Towards Profiling Knowledge Graphs

Heiko Paulheim

University of Mannheim, Data and Web Science Group

Abstract. Knowledge Graphs, such as DBpedia, YAGO, or Wikidata, are valuable resources for building intelligent applications like data analytics tools or recommender systems. Understanding what is in those knowledge graphs is a crucial prerequisite for selecing a Knowledge Graph for a task at hand. Hence, Knowledge Graph profiling - i.e., quantifying the structure and contents of knowledge graphs, as well as their differences - is essential for fully utilizing the power of Knowledge Graphs. In this paper, I will discuss methods for Knowledge Graph profiling, depict crucial differences of the big, well-known Knowledge Graphs, like DBpedia, YAGO, and Wikidata, and throw a glance at current developments of new, complementary Knowledge Graphs such as *DBkWik* and *WebIsALOD*.

1 Knowledge Graphs on the Web

The term "Knowledge Graph" was coined by Google when they introduced their knowledge graph as a backbone of a new Web search strategy in 2012, i.e., moving from pure text processing to a more symbolic representation of knowledge, using the slogan "things, not strings"[1].

Various public knowledge graphs are available on the Web, including DBpedia [14] and YAGO [26], both of which are created by extracting information from Wikipedia (the latter exploiting WordNet on top), the community edited Wikidata [27], which imports other datasets, e.g., from national libraries[2], as well as from the discontinued Freebase [19], the expert curated OpenCyc [15], and NELL [4], which exploits pattern-based knowledge extraction from a large Web corpus.

Furthermore, company-owned knowledge graphs exist, like the already mentioned Google Knowledge Graph, Google's Knowledge Vault [5], Yahoo's Knowledge Graph [2], Microsoft's Satori, and Facebook's Knowledge Graph. However, those are not publicly available, and hence neither suited to build applications by parties other than the owners, nor can they be analyzed in depth. The degree to which their size, structure, contents, and quality are known varies.

Although all these knowledge graphs contain a lot of valuable information, choosing one KG for building a specific application is not a straight forward

[1] `https://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html`

[2] `https://www.wikidata.org/wiki/Wikidata:Data_donation`

task. Depending on the domain and task at hand, some KGs might be better suited than others. However, there are no guidelines or best practices on how to choose a knowledge graph which fits a given problem. Previous works mostly report global numbers, such as the overall size of knowledge graphs, such as [16], and focus on other aspects, such as data quality [9]. In [20], we have taken a more in-depth look, showing detailed results for different classes.

2 Measures and Methods for Knowledge Graph Profiling

In general, the aim of knowledge graph profiling is to understand whether a given knowledge graph suits a certain purpose. For example, for building an application for a specific domain, backed by a knowledge graph, requires that this knowledge graph contains a reasonable amount of information about the entities in that domain, and describes them at a suitable level of detail.

There is a large body of work on measures and methods for dataset profiling for knowledge graphs and Linked datasets [6]. For analyzing knowledge graphs, we use the following three classes of metrics:

Global Measures describe the knowledge graph as a whole,

Class-based Measures describe the characteristics of entities in a given class, and

Overlap Measures describe the difference between two or more knowledge graphs.

2.1 Global Measures

The most basic question to ask about a knowledge graph is: *How large is it?* Hence, we can count the number of instances and assertions, i.e., relations between two entities, or relations of an entity to a literal value.

Second, we are often interested in the *level of detail* at which entities are are described in a knowledge graph. This is usually computed as the average *degree*, i.e., the average number of ingoing and/or outgoing edges of a node. Aside from looking at *averages*, it is often interesting to also consider the *median*, which may give a more realistic picture of the level of detail for an average entity.

For entities, relations, and degrees, one can often find different numbers in different reports. This is due to the fact that there are some methodological differences. For *entities*, some reports only count explicitly typed resources, while others count all nodes in the graph. For relations, some reports count literal assertions as well, while others do not. Furthermore, reports may differ in taking into account special relations (e.g., `owl:sameAs`), while others do not, which may have an influence on the degrees reported. Finally, some reports treat schema and instances separately, while others count, e.g., classes and instances alike when reporting the number of entities in a graph.

Finally, the timeliness of a knowledge graph can also be relevant. While some knowledge graphs are no longer developed any further, meaning that their contents become more and more outdated, others have regular – shorter or longer – release cycles, or even provide live data.

