the statistic. The number of values is approximated based on the data type of the columns and the corresponding number of rows. Most of the values are of data type *string,* followed by *numeric* values.

The tables in our corpus originate from 97 932 different websites. Here, we use the term website for each pay-level-domain (PLD), that is, the part of an URL's host that is paid for. Table 2 shows the most frequent PLDs and column headers (first non-empty row of a table). The most prominent PLD is *apple.com* (iTunes Music) while the other PLDs often refer to sport websites, e.g. *baseball-reference.com* or retailers such as *amazon.com*. The column headers give us a first impression about the topics of the Web tables. Frequently used headers are for example "5 star" and "price", indicating that the corpus contains a large amount of tables about products. Further, headers like "replies" or "latest post" point to the fact that the corpus contains data from blogs or forums. About 8.5% of all columns have an empty header.

Table 1: Characteristics of the Web Table Corpus

Tables per TLD

com	org	net	eu	uk	Σ
26.7M	3M	3M	216K	6K	33.4M

Columns, Rows and Values

	Numeric	Date	String	μ	Σ
Columns	46M	4M	86M	4.122	137M
Rows	-	-	-	21.499	716.6M
Values	995M	101M	1.9B	88.611	2.95B

3. REFERENCE KNOWLEDGE BASE

As reference knowledge base, we use DBpedia 2014[3]. Obviously, the results of our Web table profiling depend on the contents of our reference knowledge base as we can only find correspondences to classes, properties, and instances that exist in the knowledge base [12]. The DBpedia knowledge base describes 4 584 616 instances using 2 795 different properties and 685 classes. Table 3 shows frequent classes from the first three levels of the DBpedia class hierarchy ('+': second level, '|-': third level).

[2] `http://commoncrawl.org/`

[3] `http://wiki.dbpedia.org/data-set-2014`

Table 2: Most Frequent PLDs and Column Headers

PLDs	Tables	Headers	Tables
apple.com	50 910	no header	14 495 456
patrickoborn.com	45 500	5 star:	2 402 376
baseball-reference.com	25 647	name	1 813 064
latestf1news.com	17 726	price	1 771 361
nascar.com	17 465	date	1 603 938
amazon.com	16 551	amazon price	1 178 559
baseball prospectus.com	16 244	formats	1 066 836
wikipedia.org	13 993	title	9 132 60
inkjetsuperstore.com	12 282	time	856 401
flightmemory.com	8 044	description	773 883
sportfanatic.net	7 596	size	692 251
tennisguru.net	7 504	replies	605 075
windshieldguy.com	7 305	used from	589 278
donberg-electronique.com	6 734	new from	589 259
citytowninfo.com	6 293	year	579 726
juggle.com	5 752	location	546 856
deadline.com	5 274	album	526 375
blogspot.com	4 762	type	501 747
7digital.com	4 462	latest post	421 737
electronic-spare-parts.com	4 421	discussion	412 672

Table 3: Selected Frequent DBpedia Classes

DBpedia Class	Instances
+ Person	1 445 104
|- Athlete	280 976
+ Organisation	241 286
|- EducationalInstitution	35 190
Place	725 546
|- Country	1 694
Work	396 046
+ MusicalWork	162 397
+ Software	25 649
Species	283 341

4. TABLE MATCHING

We use the $T2K$ Match famework [18] to match the WDC Web Tables corpus and the DBpedia knowledge base. In this section, we give an overview of the matching framework and report statistics about the discovered correspondences. These correspondences help us to understand the contents of the tables and their topical overlap with DBpedia.

4.1 Matching Framework

The $T2K$ Match framework employs an iterative approach to match entity-attribute tables to knowledge bases. An entity-attribute table covers a set of entities (rows in Web tables) which are described by a set of possibly multi-valued attributes (columns). Further, the framework requires entity-attribute tables to contain an entity label attribute with natural language labels for the described entities (e.g. New York, Barak Obama). With $T2K$ Match, we create correspondences that assign a class to each Web table, an instance to each entity and a property to each attribute (if possible). The source code of the matching framework as well as the source code of the data fusion component that will be used in Section 6 is available from the T2K website[4].

Matching Method. The $T2K$ matching method is described in detail by Ritze et al. [18]. In brief, the method initially determines a set of candidate instances for the entities in the Web table. Based on these candidates the algorithm decides for the corresponding class and calculates value-based similarity scores (using data type-specific similarity metrics and flexible value normalization). Using these scores, the algorithm iteratively refines the attribute-to-property and entity-to-instance correspondences. During the whole process, the instance- and schema-level matching mutually influence each other, as one is used to weight the similarities of the other.

Framework Evaluation. We performed an evaluation of $T2K$ Match using the publicly available $T2D$ gold standard[5] which provides correspondences between Web tables and DBpedia. The gold standard contains a total of 1 748 tables with 7 983 property correspondences and 26 124 instance correspondences, distributed over 91 DBpedia classes. The framework achieved an F1-Measure of .82 for instance, .70 for property and .94 for class correspondences [18].

Reference Extension. So far, we did not consider whether a property is an object- or data type property. For string attributes with correspondences to object properties, the values should actually refer to instances in the knowledge base and not to their label. That is why we repeat the candidate selection for all the attributes that correspond to object properties. We replace the string values with the best matching candidate and change the data type of the according attribute to *reference*.

