Empirical Best Practices On Using Product-Specific Schema.org

Mayank Kejriwal,[1] **Ravi Kiran Selvam,** [1] **Chien-Chun Ni,** [2] **Nicolas Torzec** [2]
[1] University of Southern California
[2] Verizon Media
kejriwal@isi.edu, rselvam@isi.edu, chien-chun.ni@verizonmedia.com, torzecn@verizonmedia.com

Abstract

Schema.org has experienced high growth in recent years. Structured descriptions of products embedded in HTML pages are now not uncommon, especially on e-commerce websites. The Web Data Commons (WDC) project has extracted schema.org data at scale from webpages in the Common Crawl and made it available as an RDF 'knowledge graph' at scale. The portion of this data that specifically describes products offers a golden opportunity for researchers and small companies to leverage it for analytics and downstream applications. Yet, because of the broad and expansive scope of this data, it is not evident whether the data is usable in its raw form. In this paper, we do a detailed empirical study on the product-specific schema.org data made available by WDC. Rather than simple analysis, the goal of our study is to devise an empirically grounded set of best practices for using and consuming WDC product-specific schema.org data. Our studies reveal five best practices, each of which is justified by experimental data and analysis.

Introduction

Structured data has continued to play an increasingly important role for Web search and applications. In fact, according to (Cafarella, Halevy, and Madhavan 2011), the expanding quantity and heterogeneity of Web structured data has enabled new solutions to problems, especially concerning search engine optimization (SEO) and data integration spanning multiple web sources. Two application areas that have benefited greatly from structured data are e-commerce and advertisements. Because of structured data and markup, it is now easier than ever both to advertise and find products on the Web, in no small part due to the ability of search engines to make good use of this data. There is also limited evidence that structured data plays a key role in populating Web-scale knowledge graphs such as the Google Knowledge Graph that are essential to modern semantic search (Singhal 2012).

However, even though most e-commerce platforms have their own proprietary datasets, some resources do exist for smaller companies, researchers and organizations. A particularly important source of data is schema.org markup in webpages. Launched in the early 2010s by major search engines such as Google and Bing, schema.org was designed to facilitate structured (and even knowledge graph) search applications on the Web. The Web Data Commons (WDC) project has crawled increasing amounts of schema.org data in recent years (Mühleisen and Bizer 2012), including in the e-commerce and products domain. WDC schema.org data has broad coverage at the level of 'pay-level domains' (such as rakuten.com), languages and product categories, providing a golden opportunity for researchers to use this data in downstream applications and analyses.

Yet, there are also considerable challenges in using this data. Not including noise due to variations and misspellings, language tags on text literals may be inaccurate, and there may be skew both in the distribution of languages and pay-level domains. Some of the properties may have less semantic validity than others. All of this is further made difficult by the fact that WDC product-specific schema.org datasets are non-trivial in size (with each year's data comprising hundreds of gigabytes even in compressed format), which precludes significant manual processing and labeling (or 'eyeballing'). What is needed and is currently lacking is a set of empirically grounded best practices for consuming product-specific schema.org data released by the WDC. These best practices, once determined, could then be implemented in practice by any engineer looking to consume these datasets in applications and processes of their own.

In this paper, we conduct a series of empirical studies to describe and justify such a set of practices. We use 2018 product-specific schema.org WDC for this purpose. In total, we conduct three broad studies investigating issues ranging from skewness in language distributions to data localization. From our experiments, we distill five best practices for both researchers and practitioners. We show that, while extremely promising, schema.org data from WDC should be used while keeping these best practices in mind to avoid issues of quality, scale and bias. Future work may further refine these practices and supplement them significantly.

Related Work

The research presented in this paper is related to several of the existing lines of work that we briefly describe below. There has been a considerable amount of work on schema.org already, both in describing its principles and

its evolution. For example, the authors in (Patel-Schneider 2014) describe the core principles behind a plausible version of schema.org and state the formal semantics of using schema.org. In a related, but different vein, the authors in (Meusel, Bizer, and Paulheim 2015) perform large scale analysis of the usage of schema.org vocabularies over time.

