Reflections on: KnowMore - Knowledge Base Augmentation with Structured Web Markup

Ran Yu[1], Ujwal Gadiraju[2], Besnik Fetahu[2], Oliver Lehmberg[3],
Dominique Ritze[3], and Stefan Dietze[1,2,4]

[1] GESIS - Leibniz Institute for the Social Sciences, 50667 Cologne, Germany
{ran.yu, stefan.dietze}@gesis.org
[2] L3S Research Center, 30167 Hannover, Germany
{gadiraju,fetahu}@l3s.de
[3] University of Mannheim, 68159 Mannheim, Germany
{oli,dominique}@informatik.uni-mannheim.de
[4] Heinrich-Heine-University Düsseldorf

Abstract. Knowledge bases are in widespread use for aiding tasks such as information extraction and information retrieval. However, knowledge bases are known to be inherently incomplete. As a complimentary data source, embedded entity markup based on Microdata, RDFa, and Microformats have become prevalent on the Web. RDF statements extracted from markup are fundamentally different from traditional knowledge graphs: entity descriptions are flat, facts are highly redundant and of varied quality, and, explicit links are missing despite a vast amount of coreferences. Therefore, data fusion is required in order to facilitate the use of markup data for KBA. We present a novel data fusion approach which addresses these issues. We perform a thorough evaluation on a subset of the Web Data Commons dataset and show significant potential for augmenting existing knowledge bases. A comparison with existing data fusion baselines demonstrates superior performance of our approach when applied to Web markup data.

1 Introduction

Knowledge bases (KBs) such as Freebase [1] or YAGO [8] are in widespread use to aid a variety of applications and tasks such as Web search and Named Entity Disambiguation (NED). While KBs capture large amounts of factual knowledge, their coverage and completeness vary heavily across different types or domains. In particular, there is a large percentage of less popular (long-tail) entities and properties that are underrepresented. Recent efforts in knowledge base augmentation (KBA) aim at exploiting data extracted from the Web to fill in missing statements. These approaches extract triples from Web documents [2], or exploit semi-structured data from Web tables [6, 7]. After extracting values, data fusion techniques are used to identify the most suitable value (or fact) from a given set of observed values.

Building on standards such as RDFa, Microdata and Microformats, and driven by initiatives such as schema.org, a joint effort led by Google, Yahoo!, Bing and Yandex, markup data has become prevalent on the Web. Through its wide availability, markup lends itself as a diverse source of input data for KBA. However, RDF statements extracted from markup are fundamentally different from traditional knowledge graphs:

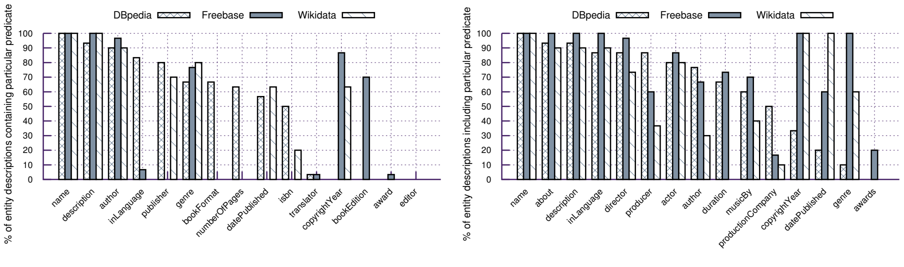
entity descriptions are flat, facts are highly redundant and of varied quality, and, explicit links are missing despite a vast amount of coreferences.