4.2 Correspondence Statistics

Table 4 shows statistics about the matched Web tables with respect to their corresponding DBpedia class (not a complete list). T_0 is the set of tables for which at least the entity label attribute and thus a set of entities could be matched to DBpedia. T_c covers all tables which in addition have a property correspondence. V_c is the amount of cells (values) contained in tables of T_c for which an instance correspondence exists for its entity and a property correspondence for its attribute. In other words, V_c expresses how many triples can be generated from the tables. These numbers are further divided according to their data type in the last four columns of the table.

Tables. Altogether, 949 970 of 33.3 million Web tables have correspondences to DBpedia instances (T_0). These tables have correspondences to a total of 361 different classes from the DBpedia ontology. Such tables describe instances which can be found in DBpedia and are potentially useful for set expansion tasks [23] which add missing instances to the knowledge base. If we additionally require a property correspondence, we find 301 450 tables (T_c) which match altogether 274 different DBpedia classes. These tables are

[4] `http://dws.informatik.uni-mannheim.de/en/research/T2K`

[5] `http://webdatacommons.org/webtables/goldstandard.html`

potentially useful for slot filling tasks [20] which add missing values to the knowledge base. For such a task, the tables contain a total of 8 million values (V_c) which might either already exist in or might be new to the knowledge base. The fact that only 2.85% of all Web tables can be matched to DBpedia indicates that the topical overlap between the tables and the knowledge base is rather low, assuming that the matching step detected all relevant correspondences. This is in line with the frequency of column headers as shown in Table 2. The most frequent headers are centered around products, e.g. "price", "5 star" and such entities are rarely represented in DBpedia.

Classes. To profile the topical overlap between the Web tables and DBpedia, we picked the most frequently matched DBpedia classes from the first levels of the DBpedia class hierarchy and provide detailed statistics for these classes in Table 4. Almost 50% of the Web tables describe *Persons* and *Organisations*, followed by tables covering *Work*. It is no surprise that we find a majority of correspondences for these classes, as they are also in the three most frequent classes in DBpedia (see Table 3). More interesting is the fact that the second most frequent class in DBpedia, *Place* is much less frequent in the Web tables, although it is twice as large as *Work* in the knowledge base. This either indicates that places are underrepresented in the Web tables corpus or that the matching framework has trouble detecting this class. We can further see that only 18% of the tables about places have a property correspondence. Thus beside of being underrepresented, we also find signs for a schema mismatch between the DBpedia ontology and the Web tables.

Data Types. Let us now have a look at the distribution of data types. In the full Web table corpus, the majority of values was of data type *string*, followed by *numeric* and then *date*. Among the values in the matched tables T_c, the majority is now formed by *date*, followed by *numeric*, *string* and *reference*. As the *reference* type requires a matching step, these values appear as *string* in the statistics about the full corpus. Reasons for the change in the distribution can be the following: Either the Web tables have a tendency towards factual data, like dates and numbers, or the schema overlap between the tables and DBpedia consists mainly of properties with these data types. Another reason could be that the matcher allows for more variation in the values with these data types than for strings, resulting in more overall correspondences.

Instance Distribution. In total, we find 13 726 582 instance correspondences for 717 174 unique instances, which is 15.6% of all instances in DBpedia. Figure 1 shows the complementary cumulative distribution function (or tail distribution) of the fraction of instances (y-axis) that have correspondences in a given number of Web tables (x-axis). From this figure we can see that 70% of all instances that have correspondences can be found in more than one Web table. 55% have three or more sources and 25% have at least ten sources. So, for more than two thirds of all instances, we find evidence in more than a single Web table. Looking at the other end of the distribution, about 3% of the instances are described within more than 100 tables.

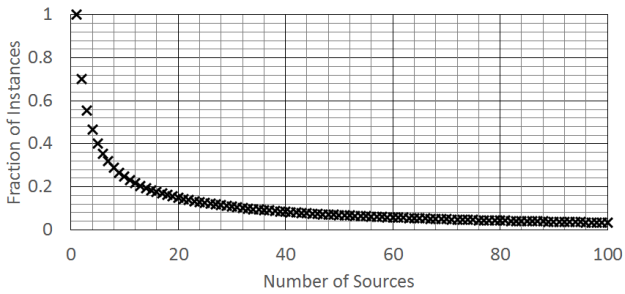

Figure 1: Distribution of Instance Correspondences

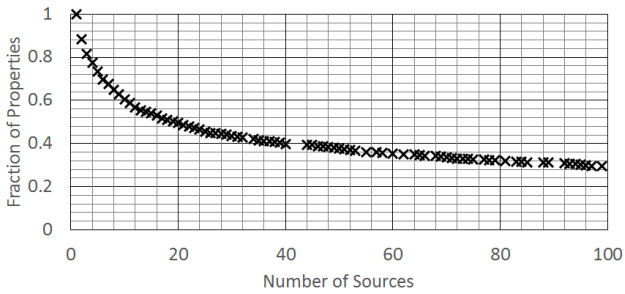

Figure 2: Distribution of Property Correspondences

Property Distribution. Aggregated over all tables, we find a total of 562 445 property correspondences for 721 unique properties. Figure 2 shows the tail distribution of the fraction of properties (y-axis) that have correspondences in a given number of Web tables (x-axis). 88% of all properties have correspondences from at least two Web tables. 81% can be found in three or more Web tables and 60% of all properties have correspondences from at least ten Web tables. About 30% of all properties have more than 100 correspondences. As possibly expected, we find more sources for each property than we do for the instances.

Table 5 lists some examples for frequent instances and properties for selected classes in order to give an impression of the detected correspondences. All of these instances are more or less commonly known, which is in line with the intuitive expectation that more popular entities are found more often.