A more closely related work is (Meusel and Paulheim 2015), which describes the set of simple heuristics that could be applied to WDC microdata (Meusel, Petrovski, and Bizer 2014) so that consumers can use them to fix common mistakes as a post-processing step. The authors of (Kärle et al. 2016) demonstrate a similar analysis of the validity of schema.org concepts in the hotel domain.

Good example of work that is more e-commerce oriented is (Petrovski, Bryl, and Bizer 2014), which describes the task of integrating the descriptions of electronic products from websites that use microdata markup to represent information and the various challenges that the authors faced. Yet another example is (Ristoski et al. 2018), which uses the structured data from the web as supervision for training feature extraction models to extract attribute-value pairs from textual descriptions of products.

Other examples of work that are related to schema.org analysis but that are too extensive to describe here include (Nam and Kejriwal 2018), (Mihindukulasooriya et al. 2017), (Beek et al. 2014), (Abedjan et al. 2014), (Abedjan, Lorey, and Naumann 2012) and (Meusel, Ritze, and Paulheim 2016).

Raw Data

As mentioned in the introduction, *schema.org* (R. V. Guha 2016) is a collaborative effort by major search engines such as Google, Yahoo, Microsoft, Yandex and open community members to create, maintain and promote schemas for publishing embedded structured data on webpages. Schema.org has vocabularies that support different encoding schemes like microdata, RDFa and JSON-LD. Schema.org vocabularies are used by more than 10 million websites to add markup to their webpages. Schema.org markup helps the search engines better understand the information present in webpages much better, which in turn facilitates richer search experiences for search engine users.

The full extent of schema.org on the Web may not be known to any individual or organization beyond a large search engine such as Google. The Common Crawl is an initiative to allow researchers and the general public to have access to reasonably high-quality crawl data that was previously only available to major search engines. The schema.org portion of the Web Data Commons (WDC) project supports researchers and companies in exploiting the structured information available on the Web by extracting schema.org and other kinds of structured data from the webpages in the Common Crawl and making the data available. Conveniently, WDC also provides *class-specific* subsets of the extraction corpus for a selection of schema.org classes. Such subsets only (or mostly) contain instances of a specific class (e.g., Products, Books, Movies etc.) which is especially convenient for domain-specific analysis.

Entity Class	Entity Count
http://schema.org/Product	307.3 M
http://schema.org/Offer	236.3 M
http://schema.org/ListItem	65.7 M
http://data-vocabulary.org/Breadcrumb	45.5 M
http://schema.org/AggregateRating	30.4 M

Table 1: Breakdown of the nodes by the *Entity* class (only the 5 most frequent entity classes are shown).

We used the *Product-specific* subset of the schema.org data contained in the November 2018 version[1] of the Web Data Commons Microdata dataset. We will refer to this dataset as the *Product dataset* in subsequent sections. The size of this dataset is roughly 112.7 GB in compressed form. The dataset may be downloaded as chunks with each compressed chunk being of size 1.2 GB. This dataset contains around 4.8 billion quads, 7.4 million URLs and 92,000 hosts. The top classes which are present in this dataset are shown in the Table 1.

Empirical Study

This section demonstrates the experiments conducted to study the Product dataset for determining the best practices to utilize the schema.org dataset for downstream tasks.

Data Skewness

Not all the languages are equally represented in the Product dataset. It may be critical for certain NLP and multi-lingual applications to understand the distribution of languages in the dataset. The dataset can be *skewed* such that it has a significant portion of triples with text literals in a particular language, without also having a reasonable proportion of triples with text literals in other languages. This skew in language distribution might bias the results of downstream tasks that expect data in multiple languages with similar representation as in the real world (or the broader Web). We note that, like the rest of the experiments in this paper, such a skew does not necessarily mean that the entire schema.org component of the Web is skewed; only that the WDC Product dataset that we are studying is skewed. This is important to remember due to the importance of WDC in several large-scale studies involving this kind of Web data.

Furthermore, skew may arise not just at the language level but also at the level of *pay-level domains* associated with particular types of products (e.g., the pay-level domain *fineartamerica.com* lists only products related to art like paintings and home decor as opposed to clothing or electronics). This skew in pay-level domain distribution may bias the results of downstream tasks which need product data to be equally distributed among different categories.