In this work, we introduce *KnowMore*, an approach based on data fusion techniques which exploits markup crawled from the Web as source of data to aid KBA. Our approach consists of a two-fold process, where first, candidate facts for augmentation of a particular KB entity are retrieved through a combination of blocking and entity matching techniques. In a second step, correct and novel facts are selected through a supervised classification approach and an original set of features. We apply our approach to the WDC2015 dataset and demonstrate superior performance compared to state-of-the-art baselines. We also demonstrate the capability for augmenting three large-scale knowledge bases, namely Wikidata, Freebase and DBpedia through markup data based on our data fusion approach. The main contributions of our work are threefold:

– **Pipeline for data fusion on Web markup.** We propose a pipeline for data fusion that is tailored to the specific challenges arising from the characteristics of Web markup.
– **Model & feature set.** We propose a novel data fusion approach consisting of a supervised classification model, utilising an original set of features geared towards validating correctness and relevance of markup facts.
– **Knowledge base augementation from markup data.** As part of our experimental evaluation, we demonstrate the use of fused markup data for augmenting three well-established knowledge bases.

2 Motivation & Problem Definition

Motivation. For a preliminary analysis of *DBpedia*, *Freebase* and *Wikidata*, we randomly select 30 Wikipedia entities of type *Movie* and *Book* and retrieve the corresponding entity descriptions from all three KBs. We select the 15 most frequently populated properties for each type and provide equivalence mappings across all KB schemas as well as the *schema.org* vocabulary manually. Since all vocabulary terms and types in the following refer to *schema.org*, prefixes are omitted. Figure 1 shows the proportion of instances for which the respective properties are populated. We observe a large amount of empty slots across all KBs for most of the properties, with an average proportion of missing statements for books (movies) of 49.8% (37.1%) for DBpedia, 63.8% (23.3%) for Freebase and 60.9 % (40%) for Wikidata.



(a) Book (b) Movie

Fig. 1: Proportion of book and movie instances per KB that include selected popular predicates.

In addition, coverage varies heavily across different properties, with properties such as *editor* or *translator* being hardly present in any of the KBs. Tail entities/types as well as time-dependent properties which require frequent updates, such as the *award* of a book, are prevalent in markup data [4], yet tend to be underrepresented in structured KBs. Hence, markup data lends itself as data source for the KBA task. However, given the specific characteristics of markup data [9], namely the large amount of coreferences and near-duplicates, the lack of links and the variety of errors, data fusion techniques are required which are tailored to the specific task of KBA from Web markup.

Problem Definition. Our work is concerned with entity descriptions extracted from structured Web markup. We refer to such a dataset as M, where the WDC dataset is an example. Data in M consists of entity descriptions e_i, each consisting of a set of RDF quads, i.e. a set of $\langle s, p, o, u \rangle$ quadruples which are referring to entities. The elements $\langle s, p, o, u \rangle$ of the quadruple represent subject, predicate, object and the URL of the document from which the triple $\langle s, p, o \rangle$ has been extracted, respectively.

There exist $n \geq 0$ subjects $\{s_1, s_2, ..., s_n\}$, and consequently, n entity descriptions $e_i = \langle s_i, p_i, o_i \rangle \in E$ which represent a particular query entity q in M. Here, E is the set of all entity descriptions which (co)refer to entity q. We define a property-value pair $\langle p, o \rangle$ describing the entity q as a fact of q. Note that we explicitly consider multi-valued properties, i.e. a particular predicate p might be involved in more than one fact for a particular entity q. We define the task of augmenting an entity description e_q, representing a query entity q within a particular KB from data in a markup corpus M as follows:

Definition 1. *KBA task: For a query entity q that is represented through an entity description e_q in a KB, we aim at selecting a subset F_{nov} from M, where each fact $f_i \in F_{nov}$ represents a valid fact which augments the entity description e_q for q.*

F_{nov} represents the final output of the KnowMore pipeline. We consider a fact valid for augmentation, if it meets the following criteria:

- A fact is *correct* with respect to query entity q, i.e. consistent with the real world regarding query entity q according to some ground truth (Section 4).
- A fact represents *novel*, i.e. not duplicate or near-duplicate, information with regard to the entity description e_q of q in a given KB.
- The predicate p_i of fact $\langle p_i, o_i \rangle$ should already be reflected in a KBs given schema.