5. GROUPING TRIPLES

In this section, we present statistics about the internal overlap in the Web tables corpus. First, we generate triples using the detected correspondences. We then group these triples according to their subject and predicate, e.g. all triples with subject <dbp:Germany> and predicate <dbp:populationTotal> are grouped, which results in a group of multiple values for this subject/property combination.

Group Size Distribution. Out of the 8 million triples that we can generate from the Web tables, 929 170 groups of triples can be formed. Figure 3 shows the tail distribution of group sizes. 58% of all groups contain triples from at least two sources, 39% from at least three sources. Triples from ten or more sources can be found for 13% of all groups. Very frequent groups, which are supported by at least 100

Table 4: Correspondence Statistics

DBpedia Class	Number of Tables/Values			V_c Data Type			
	T_0	T_c	V_c	Numeric	Date	String	Reference
+ Person	265 685	103 801	4 176 370	2 117 793	1 588 475	266 628	203 474
|- Athlete	243 322	95 916	3 861 641	2 084 017	1 435 775	163 771	178 078
|- Artist	9 981	2 356	18 886	3	11 527	3 499	3 857
|- Politician	3 701	1 388	18 505	10	7 725	3 393	7 377
|- Office Holder	2 178	1 435	131 633	30	66 762	59 332	5 509
+ Organisation	194 317	36 402	573 633	99 714	187 370	100 710	185 839
|- Company	97 891	6 943	203 899	58 621	83 001	34 665	27 612
|- SportsTeam	50 043	2 722	31 866	2 206	22 368	43	7 249
|- Educational Institution	25 737	14 415	238 365	38 056	64 578	13 334	122 397
|- Broadcaster	14 515	11 315	93 042	564	13 095	52 186	27 197
Work	269 570	127 677	2 284 916	109 265	1 354 923	33 091	787 637
+ MusicalWork	138 676	80 880	1 131 167	64 545	396 940	7 610	662 072
+ Film	43 163	9 725	256 425	10 844	198 913	14 382	32 286
+ Software	39 382	23 829	486 868	418	414 092	9 194	63 164
Place	133 141	24 341	859 995	413 375	273 510	84 111	88 999
+ PopulatedPlace	119 361	21 486	787 854	405 406	257 780	57 064	67 604
|- Country	36 009	6 556	208 886	93 107	66 492	31 793	17 494
|- Settlement	17 388	2 672	17 585	4 492	6 662	2 444	3 987
|- Region	12 109	427	5 625	3 097	897	292	1 339
+ ArchitecturalStructure	10 136	1 815	46 067	3 976	7 387	23 110	11 594
+ NaturalPlace	1 704	254	2 568	866	696	340	666
Species	14 247	4 893	83 359	-	7 902	38 682	36 775
Σ	949 970	301 450	8 037 562	2 751 105	3 437 420	536 526	1 312 511

Table 5: Examples for Frequent Instances and Properties

DBpedia Class	Instance	#Correspondences	Property	#Correspondences
Athlete	Jeff Gordon	15 826	team	7 982
	Fernando Alonso	14 870	championships	4 464
Country	China	13 515	capital	965
	France	13 300	currency	508
Office Holder	John McCain	329	religion	74
	Barack Obama	328	vicePresident	66
Company	Toshiba	59 112	formationDate	1 016
	Nortel	45 573	iataAirlineCode	714
Musical	Can't Help Falling in Love	1 403	releaseDate	60 473
	Hold It Against Me	1 801	musicalBand	27 832
Educational Institution	University of Phoenix	2 486	state	998
	Purdue University	2 325	numberOfStudents	707
Species	Great Egret	541	genus	3 706
	Rainbow trout	329	sire	207

sources, constitute 1% of all groups. Assuming that the matching step found all correspondences, this distribution in combination with the low overlap between the Web tables and DBpedia, which we observed earlier, shows that the Web tables contain a wide range of different triples, but most of these triples are only provided by a small number of sources. Such triples are more likely to be new to the knowledge base, as we expect frequently stated triples to be already existing. But these new triples come with a drawback: As they are only supported by few sources, it will be difficult for a fusion strategy to find the correct values in the groups. For 42% of the subject/property combinations only a single value is present (group size=1), meaning that a fusion strategy cannot choose between different values but can just determine if it wants to accept or discard the single existing value.

Classes. Table 6 shows our selected classes again. The second column indicates the number of groups G that were formed for the respective class and the third column states the ratio of this number to the total number of triples (column V_c in Table 4). This ratio is high if we cannot group many triples -319(t)1(6)-1m [(o)1(n)1(lye1s)-239(436(Th1(rI-340(g)15341(w)25s)-262

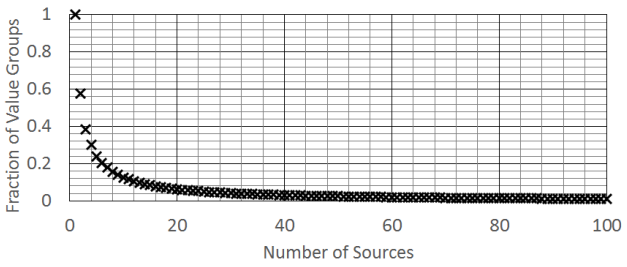

Figure 3: Distribution of Group Sizes

Table 6: Groups by Class

DBpedia Class	G	G/V_c
+ Person	366 048	.088
|– Athlete	284 213	.074
|– Artist	6 842	.362
|– OfficeHolder	6 559	.354
|– Politician	11 362	.086
+ Organisation	87 527	.153
|– Company	25 164	.123
|– SportsTeam	2 453	.077
|– EducationalInstitution	35 736	.150
|– Broadcaster	21 687	.233
Work	331 071	.145
+ MusicalWork	201 186	.178
+ Film	56 610	.221
+ Software	33 552	.069
Place	100 673	.117
+ PopulatedPlace	5 709	.027
|– Settlement	1 879	.107
|– Region	1 193	.212
+ ArchitecturalStructure	12 037	.468
Species	23 809	.286
Σ	929 170	.012