To study issues of skewness in the Product dataset, we first parse and extract the language tags that are explicitly associated with text literals by virtue of being represented in RDF. We found 1,072 unique language tags in the dataset.

[1] http://webdatacommons.org/structureddata/2018-12/stats/stats.html

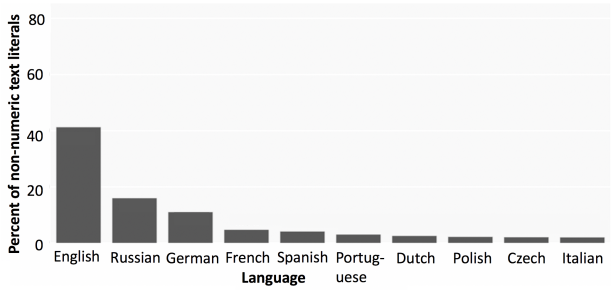

Figure 1: Breakdown of the non-numeric text literals by associated language tags (only the 10 most frequent languages are shown). A non-numeric text literal is one where at least 50% of the characters are non-digits.

We observed that many language tags are simply variations of the same language but associated with different countries. We reconciled all those language tags which represented the same language into a single cluster, yielding 249 unique language 'clusters'. From Figure 1, we observe that the dataset is skewed towards having higher numbers of triples with text literals in English, followed by Russian and German.

We investigated further to check if the language tags associated with text literals are consistent with the *actual* language of the text literals, since the explicit tags could be incorrect. We randomly sampled a small subset of triples containing text literals and examined them manually. We found some evidence of disagreement between the actual language of the text literals and their associated language tags. Since it is not possible to manually peruse the whole dataset, we opted to design an alternate experiment to test inaccuracy in language tag declarations.

Specifically, we used a pre-trained fastText-based language identifier model[2] which can recognize 176 languages in a fast and accurate manner. The language identifier was trained on data from Wikipedia, Tatoeba, and SETimes and achieved more than 93 % accuracy on many standard language identification datasets from Wikipedia, TCL and EuroGov. Since the dataset is still too large for all text literals to be tagged using fastText-based language identifier without expending considerable computational resources, we randomly sampled 1/100th of the total number of text literals in each chunk of the dataset, yielding 100,000 samples per chunk. With 97 chunks, the total number of samples considered this evaluation is 10 million, which is large enough to draw reasonable conclusions. We also found many text literals had numeric data like prices, dates, phone numbers, dimensions, model numbers, and time intervals. These numeric text literals are not associated with any particular language and will cause obvious problems in estimating the level of agreement between explicitly declared language tags and the outputs of the automatic language identifier. Hence, we removed these numeric text literals (text literals having at least 50% of the characters as digits) from our evaluation

[2]The pretrained model can be downloaded from https://fasttext.cc/docs/en/language-identification.html

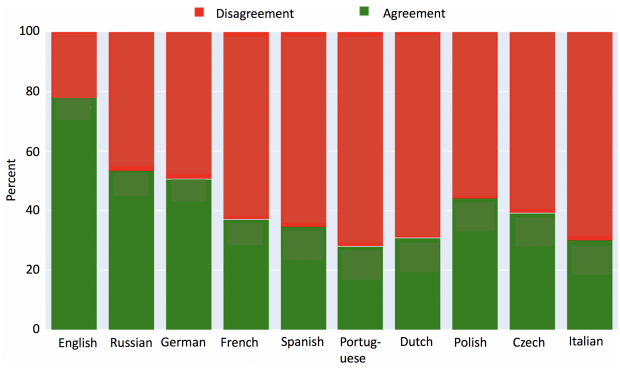

Figure 2: Breakdown of 'Agreement' vs. 'Disagreement' proportions for each language, with 'Agreement' of a language defined as the percent of non-numeric text literals with an associated language tag that is consistent with the actual language of the text literals.

by conducting some preprocessing.