3 Approach

3.1 Entity Matching

The first step, $KnowMore_{match}$, aims at obtaining candidate facts $f_i \in F$ by collecting the set E of coreferring entity descriptions $e_i \in E$ from M which describe q and corefer to the entity description e_q in a given KB. We use a three step approach in order to efficiently achieve high accuracy results.

Data Cleansing. This step aims at (i) resolving object references and (b) fixing common errors [3] to improve overall usability of the data. Given the prevalence of literals in Web markup and the need to homogenise entity descriptions for further processing, we resolve object references into literals by replacing object URIs with the labels of the corresponding entity. In addition, based on earlier work [3] which studied common errors in Web markup, we implement heuristics and apply these to E as a cleansing step to fix wrong namespaces, and handle undefined types and properties.

Blocking. We implement the blocking step through entity retrieval using the BM25 model to reduce the search space. We created an index for each type-specific subset using Lucene, and then use the *label* of e_q to query the field *name* within a type-specific index. This result in a set of candidate entity descriptions $e_i^0 \in E_0$ that potentially describe the same entity as e_q.

Entity Matching. This step is for the validation of each entity description $e_i^0 \in E_0$ in the result of the blocking step. We use supervised classification on the similarity vector between e_i^0 and e_q. In order to compute the similarity for each property, we consider all properties as attributes of the feature space $\overrightarrow{A} = \{a_1, a_2, ..., a_n\}$, so that each entity description e can be represented as a vector of values $\overrightarrow{v} = \{o_{a1}, o_{a_2}, ..., o_{a_n}\}$ which represent the objects of the considered $\langle p, o \rangle$ tuples. We construct a similarity vector $\overrightarrow{sim}(\overrightarrow{v^{KB}}, \overrightarrow{v})$ between e_q and each entity description $e_i^0 \in E_0$ as in Equation 1.

$$\overrightarrow{sim}(\overrightarrow{v^{KB}}, \overrightarrow{v}) = \{\lambda_{a_1}, \lambda_{a_2}, ..., \lambda_{a_n}\} \tag{1}$$

$$\lambda_{a_i} = sim(o_{a_i}^{KB}, o_{a_i}) \tag{2}$$

In order to compute $sim(o_{a_i}^{KB}, o_{a_i})$, we employ datatype-specific similarity metrics, i.e., we implemented one similarity measure for each *schema.org* datatype, and automatically select the appropriate metric. We then train a supervised classification model, to make the decision whether or not e_i^0 is a match for e_q. We experimented with several state-of-the-art classifiers (SVM, Logistic Regression and Naive Bayes). Since Naive Bayes achieves a $F1$ score that is 0.08 higher than the best SVM (linear kernel), and 0.123 higher than the Logistic Regression (LR), throughout the remaining paper we rely on a trained Naive Bayes classifier unless otherwise stated.

3.2 Data fusion

During the data fusion step, $KnowMore_{class}$, we aim at selecting a subset $F_{nov} \subset F$ that fulfills the criteria as listed in Section 2. More specifically, we introduce *data fusion* techniques based on supervised classification to ensure the correctness and two deduplication steps to ensure novelty, namely deduplication with respect to M ($KnowMore_{ded}$) and deduplication with respect to the KB ($KnowMore_{nov}$).

Table 1: Features for supervised data fusion from markup data.