Data Types. Figure 4 and Table 7 show the data type distribution at different stages of our data integration process. At first, we have the full Web tables corpus (*Corpus*). Afterward, we match the corpus (*Matched*) and finally group the generated triples (*Grouped*). As we already discussed the change in the distribution between the full corpus and the correspondences, we now focus on the transition from correspondences to groups, where all triples with the same subject/property combination are put together. The last column in Table 7 shows the ratio of this grouping process. We see that, on average, each group contains 8.46 triples. The largest group sizes can be observed for *numeric* triples, where on average 13.59 triples form a group. *Date* groups are also relatively large with about 9 triples per group. *String* and *reference* groups, however, are quite small with only about 4 to 5 triples per group. Figure 4 shows the number of triples per data type as proportions in each step. Here it becomes obvious how the large fraction of *string* values in the complete corpus is replaced by *date* and *numeric* in the correspondences. In the grouped stage, we see how the relative size of *string* and *reference* increases again, as many *date* and *numeric* values are grouped together.

Table 7: Distribution of Data Types

Data Type	Corpus	Matched	Grouped	Ratio
Numeric	995M	2 751 105	202 362	13.59
Date	101M	3 437 420	379 240	9.06
String	19 000M	536 526	86 330	4.29
Reference	0M	1 312 511	261 238	5.02
Σ	20 096M	8 037 562	929 170	8.46

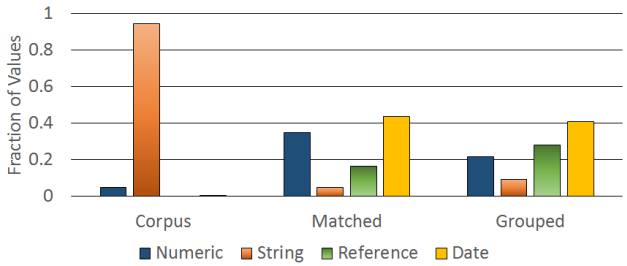

Figure 4: Distribution of Data Types Aggregated by Their Steps

6. DATA FUSION

This section investigates the quality of new triples that can be generated by fusing [2, 4] Web table data. To this end, we first establish our evaluation methodology, which uses the existing triples in the knowledge base as ground truth, an assumption that we additionally check by a manual evaluation. Then, we compare the performance of three different data fusion strategies: One strategy with a knowledge-based quality measure, one that uses PageRank as an external quality indicator, and a voting-based baseline approach.

6.1 Evaluation Methodology

We evaluate the correctness of the generated triples by comparing them to triples which already exist in the DBpedia knowledge base (overlapping triples). For this comparison, we apply the Local Closed World Assumption (LCWA) as proposed by Dong et al. [9]: For triples with subject s, predicate p and object o, let $O(s,p)$ be the set of objects for s,p existing in a knowledge base (overlapping triples). If a triple (s,p,o) is in $O(s,p)$, we assume the triple to be true. Otherwise, if (s,p,o) is not in $O(s,p)$ and $O(s,p)$ is not empty, the triple is said to be incorrect. In cases where $O(s,p)$ is empty, we exclude the triple from the evaluation (non-overlapping triples). Dong et al. show that this assumption is a valid approximation.

We use the LCWA to enable a large-scale automatic evaluation to the results of our fusion step. This gives us an estimate of the performance for the respective fusion strategy (which we double-check with a manual evaluation described in Section 6.5). As we cannot expect data from Web tables to be perfectly clean, we allow for minor deviations when comparing generated triples to overlapping triples from the knowledge base: Numeric values are treated as equal if they do not deviate more than 5%. For dates, the day, month and year parts must exactly match, if they are available. Whenever a Web table or DBpedia only contains the year part, we only compare this information. For strings, we use Generalized Jaccard with Levenshtein similarity for token comparisons. References, however, must be exact matches.

71 981 .091

6.2 Upper Bound of the Fusion Performance

By checking which groups contain correct triples that already exist in the knowledge base, we can estimate the upper bound of the data fusion performance, meaning that we can estimate the maximal number of correct triples that could be produced by a hypothetical, ideal data fusion strategy. For 691 622 of the 929 170 groups, the set of objects $O(s, p)$ for s,p is not empty, meaning that they overlap with DBpedia. On these groups, we apply the similarity measures described above and find that the number of groups containing the correct triple is $correct_{max} = 310\,284$. Using this maximal number of correct triples together with the the number of correct triples $fused_{correct}$ that are produced by a specific data fusion strategy and the total number of triples $fused_{total}$ generated by the fusion strategy, we can define precision and recall as in Equations 1 and 2.

$$Precision = \frac{fused_{correct}}{fused_{total}} \quad (1)$$

$$Recall = \frac{fused_{correct}}{correct_{max}} \quad (2)$$

The upper bound of the fusion performance $correct_{max}$ is an important finding about the Web tables corpus and the matching methods that we applied so far, as it restricts the quality of the data that can theoretically be generated from the groups by the ideal data fusion strategy. We observe that only 45% of all groups G with non-empty $O(s, p)$ contain the correct triple at least once, which seems quite low at first sight. But we have to take into account that it requires correct matching decisions about the class, property and instance and, in the case of a *reference* data type, also a correct transformation of the string into an instance. Multiplying the errors happening in all these matching decisions explains this percentage and does not even take into account that information in the Web tables might also simply be wrong or outdated.