2.2 Class-based Measures

For class-based measures, the same metrics as for global measures can be used, i.e., how many entities exist in a certain class, and at which level of detail are they described?

There are two problems that may arise when counting the number of instances in a class, and reporting that number as *There are are N entities of type X in this knowledge graph*. First, the type assertions in a knowledge graph are not guaranteed to be complete. In fact, in [17], we presented an estimate of the number of missing type assertions in DBpedia. By comparing two knowledge graphs – i.e., DBpedia and YAGO – and counting the untyped instances in DBpedia that have a type in YAGO which has a corresponding type in the DBpedia ontology, we found that DBpedia has at least 2.6M missing type assertions. Hence, counting type instances based on type assertions often underestimates the actual counts.

The second problem occurs with modeling issues. For example, instances are usually counted based on asserted types, but different knowledge graphs have different modeling paradigms. For example, DBpedia and YAGO define classes for occupations of people (e.g., *Actor* or *Politician*), while Wikidata models those as a relation linking a person to a profession, while the person is only assigned the less specific type *Person*. Those complex mappings are not always easy to obtain and utilize when comparing the number of entities in a given class across knowledge graphs.

2.3 Overlap Measures

To quantify the similarity and difference of knowledge graphs, one has to analyze their overlap, i.e., the amount of instances they have in common. Although many knowledge graphs are served as Linked Open Data [1], using interlinks on instance level with `owl:sameAs`, those interlinks are not necessarily complete, i.e., the *Open World Assumption*, which holds to Web knowledge graphs in general, also holds for their interlinks. Hence, they cannot be utilized directly as a measure for quantifying the overlap between two knowledge graphs. For example, from the fact that 2,000 cities in knowledge graph A are linked to cities in knowledge graph B, we cannot simply conclude that this is the number of cities contained in the intersection of A and B.

In order to estimate the actual overlap based on explicit interlinks, we use an approach first described in [20]. We first find interlinks between two knowledge graphs using an arbitrary linkage rule, e.g., interlinking all entities with the same name.

Then, using the existing interlinks, we compute the quality of a linking approach in terms of recall and precision. Given that the actual number of links is C, the number of links found by a linkage rule is F, and that the number of correct links in F is F^+, recall and precision are defined as

$$R := \frac{|F^+|}{|C|} \tag{1}$$

$$P := \frac{|F^+|}{|F|} \tag{2}$$

By resolving both to $|F^+|$ and combining the equations, we can estimate $|C|$ as

$$|C| = |F| \cdot P \cdot \frac{1}{R} \tag{3}$$

Note that in the latter formula, all variables on the right hand side – the total number of interlinks found by a linkage rule, as well as its recall and precision – are known (which is not true for F^+ in the two formulas above). For a stable estimate, we use a variety of different linkage rules, and average their estimates.

As for class-based measures, we can quantify the overlap per class, e.g., finding all persons that are contained in two knowledge graphs. Again, this may be biased by missing type statements in the knowledge graphs, but usually provides a decent approximation.

3 Global Measures: Overall Size and Shape of Knowledge Graphs

For the analysis in this paper, we focus on the public knowledge graphs DBpedia, YAGO, Wikidata, OpenCyc, and NELL.[3,4] For those five KGs, we used the most recent available versions at the time of this analysis, as shown in Table 1.

We can observe that DBpedia and YAGO have roughly the same number of instances, which is not surprising, due to their construction process, which creates an instance per Wikipedia page. Wikidata, which uses additional sources plus a community editing process, has about tree times more instances. It is remarkable that YAGO and Wikidata have roughly the same number of axioms, although Wikidata has three times more instances. This hints at a higher level of detail in YAGO, which is also reflected in the degree distributions.

OpenCyc and NELL are much smaller. NELL is particularly smaller w.r.t. axioms, not instances, i.e., the graph is less dense. This is also reflected in the degree of instances, which depicts that on average, each instance has less than seven connections. The other graphs are much denser, e.g., each instance in Wikidata has about 50 connections on average, each instance in DBpedia has about 60, and each instance in YAGO has even about 120 connections on average.

The number of entities and the degrees are not independent. There are certain effects caused by the distribution of entities contained in the different graphs: While OpenCyc contains mostly head entities, DBpedia, YAGO, and Wikidata have a larger coverage of tail entities as well. The head entities are actually described in the larger knowledge graphs at much more detail than in the smaller ones, but the overall degree distribution is rather skewed, which leads to lower averages.