Category	Notation	Feature description
Source level	t_1^r, t_2^r, t_3^r	Maximum, minimum, average PageRank score of the PLDs containing fact f
	t_4^r, t_5^r, t_6^r	Maximum, minimum, average percentage of common errors [3] of the PLDs containing fact f
	t_7^r, t_8^r, t_9^r	Maximum, minimum, average precision (based on training data) of the PLDs containing fact f
Entity level	t_1^e, t_2^e, t_3^e	Maximum, minimum, average size (number of facts) of e_i containing f
Property level	t_1^p	Predicate term
	t_2^p	Predicate frequency in F
	$\dagger t_3^p$	Amount of clusters of predicate p
	$\dagger t_4^p$	Average cluster size of predicate p
	$\dagger t_5^p$	Variance of the cluster sizes of predicate p
Fact level	t_1^f	Fact frequency in F
	$\dagger t_2^f$	Normalized cluster size that f belongs to

†-features extracted based on clustering result

Correctness - Supervised Classification. The first step ($KnowMore_{class}$) aims at detecting correct facts by learning a supervised model that produces a binary classification for a given fact $f \in F$ into one of the labels $\{$*'correct'*, *'incorrect'*$\}$. For the classification model, we have experimented with several different approaches. We rely on a Naive Bayes classification since our experiments have shown superior performance over other classifiers. The features used are listed in Table 1.

While we aim to detect the correctness of a fact, we consider characteristics of the *source*, that is the Pay-Level-Domain (PLD) from which a fact originates, the *entity description*, the *predicate* term as well as the *fact* itself. From the computed features we train the classifier for classifying the facts from F into the binary labels $\{$*'correct'*, *'incorrect'*$\}$. The *'correct'* facts form a set F_{class} that is the input for the next steps.

Novelty. A fact f is considered to be *novel* with respect to the KBA task, if it fulfills the conditions: i) is not duplicate with other facts selected from our source markup corpus M, ii) is not duplicate with any facts existing in the KB. Each of these two conditions corresponds to a deduplication step.

Deduplication with respect to $\mathbf{M}$ *(*$\mathbf{KnowMore_{ded}}$*).* We detect near-duplicates via clustering. For each predicate p, all the facts $f = \langle p, o_i \rangle$ corresponding to p are clustered into n clusters $\{c_1,\ c_2,\cdots,c_n\}$. Each cluster $c_i, i = 1, ..., n$ contains a set of near-duplicates. To fulfill i), we select only one fact from each cluster by choosing the fact that is closer to the cluster's centroid. This results in the fact set F_{ded} that is the input for next deduplication step.

Deduplication with respect to KB ($\mathbf{KnowMore_{nov}}$*).* We compute the similarity $sim(f_i, f_{KB})$ between a fact f_{KB} in a respective KB for a particular predicate p and a fact f_i for the same (mapped) predicate p in F_{ded} with the datatype-specific similarity metrics. If $sim(f_i, f_{KB})$ is higher than a threshold τ, we remove the fact. We explain τ and its configuration during the experimental Section 4.3. The facts selected from F_{nov} in this step are the final result for augmenting the KB.

Note that our deduplication step considers and supports multi-valued properties. By relying on the clustering features, computed during the fusion step, we select facts from multiple clusters (corresponding to multiple predicates) as long as they are classified as correct. As documented by the evaluation results (Section 5), this does not negatively affect precision while improving recall for multi-valued properties.

4 Experimental Setup

4.1 Ground Truth

We use the WDC2015 dataset[5], where we extracted 2 type-specific subsets consisting of entity descriptions of the *schema.org* types *Movie* and *Book*. As input for the KBA task, we randomly select 30 entities for each type *Book* and *Movie*. We evaluate the performance of our approach for augmenting entity descriptions of these 60 entities obtained from three different KBs: DBpedia (*DB*), Freebase (*FB*) and Wikidata (*WD*). To simplify the schema mapping problem between WDC data and the respective KBs while at the same time taking advantage of the large-scale data available in our corpus, we limit

[5] http://webdatacommons.org/structureddata/index.html#toc3

the task to entities annotated with the *http://schema.org* ontology for this experiment. We manually create a set of schema mappings that maps the *schema.org* vocabularies to the *DB, FB, WD* vocabularies.