6.3 Fusion Strategies

The goal of the data fusion step is to decide which triple of a group with the same subject/predicate combination will be selected as output and used by subsequent steps such as slot filling. We compare the following three data fusion strategies:

1. **Majority/Median Fusion (MM).** This data fusion strategy selects the most frequent value in the group as output (simple voting) for groups of data type *string* and *reference*. For groups of data type *numeric* and *date*, the median of all values is calculated and returned.[6] The MM strategy is thus a rather simple baseline strategy which does not take any external quality indicators into account.

2. **Knowledge-based Trust (KBT).** We extend the MM strategy by assigning a trust score to each triple. For *string* and *reference* we then apply a weighted vote and for *numeric* and *date* a weighted median. The trust score is calculated for each attribute (from the Web table that is the source of the triple) as the number of correct overlapping triples, normalized by the total number of overlapping triples. For comparing triples from the Web table with triples from the knowledge base, we allow the same minor deviations as described in Section 6.1. In addition to weighting the values, we completely filter out all attributes with a trust score below .35. By calculating the score from the overlapping triples, we create a measure of correctness for the attribute that is the source of the respective triples. This corresponds to the concept of knowledge-based trust [10, 26] as we weight each triple by the correctness of the other information provided by same source (here: Web table) with regard to information that is considered to be trustworthy (the knowledge base). Note that as we use the same methods for the calculation of the trust score and the evaluation, we apply a 5-fold cross-validation for this strategy.

3. **PageRank-based Trust (PR).** This strategy works like the KBT strategy, with the difference that the score assigned to each triple is the normalized PageRank [16] of the website that is the source of the corresponding Web table. Over the last decade, PageRank was widely used to assess the quality of Web content and has also previously been used for data fusion [17]. The PageRank scores are calculated on the host-level using the 128 billion hyperlinks contained in the 2012 version of the CommonCrawl.[7] Filtering does not improve the results for the PageRank scores. In contrast to KBT, PageRank relies on hyperlinks as quality indicators while KBT relies on comparing Web data to previously trusted data.

[6]Note that we do not take the modal value since the groups tend to be small such that outliers could be determined as output.

6.4 Comparison of the Fusion Strategies

In this section, we report the performance of the different fusion strategies. This is done for two reasons. First, we want to determine which strategy works best for the given data set. This strategy is then used for further evaluations and to report more information about the potential of Web tables for slot filling. Second, we want to examine the recent claim by Dong et al. [10] that knowledge-based trust outperforms a strategy with PageRank as quality indicator [9].

Table 8 shows the number of overlapping fused triples F_o and non-overlapping fused triples F

Our baseline approach, MM, does not apply any filtering, hence the precision can maximally be 45% ($correct_{max}$ divided by F_o). Taking this into account, the achieved precision of 36.9% is at an acceptable level for a simple approach. The MM fusion is able to identify the correct triple for 82.3% of all groups. This also includes all groups of size one, where the fusion cannot choose from multiple triples and just forwards the received input as output. The second approach, KBT, filters out attributes with a low trust score and can hence decide not to produce a triple from a given group. This results in a large 27 percentage point increase in precision and only has a very small trade-off in recall, which decreases by 3.8 percentage points. The third strategy, PR, does not result in any improvement over the MM baseline (not even if we completely filter out values with low PageRank scores). Thus, we can confirm the finding of Dong et al. [10] that the quality of a Web source is not necessarily determined by its popularity. As KBT performs best, we choose this fusion strategy for further investigations.

6.5 Manual Evaluation

We perform two manual evaluations in order to verify the fusion results. First, we test the LCWA by manually evaluating a sample of overlapping fused triples. Second, we manually evaluate a sample of non-overlapping fused triples to determine whether the performance on overlapping fused triples can be transferred to non-overlapping fused triples.

To test the LCWA, we manually evaluate a set of 1 000 overlapping fused triples. The automatic evaluation of this sample according to Section 6.1 results in a precision of .678, while three human annotators determine a precision of .716. Overall 958 out of 1 000 triples were evaluated correctly by the automatic evaluation, which results in an error rate of 4.2%. This result is a signal for the validity of the LCWA and justifies its application for our experiments. However, during the manual evaluation we spot some error categories, which shed light on possible shortcomings of this method:

- **Changes Over Time.** Objects that are changing over time can be outdated in the knowledge base, leading to an incorrect evaluation of more up-to-date Web tables. Since the up-to-dateness of knowledge bases is an important motivation we will focus on the temporal dimension in our future work.
- **Different Granularity.** Objects can have different levels of granularity, e.g. the *city* of the *Emroy university* is *Druid Hills Georgia* in DBpedia. In the Web tables, we find the object *"Atlanta"*. These labels do not look similar to string comparison functions. But knowing that *Druid Hills Georgia* is a community in the metropolitan area of Atlanta, this triple can be regarded as correct.
- **Missing Objects in Lists.** If a list is incomplete in the knowledge base, the automatic evaluation fails for cases in which a Web table contains a correct, but missing value.

The second question we want to investigate is whether the performance that we estimate using the LCWA based on the overlapping fused triples can be transferred to the non-overlapping fused triples. As the non-overlapping fused triples are the candidates for slot filling, an evaluation that cannot be transferred would not be suitable for this task. Hence, we manually evaluate another sample of 500 randomly selected, non-overlapping fused triples. On this sample, the KBT strategy achieves a precision of .624.[8] The determined precision is very close to the one that was estimated using the LCWA on the overlapping fused triples (.639), which we take as an indication for the validity of transferring the performance to non-overlapping fused triples.