Figure 2 shows the levels of agreement and disagreement for the 10 most popular languages found in the dataset. We find that disagreement is much higher for some languages than others, but all languages have a non-trivial share of disagreement. In other words, for any quality-critical application, explicit language tags should not be directly trusted. Since the automatic identifier itself is not perfect, we recommend using only that subset of triples (or triples with text literals) where there is agreement between the automatically identified, and explicitly declared, language tags.

To study skewness at the level of pay-level domains, we attempted to determine if the most common websites have an impact on the type of product nodes that are present in the entire dataset. We found that the dataset contains nodes extracted from around 1 million pay-level domains and is not skewed towards having a significant portion of nodes associated with a particular pay-level domain. As can be seen from the ten common pay-level domains in Figure 3, node count associated with each of the ten pay-level domains obeys a roughly linear distribution, unlike long-tailed or power-law distributions that may have been expected in some other domains. We further examined whether the nodes are crawled from trusted pay-level domains. A pay-level domain is defined by us as *trusted domain* if it has low Google PageRank, since the PageRank judges the "value of a page" by measuring the quality and quantity of other pages that link to it. The main purpose of PageRank is to determine the relative importance or relative trust of a given webpage. A webpage having a low PageRank is relatively more important or trustworthy than a webpage having high PageRank. Hence, we computed the Google PageRanks[3] for the 10 common pay-level-domains, with results shown in Figure 3.

[3]The Google PageRank was computed using the Open PageRank API, https://www.domcop.com/openpagerank/what-is-openpagerank

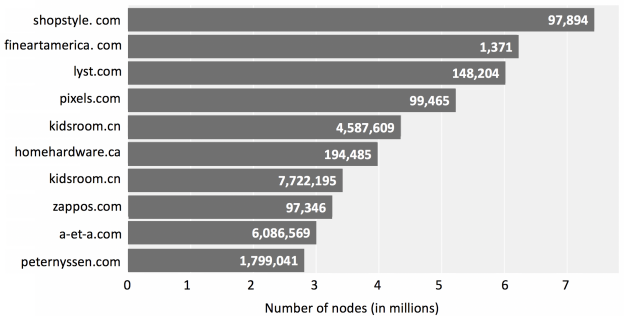

Figure 3: Node counts associated with each pay-level domain (e.g., peternyssen.com), along with the Google PageRank for that domain (within the bar). Only the 10 most frequent pay-level domains are shown.

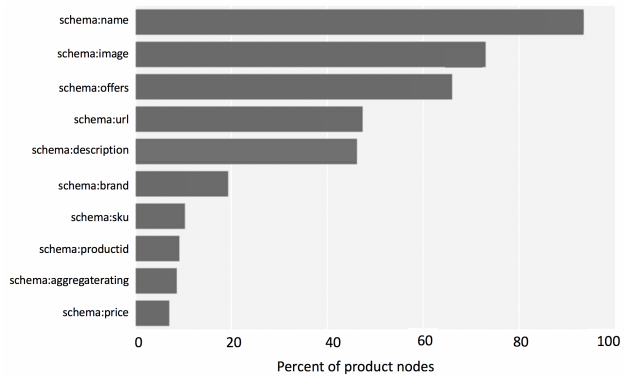

Figure 4: Percent of product nodes associated with a product property (only the 10 most frequent properties are shown). We use *schema:* as a shorthand prefix for http://schema.org/product/.

Characterization of Product Data

The websites that embed the information using schema.org markup are used for extraction to create the Product dataset. We hypothesize that not all the information that is present in the Product dataset is *semantically valid*. For example, the product name property which is extracted for a particular product node might not be valid if it contains "Null" or "N/A" or if it contains another piece of information such as the URL (rather than the literal representing the name of the product). In this section, we design, and present findings on, a preliminary study to study this issue further, as semantic validity is also important for quality-critical applications.

Specifically, we study the semantic validity of properties of product nodes based on heuristic constraints. By heuristic constraints, we mean conditions that we intuitively associate with both the type of the property value or *object* of the product property (e.g., text literal for some property values, vs. node or even URL as object). We manually determined a set of heuristic rules to check validity of the product properties by randomly sampling a subset of triples associated with each product property and examining them[4]. Since the number of unique product properties is very large, it is not practical to examine all the product properties; hence we limited our exploration to the ten most frequently occurring product properties. Figure 4 describes the percent of product nodes associated with each of the 10 most frequent product properties.