Data Fusion - Correctness. We used crowdsourcing to build a ground truth for the correctness of facts $f_i \in F$. For the valid entity descriptions in E, we acquire labels for all distinct facts, as either *correct* or *incorrect* with respect to q. We acquired 5 judgments from distinct workers for each entity and corresponding facts through Crowdflower.

Data Fusion - Novelty. We built ground truths for validating (i) deduplication performance within M, as well as (ii) novelty with respect to the different *KBs*. Authors of this paper acted as experts and designed a coding frame to decide whether a fact is novel. After resolving disagreements on the coding frame on a subset of the data, every fact was associated with one expert label through manual deliberation.

4.2 Metrics

We consider distinct metrics for evaluating each step of our approach.

- $KnowMore_{class}$. We evaluate the performance of the approaches through standard precision P, recall R and $F1$ scores, based on our ground truth.
- $KnowMore_{ded}$. We evaluate the performance of deduplication with respect to M using $Dist\%$ - the percentage of distinct facts within the respective result set. We compare between $Dist\%$ (F_{ded}) and $Dist\%$ (F_{class}), that is, before and after the deduplication within M.
- $KnowMore_{nov}$. For evaluating the performance of deduplication with respect to a given KB, we measure the novelty as Nov - the percentage of novel facts - and compare between Nov (F_{ded}) and Nov (F_{nov}), that is, the novelty before and after this step. We also measure the recall R - the percentage of distinct and accurate facts in F_{ded} that have been selected by $KnowMore_{nov}$ into F_{nov}.

Furthermore, we demonstrate the potential of our approach for augmenting a given KB by measuring the *coverage gain*. The coverage gain of predicate p is computed as the percentage of entity descriptions having p populated through the $KnowMore$ approach (i.e. after step $KnowMore_{nov}$) with at least one fact $\langle p, o \rangle$, out of the ones that did not have statement involving property p within the KB before augmentation.

4.3 Configuration & Baselines

Configuration. For the entity matching step, we use Lucene for indexing and BM25 retrieval with the Lucene default configuration where $k_1 = 1.2$, $b = 0.75$. For the deduplication with respect to KBs, we report the evaluation result of $KnowMore_{nov}$ using different $\tau = \{0.3, 0.5, 0.7\}$ in Section 5.

Baselines. We compare ($KnowMore_{class}$) with $PrecRecCorr$ that is proposed by Pochampally et al. [5] and $CBFS$ [10]. To the best of our knowledge, the $CBFS$ approach is the only available method so far geared towards the challenges of markup data, while $PrecRecCorr$ represents a recent and highly related data fusion baseline.

- $PrecRecCorr$: facts selected based on the approach from candidate set F. We consider each PLD as a source and implemented the *exact solution* as described in the paper. We use the threshold as presented in the paper, i.e. 0.5, to classify facts.

- $CBFS$: facts selected based on the $CBFS$ approach from F. The $CBFS$ approach clusters the associated values at the predicate level into n clusters $(c_1, c_2, \cdots, c_n) \in C$. Facts that are closest to the centroid of each cluster are selected, provided the cluster meet the criteria that its size is larger than half of the largest cluster size.

5 Evaluation Results

Correctness - Data Fusion. The results for $KnowMore_{class}$ and the baselines are shown in Table 2. Our chosen configuration, i.e. using a Naive Bayes classifier achieves highest $F1$ scores among all the different configurations. The presented F1 score of the $PrecRecCorr$ baseline is the best possible configuration for our given task, where we experimented with different thresholds ([0,1], gap 0.1) as discussed in [5] and identified 0.5 experimentally as the best possible configuration. We observe that the $F1$ score of our approach is 0.141 higher than $PrecRecCorr$ and 0.119 higher than $CBFS$ on average across datasets. This indicates that our approach provides the most efficient balance between precision and recall across the investigated datasets. Although, the precision of the baseline approach $PrecRecCorr$ is 0.013 higher than the one from $KnowMore_{class}$ on the *Book* dataset, the baseline fails to recall a large amount of correct facts, where the recall of $KnowMore_{class}$ is 0.388 higher. This also is reflected in the average size of entity descriptions obtained through both approaches, where the entity descriptions from $PrecRecCorr$ consist of 4.88 statements on average, and the ones from $KnowMore_{class}$ are 8.83, indicating a larger potential for the KBA task.