[3] Freebase was discarded as it is discontinued, and non-public KGs were not considered, as it is impossible to run the analysis on non-public data.

[4] Scripts are available at `https://github.com/dringler/KnowledgeGraphAnalysis`.

Table 1: Global Properties of the Knowledge Graphs compared in this paper [20]

	DBpedia	YAGO	Wikidata	OpenCyc	NELL
Version	2016-04	YAGO3	2016-08-01	2016-09-05	08m.995
# instances	5,109,890	5,130,031	17,581,152	118,125	1,974,297
# axioms	397,831,457	1,435,808,056	1,633,309,138	2,413,894	3,402,971
avg. indegree	13.52	17.44	9.83	10.03	5.33
avg. outdegree	47.55	101.86	41.25	9.23	1.25
# classes	754	576,331	30,765	116,822	290
# relations	3,555	93,659	11,053	165	1,334
Releases	biyearly	> 1 year	live	> 1 year	1-2 days

The schema sizes also differ widely. In particular the number of classes are very different. This can be explained by different modeling styles: YAGO automatically generates very fine-grained classes, based on Wikipedia categories. Those are often complex types encoding various facts, such as "American Rock Keyboardists". KGs like DBpedia or NELL, on the other hand, use well-defined, manually curated ontologies with much fewer classes.

Since Wikidata provides live updates, it is the most timely source (together with DBpedia Live, which is a variant of DBpedia fed from an update stream of Wikipedia [11]). From the non-live sources, NELL has the fastest release cycle, providing a new release every few days. However, NELL uses a fixed corpus of Web pages, which is not updated as regularly. Thus, the short release cycles do not necessarily lead to more timely information. DBpedia has biyearly releases, and YAGO and OpenCyc have update cycles longer than a year.

4 Class-based Measures: Looking into Details

When building an intelligent, knowledge graph backed application for a specific use case, it is important to know how fit a given knowledge graph is for the domain and task at hand. To answer this question, we have picked 25 popular classes in the five knowledge graphs and performed an in-depth comparison. For those, we computed the total number of instances in the different graphs, as well as the average in and out degree. The results are depicted in figure 2.

While DBpedia and YAGO, both derived from Wikipedia, are rather comparable, there are notable differences in coverage, in particular for events, where the number of events in YAGO is more than five times larger than the number in DBpedia. On the other hand, DBpedia has information about four times as many settlements (i.e., cities, towns, and villages) as YAGO. Furthermore, the level of detail provided in YAGO is usually a bit larger than DBpedia.

The other three graphs differ a lot more. Wikidata contains twice as many persons as DBpedia and YAGO, and also outnumbers them in music albums and books. Furthermore, it provides a higher level of detail for chemical substances and particularly countries. On the other hand, there are also classes which are

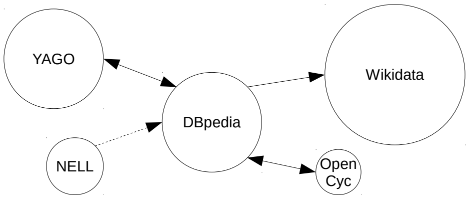


Fig. 1: Knowledge Graphs inspected in this paper, and their interlinks. Like for the Linked Open Data Cloud diagrams [24], the size of the circles reflects the number of instances in the graph (except for OpenCyc, which would have to be depicted an order of magnitude smaller). [20]

hardly represented in Wikidata, such as songs.[5] As far as Wikidata is concerned, the differences can be partially explained by the external datasets imported into the knowledge graph.

OpenCyc and NELL are generally smaller and less detailed. However, NELL has some particularly large classes, e.g., actor, song, and chemical substance, and for government organizations, it even outnumbers the other graphs. On the other hand, there are classes which are not covered by NELL at all.