6.6 Fusion Results

Now that we have tested our methodology, we report details about the data fusion results with respect to the potential of Web tables for slot filling. We show separate performance statistics for data types, classes and properties.

Table 9: Evaluation of the Datatypes

Data Type	F_o	F_{no}	Precision	Recall	F1
Numeric	28 364	10 613	.644	.452	.531
Date	171 653	23 301	.627	.806	.705
String	34 260	14 285	.755	.811	.783
Reference	144 615	16 038	.629	.871	.730

Data Types. Table 9 shows the fusion performance by data type. The first column F_o contains the number of fused triples that overlap with DBpedia, the second column F_{no} the number of non-overlapping fused triples. All performance measures are calculated on the overlapping fused triples. While the *date*, *reference* and *string* data types have a comparable performance, the recall of data type *numeric* is significantly lower. As it seems, some *numeric* attributes tend to be more noisy due to conflicting objects, changes over time or different interpretations of certain properties. Thus, even correct triples are filtered out by the KBT fusion, as the trust score is not high enough.

We further identified the following reoccurring causes of incorrect fusion results:

- **Conversion Issues.** Some conversions like converting the date format from different countries are not easily solved. As an example, the *birthDate* of *Jeff Zatkoff* is *"6/9/1987"* according to DBpedia but we find the object *"9/6/1987"* in the Web tables. Without knowing which date format is used within the Web table, it is hard to parse the date correctly. This problem constitutes a large part of the error for the data type *date*.
- **Ambiguous Entities.** The identity resolution both for the subjects and objects of triples can make mistakes, especially if the label of the subject or object is ambiguous. This can occur with very common names of people or with musical works like album or single names, for example cover versions. A wrongly identified subject can lead to incorrect results for all data types while incorrect objects only pose a problem for the *reference* data type.

[8] 19 triples were excluded as the human annotators could not determine the correct object. This happened for example for rare properties like *bSide* of a record or *upperAge* of colleges.

Classes. Table 10 shows the fusion results for the set of classes that are also presented in Table 4. The second column contains the number of overlapping triples F_o per class while the third column shows the set of non-overlapping triples F_{no}. All performance measures in the last three columns are computed on F_o. We find the highest amount of non-overlapping fused triples for *Work*, especially *Film*, and for *Person*, especially *Athlete*. This gives another hint for which parts of DBpedia slot filling based on Web tables can be beneficial. Concerning precision and recall, we achieve the best results for *Species* and *Place*.

Table 10: Class Evaluation

DBpedia Class	F_o	F_{no}	Prec.	Rec.	F1
+ Person	117 522	15 050	.639	.723	.678
|- Athlete	84 562	9 067	.646	.679	.662
|- Artist	2 019	427	.711	.830	.766
|- OfficeHolder	3 465	510	.698	.849	.766
|- Politician	3 124	1 167	.533	.765	.628
+ Organisation	20 522	7 903	.645	.691	.667
|- Company	6 376	2 547	.700	.834	.761
|- SportsTeam	790	132	.671	.892	.766
|- Educational Institution	8 844	3 132	.638	.714	.674
|- Broadcaster	4 004	1 924	.557	.459	.503
Work	189 131	27 867	.614	.828	.705
+ MusicalWork	118 511	8 427	.599	.830	.695
+ Film	29 903	12 143	.573	.803	.669
+ Software	17 554	2 766	.591	.760	.665
Place	32 855	9 871	.767	.858	.810
+ PopulatedPlace	16 604	6 704	.711	.779	.743
|- Country	2 084	433	.738	.690	.713
|- Settlement	540	224	.583	.669	.623
|- Region	362	70	.587	.784	.671
+ Architectural Structure	10 441	1 775	.834	.940	.884
+ NaturalPlace	743	64	.843	.940	.889
Species	9 016	1 429	.783	.892	.834

Properties. Table 11 shows the performance for selected properties. In the first four columns we can find the properties with the highest number of overlapping fused triples while the next four columns depict the properties with the highest number of non-overlapping fused triples. Further, a selection of properties with a high precision and at least 50 non-overlapping fused triples can be found in the third column. The columns labeled "ratio" show the ratio between the number of fused triples as given in the preceeding columns and the total number of triples (distinct subjects) for this property in DBpedia.

Looking at the properties with the most overlapping fused triples, we can again see that the majority of the topical overlap between DBpedia and the Web tables is about *Work* (releaseDate) and *Person* (birthDate). Concerning the precision, most properties are close to our overall average performance, with exceptions being *musicalArtist* and *number* with a lower precision. Supposedly, this is caused by number (e.g. the number of a baseball player in a certain team) being a time-varying property. For *musicalArtist*, the identity resolution could be a problem, as this property is applied to songs, which can often have ambiguous labels.

For the properties with the most non-overlapping fused triples, we approximate the precision with the precision that was achieved on the overlapping fused triples for the same property. The ratio column shows the potential for slot filling. We can almost double the number of *publicationDate* triples and increase the amount of *releaseDate* triples in the knowledge base by 11%.

To illustrate in which cases a slot filling approach would result in very high quality data, the last set of columns shows properties with high precision. While the properties with the highest precision can only add a rather small amount of non-overlapping fused triples, the properties *throwingSide* (for *BaseballPlayer*), *icaoLocationIdentifier* (for *Place*) and *family* (for *Species*) add thousands of triples with an above average precision.