Table 2: Performance of $KnowMore_{class}$ and baselines.

	Movie			**Book**		
Approach	**P**	**R**	**F1**	**P**	**R**	**F1**
$\mathbf{KnowMore_{class}}$	**0.954**	0.896	**0.924**	0.880	**0.868**	**0.874**
PrecRecCorr	0.924	0.861	0.891	**0.893**	0.48	0.624
CBFS	0.802	0.752	0.776	0.733	0.842	0.784

Table 3: Diversity $Dist\%$ before and after deduplication.

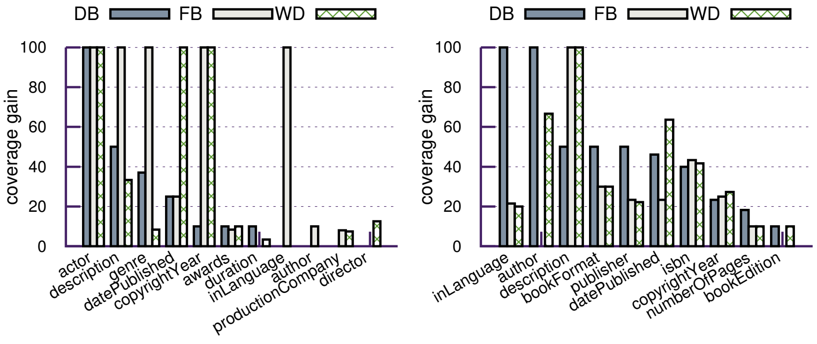
Dataset	$\mathbf{Dist\%(F_{class})}$	$\mathbf{Dist\%(F_{ded})}$
Movie	94.8	**96.1**
Book	82.1	**95.6**

Diversity. Table 3 presents the evaluation result before ($Dist\%$ (F_{class})) and after ($Dist\%$ (F_{ded})) the step $KnowMore_{ded}$. The $Dist\%$ of facts improves by 1.3 percentage points for the *Movie* dataset and by 13.5 percentage points for the *Book* dataset. The less improvement gain for the *Movie* dataset presumably is due to the nature of the randomly selected *Movie* entities. As these appear to be mostly tail entities, candidate facts in our markup corpus M are fewer and less redundant. Hence, the amount of duplicates and near-duplicates is smaller, reducing the effect of the deduplication step.

Table 4: Novelty of F_{ded} and F_{nov} with respect to target KBs.

		Nov				**R**			
	KB	$\mathbf{F_{ded}}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.3}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.5}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.7}$	$\mathbf{F_{ded}}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.3}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.5}$	$\mathbf{F_{nov}}, \tau = \mathbf{0.7}$
Movie	**DBpedia**	0.631	**0.963**	0.962	0.962	1	0.927	**0.939**	**0.939**
	Freebase	0.527	**0.747**	0.742	0.742	1	0.942	**0.957**	**0.957**
	Wikidata	0.412	**0.929**	0.929	0.897	1	**0.963**	**0.963**	**0.963**
Book	**DBpedia**	0.736	**0.962**	0.929	0.92	1	0.826	0.848	**0.870**
	Freebase	0.639	**0.915**	0.846	0.825	1	0.833	**0.846**	**0.846**
	Wikidata	0.705	**0.944**	0.933	0.923	1	0.791	0.814	**0.837**