While exploring the sample triples associated with product properties, we observed another kind of invalidity for certain properties like "product name" and "brand". Specifically, we found that even when the product property satisfies the heuristic rules, the object still contains information that is associated with another property instead of the stated property. For example, we observed certain triples that contained a product ID or product description as objects of the "product name" property. These kinds of semantic errors are much harder to detect, and are left for future study.

Furthermore, we define a product *node* to be valid if it contains at least five valid properties out of the ten most frequently occurring properties and satisfies the heuristic rule that the length of the preprocessed text literal associated with product name property is less than the length of the preprocessed text literal associated with product description property. Note that, in preprocessing text literals, we take standard steps such as removing extraneous white spaces such as tabs, new lines and other such characters. We found over 32 million valid product nodes (which amount to 10.66% of the total product nodes in the Product dataset) that satisfied our manually determined heuristic constraint for being a valid product node. Figure 5 describes the percent of product nodes associated with property values that are determined to be valid (by satisfying the manually determined heuristic rules). In looking at Figure 5, we find that, while all properties are more valid than invalid on average, there are some properties that are not as untrustworthy as others. For example, the property expressing the SKU[5] of a product has an 'invalid' percentage of 38, while the product's URL has an invalid percentage of only 11.15. This suggests that schema.org publishers are less careful and rigorous about publishing some properties than others. Any experiments that try to use such data for tasks like distant or weak supervision, have to keep this factor in mind, since some of the properties may end up introducing noise.

Overall Data Completeness

When describing the raw data, we noted that even in compressed form, the entire Product dataset can be well over 100 GB. While this is easy to handle using a moderately sized Hadoop cluster, it is big enough that individual researchers and smaller organizations without access to such clusters (or operating on tight budgets) will be looking for approximate ways of extracting product nodes and their properties.

[4]These heuristics are not reproduced here due to lack of space, but we can reproduce them in an appendix or website upon request. An example of such a heuristic constraint is: an 'aggregaterating' property must not contain as its value a text literal, 'N/A' or 'NULL', nor must it be an RDF node identifier. This constraint makes intuitive sense since valid 'aggregaterating' property values are real numbers.

[5]Stock Keeping Unit.

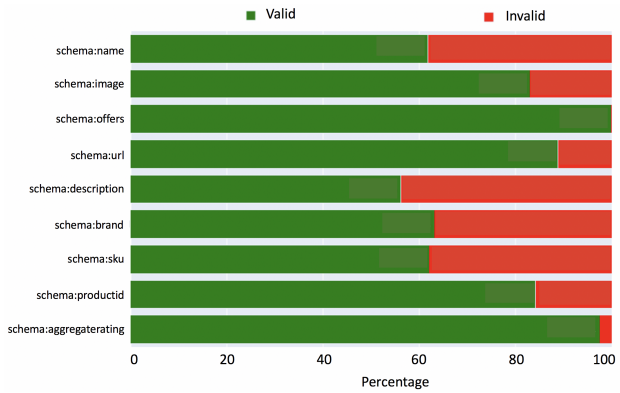

Figure 5: Breakdown of valid vs. invalid product node percentages for each product property. 'Valid product node percentage' is defined as the percent of product nodes with a property value that satisfies the heuristic rules of validity determined for that property. Note that only the 10 most frequent product properties (except the "product price" property) are shown in the figure.

In fact, the observation that all the triples associated with a particular product node do not all appear *together* in the Product dataset and may be spread out in the dataset incurs significant processing challenges in a non-parallel architecture.

For certain downstream tasks, one may need all the triples associated with a particular node to be in memory. The easiest solution is to store the entire dataset in memory but this is not practical due to dataset size. To overcome this problem, we may need to sequentially read the triples associated with nodes from the dataset and have a window with a suitable size to ensure that we have 'captured' all the triples associated with a particular node within that window.