7. RELATED WORK

Recent studies have shown that knowledge extracted from Web tables can be useful for applications like table search [21, 1], table extension [25, 14, 8], and knowledge base augmentation [22, 9, 19]. For most of these applications, the matching of Web tables plays an important role [28, 15, 25, 3].

In order to judge the potential of Web tables for different applications, it is essential to have an understanding of the data profile and topical distribution of large Web table corpora. Hassanzadeh et al. [12] analyzed the topical distribution of the same table corpus that we also use for this paper by matching columns to classes of different knowledge bases. By comparing the Web tables to DBpedia, YAGO and Schema.org data, they show that the size and topics that are covered by the knowledge base strongly influence the distribution of correspondences that are discovered. Similar to this work, they also find out that only relatively small fraction of the Web tables can be matched to a knowledge base. In our work, we go beyond their analysis and do not only consider DBpedia classes but also properties and instances. We also examine the potential of Web tables for filling missing values in the knowledge base.

Several other works focus on the construction of knowledge bases by using Web sources [24]. Dong et al. [9] present a method for automatically constructing a web-scale probabilistic knowledge base by combining data from four types of sources: Web texts, DOM trees, Web tables and semantic annotations (such as schema.org). Multiple extractors are used for each kind of source, generating altogether 1.6B triples with only 0.5% originating from Web tables. Around $0.6M$ of all Web table triples are considered as high quality which is comparable to our results. By combining different sources, they show that the probability to find a correct triple increases. To automatically evaluate their approach, they use LCWA and show its validity. Our work confirms the applicability of this assumption. In addition, we show that quality approximations based on the LCWA can even be transferred to non-overlapping new triples.

Sekhavat et al. [19] augment an existing knowledge base with facts from Web tables by levering a Web text corpus and natural language patterns associated with relations in the knowledge base. With a selection of spreadsheets from two web sites, they generate facts and show the potential of filling missing triples in YAGO. The InfoGather system [25]

Table 11: Fusion Results for Various Properties

Most Overlapping Triples				Most Non-Overlapping Triples				Highest Precision		
Property	F_o	Prec.	Ratio	Property	F_{no}	Ratio	Prec.	Property	F_{no}	Prec.
releaseDate	92383	.628	.670	releaseDate	15836	.115	.628	numberOf Islands	157	1.00
birthDate	61636	.769	.055	number	3557	.059	.383	province	67	1.00
artist	25563	.649	.268	publicationDate	2693	.964	.688	seniority	60	1.00
musicalArtist	20663	.288	.527	alias	1471	.011	.436	sire	366	.990
musicalBand	18160	.498	.463	locationCountry	1304	.089	.564	games	247	.973
director	8082	.623	.095	country	1242	.002	.667	illustrator	81	.969
activeYears StartDate	7934	.658	.116	synonym	1240	.014	.559	iso6391Code	236	.967
activeYears EndDate	7861	.710	.140	status	1116	.042	.421	throwingSide	2500	.961
deathDate	7448	.625	.015	birthDate	1000	.001	.769	icaoLocation Identifier	5459	.941
number	6160	.383	.103	artist	971	.010	.649	family	4760	.846

performs entity augmentation as well as attribute discovery by exploiting indirect matches among the Web tables as well as the page context surrounding the tables. Similar to our results, they detect that numeric and time-varying attributes pose a challenge which they tackle with the InfoGather+ system [27]. Gupta et al. [11] explores the use of Web text and Web tables combined with query stream data to construct a large ontology of binary attributes, called Biperpedia. They use Biperpedia to better understand attributes of Web tables and also confirm that only a small fraction of Web table attributes can be matched to an existing knowledge base (in their case Freebase).

The concept of knowledge-based trust has been introduced by Dong et al. [10] who show that the trustworthiness of Web sources can be estimated by comparing the information from Web source to a trusted knowledge base. They estimate the correctness of information and the trustworthiness of sources using probabilistic inference. We use the idea of knowledge-based trust for the scoring of triples during the fusion and show that the trust scores help to filter out incorrect information. The comparison with PageRank indicates in both cases that exploiting hyperlinks is not necessarily the best approach for judging the quality of Web data sources.

8. CONCLUSION

In this paper, we presented the results of profiling a large corpus of Web tables with regard to its potential for filling missing values in knowledge bases. To the best of our knowledge, we are the first to provide such an in-depth analysis for a publicly available corpus with openly accessible methods (the original web crawl, the extraction framework, and the matching and fusion code are available for download).

Our results show that the majority of the Web tables does not contain data that can be related to the DBpedia knowledge base. Only 1.3% of all tables that were extracted from the Web crawl contained relational data. Out of these relational tables, about 3% could be matched to DBpedia. These percentages are comparable to the results of other studies [7, 9, 12].

In the set of Web tables that could be matched to DBpedia, we found a distribution of instances over classes which is similar to the overall distribution in DBpedia. While some deviations are worth further investigation, this confirms the findings from Hassanzadeh et al. [12] that the correspondence distribution strongly depends on the content of the knowledge base. Looking at the frequency distributions, we can state that 70% of the matching DBpedia instances are described within at least two Web tables. For properties this holds for 88%. Using the correspondences from the matching step, we grouped the resulting triples by subject and predicate which results in about 1 million groups of alternative values (average group size 8.5).