Novelty with respect to KB. The results before ($Nov\ (F_{ded})$) and after ($Nov\ (F_{nov})$) the deduplication for specific KBs using different similarity thresholds (τ) are presented in Table 4. Since our approach is not aware of the total number of novel facts for a particular entity description on the Web a priori, in this evaluation, we consider all the novel facts in F_{ded} as the gold standard, and compute the recall of F_{nov} after applying the $KnowMore_{nov}$ accordingly. We evaluate the performance of $KnowMore_{nov}$ using τ in $\{0.3, 0.5, 0.7\}$. As shown in Table 4, even though there is a trade-off between *novelty* and *recall*, different values of τ do not have a strong influence on the evaluation metrics. One of the reasons is that, a large proportion of facts have non-literal (e.g. numeric) values. While our datatype-specific similarity computes a binary (0 or 1) score in these cases, it is not influenced by the selection of τ.



(a) Movie

(b) Book

Fig. 2: Proportion of augmented entity descriptions with $KnowMore$.

Coverage Gain. Figure 2 shows the coverage gain on the previously empty slots as shown in Figure 1 per predicate and KB for our selected entities. Based on the result, the $KnowMore$ pipeline shows a coverage gain of 34.75% on average across different properties for DBpedia, 39.42% for Freebase and 36.49% for Wikidata. We observe that the obtained gain varies strongly between predicates and entity types, with a generally higher gain for book-related facts. For instance, within the *Movie* case, for property *actor* we were able to gain 100% coverage in both DBpedia and Freebase, while the property *award* shows a coverage gain of 10% or less for all three KBs. Reasons behind low coverage gain for a particular property are 2-fold: 1) the lack of data in the Web markup data corpus, and 2) the lack of true facts in the real world for a particular attribute, e.g. only a small proportion of movies have won an award. On average, we obtained 2.8 (6.8) facts for each movie (book) entity in our experimental dataset.

6 Conlusions

We have introduced $KnowMore$, an approach towards knowledge base augmentation from large-scale Web markup data. We apply it to the WDC2015 corpus and augment three established KBs. Evaluation results suggest superior performance of our approach with respect to novelty as well as correctness compared to state-of-the-art data fusion baselines. Our experimental results indicate comparably consistent performance across a variety of types, whereas the performance of baseline methods tends to vary strongly.

References

1. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, Vancouver, Canada, 2008. ACM. DOI: 10.1145/1376616.1376746.
2. 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 *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 2014.
3. R. Meusel and H. Paulheim. Heuristics for fixing common errors in deployed schema.org Microdata. In *The Semantic Web. Latest Advances and New Domains: 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 – June 4, 2015. Proceedings*, 2015.
4. R. Meusel, D. Ritze, and H. Paulheim. Towards more accurate statistical profiling of deployed schema.org Microdata. *J. Data and Information Quality*, 8(1):3:1–3:31, Oct. 2016. DOI: 10.1145/2992788.
5. R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 433–444, New York, NY, USA, 2014. ACM. DOI: 10.1145/2588555.2593674.
6. D. Ritze, O. Lehmberg, and C. Bizer. Matching HTML tables to DBpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS '15, pages 10:1–10:6, New York, NY, USA, 2015. ACM. DOI: 10.1145/2797115.2797118.
7. D. Ritze, O. Lehmberg, Y. Oulabi, and C. Bizer. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 251–261, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee. DOI: 10.1145/2872427.2883017.
8. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. DOI: 10.1145/1242572.1242667.
9. R. Yu, B. Fetahu, U. Gadiraju, and S. Dietze. A survey on challenges in Web markup data for entity retrieval. In *International Semantic Web Conference (Posters & Demos), Kobe, Japan, October 17-21*, 2016.
10. R. Yu, U. Gadiraju, X. Zhu, B. Fetahu, and S. Dietze. Towards entity summarisation on structured web markup. In H. Sack, G. Rizzo, N. Steinmetz, D. Mladenić, S. Auer, and C. Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers*, pages 69–73, Cham, 2016. Springer International Publishing. DOI:10.1007/978-3-319-47602-5_15.