More specifically, to find the extent of the 'spread' of the triples associated with a particular node, we adopt the following mechanism. We attempt to discover if there is a *window* of sufficiently small size within which we could find all the triples associated with a given node starting from the first triple associated with that node, at least most of the time. The value of this size would then tell us whether the dataset has sufficient *localization* (i.e. less spread in the sense defined above).

To study localization in the Product dataset, we computed the minimum such window size for every product node. The distribution of these window sizes associated with all the nodes in the Product dataset can be found in the Figure 6. The average window size is found to be 27 and the 99th percentile value of window size distribution occurs at size 145. In other words, if we slide through the Product dataset with a window of size 145, starting from the first triple associated with a given node, we will be able to find all the triples associated with that node within the window containing the next 145 triples (at least for 99% of the cases).

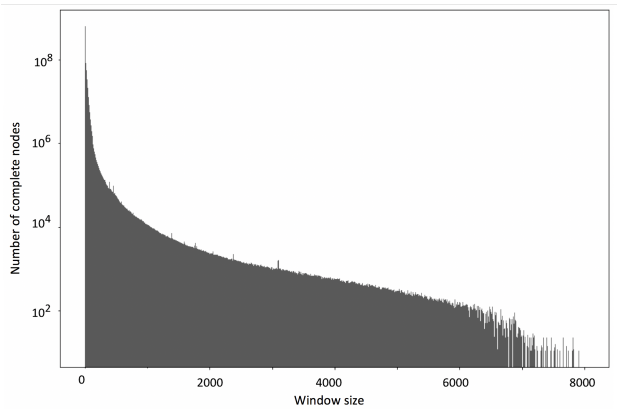

Figure 6: Number of complete nodes within the given window size. A complete node is one which has all triples associated with it occurring within a particular window size.

Summary of Best Practices

This section succinctly lists some of the best practices determined through the previously described empirical studies for using the Product dataset in downstream applications and analyses.

- If the downstream task is highly dependent on the language of the dataset, we recommend not directly trusting the language tags associated with the text literals. Our experiments showed that using even a relatively straightforward language identifier model such as the one built using common fastText embeddings (to identify the language of a given text) may be more trustworthy than the explicit tag.
- For downstream tasks that need product nodes from different categories, we recommend selecting the subset of pay-level domains that cover all categories of products and extract products only from the triples associated with those pay-level domains. Experimental data shows that the distribution of pay-level domains is not skewed[6] and downstream applications that use the data associated with such subsets have less likelihood of being biased.
- If there is a need to extract trusted data from the Product dataset, we recommend limiting the extraction of information from triples associated with pay-level domains having a low PageRank. Low PageRank for a pay-level domain indicates that it is of high importance and trust. A simple API can be used to get PageRank for pay-level domains.
- We recommend running simple heuristic checks to confirm semantic validity of properties associated with the product nodes before ingesting property values ('objects') in downstream tasks. While our rules were manually determined, there are opportunities for discovering other

[6]However, it must also be noted again that this is not a general truth about 'e-commerce' market share, only about the schema.org data. The lack of Amazon's data in schema.org may also account for why we observe less of a winner-takes-all curve.

such rules (especially automatically) exist in future research.

- To address concerns relating to Big Data, we recommend using a window of size 145 or more while sliding through the triples of the Product dataset. Our experiments show that having a window size of 145 ensures that we find all the properties associated with a given node within that window in 99% of the cases. By following this best practice, the use of expensive computing infrastructure and setup can be avoided.

Conclusion

Schema.org and structured data have become highly significant in recent times. In this paper, we studied a product-specific schema.org dataset made recently available by the Web Data Commons project, and used a set of carefully designed empirical studies to devise a set of best practices that could be used by industry to extract value from the raw data. We noted several issues that must be borne in mind by quality-conscious practitioners seeking to use this data, including semantic validity of properties and disagreement between explicitly stated language tags and the actual language of the underlying text literals. We recommended a set of best practices based on these findings. For example, we noted that, if the downstream task is highly dependent on the language of the dataset, directly trusting the tags is not the best course of action. Instead, one may want to use an ensemble based both on explicit tags and an automated language detection algorithm. Future work may reveal other best practices that could supplement our own and that could lead to more industrial adoption of schema.org.