In subsequent experiments, we examined recent results in the area of data fusion. An important finding is that we can experimentally confirm the applicability of the Local Closed World Assumption and even show that the performance approximation for overlapping fused triples is transferable to fused triples with no overlap in the target knowledge base. We apply this assumption for the comparison of several fusion strategies and find that knowledge-based trust outperforms PageRank-based fusion as well as a voting-based baseline strategy.

Finally, we had a look at the outcomes of the data fusion process and examined the slot filling potential of Web tables for DBpedia in terms of quality as well as quantity. Again, we provide detailed statistics for the different classes and properties to get an impression which parts of DBpedia can especially benefit from slot filling with Web tables data. As an example, we can almost double the number of *publicationDate* triples in DBpedia.

9. REFERENCES

[1] S. Balakrishnan, A. Y. Halevy, and B. Harb. Applying WebTables in Practice. In *Proc. of the 7th Biennial Conference on Innovative Data Systems Research*, CIDR '15, 2015.

[2] J. Bleiholder and F. Naumann. Data fusion. *ACM Comput. Surv.*, 41(1):1—41, 2009.

[3] K. Braunschweig, M. Thiele, J. Eberius, and W. Lehner. Column-specific Context Extraction for Web Tables. In *Proc. of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 1072–1077, 2015.

[4] V. Bryl and C. Bizer. Learning conflict resolution strategies for cross-language wikipedia data fusion. In

Proc. of the 23rd Int. Conference on World Wide Web Companion, WWW '14, pages 1129–1134, 2014.

[5] M. Cafarella, Y. Halevy, Alonand Zhang, D. Z. Wang, and E. Wu. Uncovering the Relational Web. In *Proc. of the WebDB Workshop*, 2008.

[6] M. J. Cafarella, A. Halevy, and N. Khoussainova. Data Integration for the Relational Web. *Proc. of the VLDB Endow.*, 2:1090–1101, 2009.

[7] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. WebTables: Exploring the Power of Tables on the Web. *Proc. of the VLDB Endow.*, 1:538–549, 2008.

[8] A. Das Sarma, L. Fang, N. Gupta, A. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu. Finding Related Tables. In *Proc. of the Int. Conference on Management of Data*, pages 817–828, 2012.

[9] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proc. of the 20th SIGKDD*, pages 601–610, 2014.

[10] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based Trust: Estimating the Trustworthiness of Web Sources. *Proc. of the VLDB Endow.*, 8(9):938–949, 2015.

[11] R. Gupta, A. Halevy, X. Wang, S. Whang, and F. Wu. Biperpedia: An Ontology for Search Applications. In *Proc. of the 40th Int. Conference on Very Large Data Bases*, 2014.

[12] O. Hassanzadeh, M. J. Ward, M. Rodriguez-Muro, and K. Srinivas. Understanding a large corpus of web tables through matching with knowledge bases: an empirical study. In *Proc. of the 10th Int. Workshop on Ontology Matching*, 2015.

[13] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[14] O. Lehmberg, D. Ritze, P. Ristoski, R. Meusel, H. Paulheim, and C. Bizer. The Mannheim Search Join Engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:159–166, 2015.

[15] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and Searching Web Tables Using Entities, Types and Relationships. *Proc.of the VLDB Endow.*, 3:1338–1347, 2010.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the Web. Technical report, Stanford InfoLab, 1999.

[17] J. Pasternack and D. Roth. Knowing What to Believe (when You Already Know Something). In *Proc. of the 23rd Int. Conference on Computational Linguistics*, pages 877–885, 2010.

[18] D. Ritze, O. Lehmberg, and C. Bizer. Matching HTML Tables to DBpedia. In *Proc. of the 5th Int. Conference on Web Intelligence, Mining and Semantics*, 2015.

[19] Y. A. Sekhavat, F. di Paolo, D. Barbosa, and P. Merialdo. Knowledge Base Augmentation using Tabular Data. In *Proc. of the 7th Workshop on Linked Data on the Web*, 2014.

[20] M. Surdeanu and H. Ji. Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation. `http://nlp.cs.rpi.edu/paper/sf2014overview.pdf`, 2014.

[21] P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering Semantics of Tables on the Web. *Proc. of the VLDB Endow.*, pages 528–538, 2011.

[22] J. Wang, H. Wang, Z. Wang, and K. Q. Zhu. Understanding Tables on the Web. In *Proc. of the 31st Int. Conf. on Conceptual Modeling*, pages 141–155, 2012.

[23] R. C. Wang and W. W. Cohen. Iterative set expansion of named entities using the web. In *Proc. of the 8th IEEE Int. Conference on Data Mining*, ICDM '08, pages 1091–1096, 2008.

[24] G. Weikum and M. Theobald. From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In *Proc. of the 29th Symp. on Principles of Database Systems*, pages 65–76, 2010.

[25] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. InfoGather: Entity Augmentation and Attribute Discovery by Holistic Matching with Web Tables. In *Proc. of the 2012 SIGMOD*, pages 97–108, 2012.

[26] X. Yin and W. Tan. Semi-supervised truth discovery. In *Proc. of the 20th Int. Conference on World Wide Web*, WWW '11, pages 217–226. AC, 2011.

[27] M. Zhang and K. Chakrabarti. InfoGather+: Semantic Matching and Annotation of Numeric and Time-varying Attributes in Web Tables. In *Proc. of the 2013 ACM SIGMOD Int. Conference on Management of Data*, pages 145–156, 2013.

[28] X. Zhang, Y. Chen, J. Chen, X. Du, and L. Zou. Mapping Entity-Attribute Web Tables to Web-Scale Knowledge Bases. In *Database Systems for Advanced Applications*, pages 108–122. Springer Berlin, 2013.