References

Abedjan, Z.; Gruetze, T.; Jentzsch, A.; and Naumann, F. 2014. Profiling and mining RDF data with ProLOD++. In *2014 IEEE 30th International Conference on Data Engineering*, 1198–1201.

Abedjan, Z.; Lorey, J.; and Naumann, F. 2012. Reconciling Ontologies and the Web of Data. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM 12, 15321536. New York, NY, USA: Association for Computing Machinery. ISBN 9781450311564. doi:10.1145/2396761.2398467. URL https://doi.org/10.1145/2396761.2398467.

Beek, W.; Rietveld, L.; Bazoobandi, H. R.; Wielemaker, J.; and Schlobach, S. 2014. LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. In *The Semantic Web – ISWC 2014*, 213–228. Cham: Springer International Publishing. ISBN 978-3-319-11964-9.

Cafarella, M. J.; Halevy, A.; and Madhavan, J. 2011. Structured data on the web. *Communications of the ACM* 54(2): 72–79.

Kärle, E.; Fensel, A.; Toma, I.; and Fensel, D. 2016. Why Are There More Hotels in Tyrol than in Austria? Analyzing Schema.org Usage in the Hotel Domain. In *Information and Communication Technologies in Tourism 2016*, 99–112. Cham: Springer International Publishing. ISBN 978-3-319-28231-2.

Meusel, R.; Bizer, C.; and Paulheim, H. 2015. A Web-Scale Study of the Adoption and Evolution of the Schema.Org Vocabulary over Time. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS 15. New York, NY, USA: Association for Computing Machinery. ISBN 9781450332934. doi:10.1145/2797115.2797124.

Meusel, R.; and Paulheim, H. 2015. Heuristics for Fixing Common Errors in Deployed schema.org Microdata. In *The Semantic Web. Latest Advances and New Domains*, 152–168. Cham: Springer International Publishing. ISBN 978-3-319-18818-8.

Meusel, R.; Petrovski, P.; and Bizer, C. 2014. The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In *The Semantic Web – ISWC 2014*, 277–292. Cham: Springer International Publishing. ISBN 978-3-319-11964-9.

Meusel, R.; Ritze, D.; and Paulheim, H. 2016. Towards More Accurate Statistical Profiling of Deployed Schema.Org Microdata. *J. Data and Information Quality* 8(1). ISSN 1936-1955. doi:10.1145/2992788. URL https://doi.org/10.1145/2992788.

Mihindukulasooriya, N.; Poveda-Villalón, M.; García-Castro, R.; and Gómez-Pérez, A. 2017. Collaborative Ontology Evolution and Data Quality - An Empirical Analysis. In *OWL: Experiences and Directions – Reasoner Evaluation*, 95–114. Cham: Springer International Publishing. ISBN 978-3-319-54627-8.

Mühleisen, H.; and Bizer, C. 2012. Web Data Commons-Extracting Structured Data from Two Large Web Corpora. *LDOW* 937: 133–145.

Nam, D.; and Kejriwal, M. 2018. How Do Organizations Publish Semantic Markup? Three Case Studies Using Public Schema.org Crawls. *Computer* 51(6): 42–51.

Patel-Schneider, P. F. 2014. Analyzing Schema.org. In *The Semantic Web – ISWC 2014*, 261–276. Cham: Springer International Publishing. ISBN 978-3-319-11964-9.

Petrovski, P.; Bryl, V.; and Bizer, C. 2014. Integrating Product Data from Websites Offering Microdata Markup. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW 14 Companion, 12991304. Association for Computing Machinery. ISBN 9781450327459. doi:10.1145/2567948.2579704.

R. V. Guha, Dan Brickley, S. M. 2016. Schema.org: evolution of structured data on the web. *Communications of the ACM* 59(2): 44–51.

Ristoski, P.; Petrovski, P.; Mika, P.; and Paulheim, H. 2018. A machine learning approach for product matching and categorization. *Semantic Web* 9(5): 707–728.

Singhal, A. 2012. Introducing the knowledge graph: things, not strings. *Official google blog* 16.