5 Overlap of Knowledge Graphs

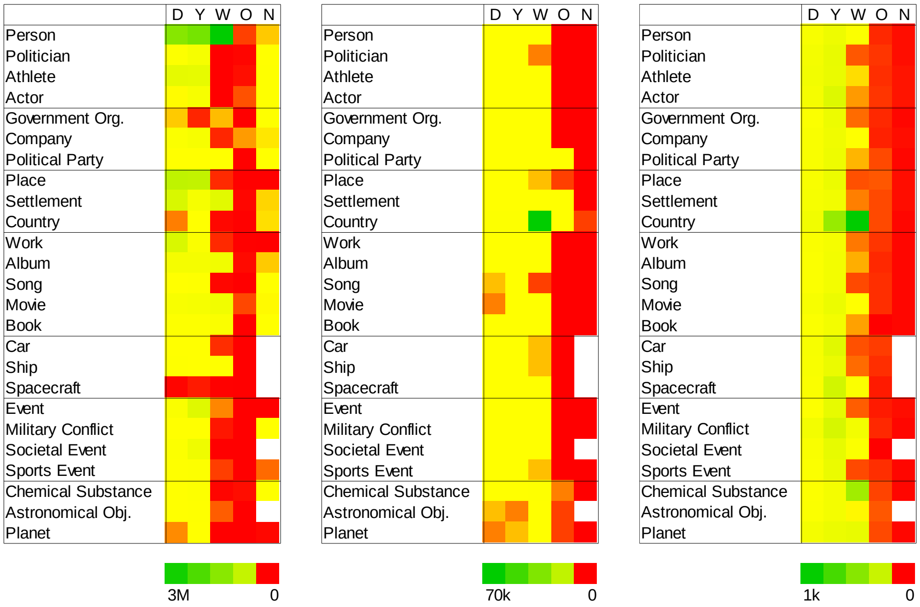
We follow the approach discussed above in section 2.3. For our analysis, we use 16 combinations of string metrics and thresholds on the instances' labels: string equality, scaled Levenshtein (thresholds 0.8, 0.9, and 1.0), Jaccard (0.6, 0.8, and 1.0), Jaro (0.9, 0.95, and 1.0), JaroWinkler (0.9, 0.95, and 1.0), and MongeElkan (0.9, 0.95, and 1.0). Furthermore, to speed up the computation, we exploit token-based blocking in a preprocessing step (where each instance is only assigned to the block of the least frequent token), and discarding blocks larger than 1M pairs.

As incomplete link sets for estimating recall and precision, we use the links between the knowledge graphs, if present. If there are no links, we exploit transitivity and symmetry of `owl:sameAs`, and follow the link path through DBpedia (see Fig. 1). NELL has no direct links to the other graphs, but links to Wikipedia pages corresponding to DBpedia instances, which we use to create links to DBpedia (indicated by the dashed line in the figure).

Fig. 3 depicts the pairwise overlap of the knowledge graphs, using the 25 classes also inspected above, according to two measures: potential gain by joining the two knowledge graphs (i.e., the relation of the union to the larger of the two graphs), and the overlap relative to the existing KG interlinks.

Overall, we can observe that merging two graphs would usually lead to a 5% increase of coverage of instances, compared to using one KG alone. The largest

[5] As discussed above, the reason why so few politicians, actors, and athletes are listed for Wikidata is that they are usually not modeled using explicit classes.

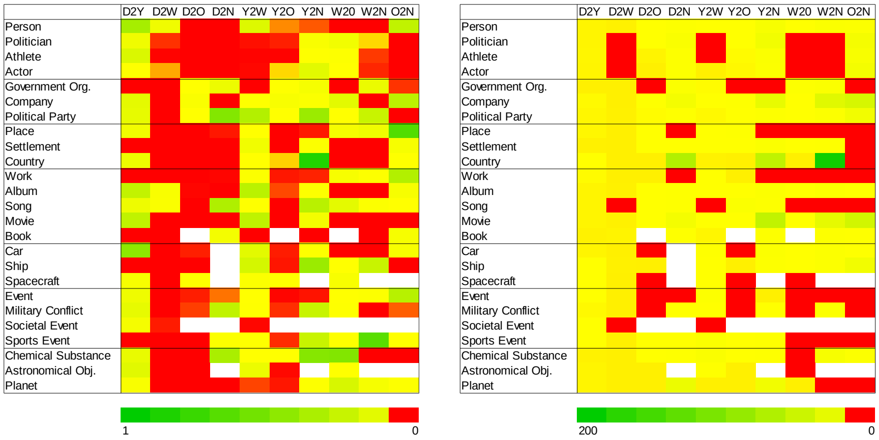


(a) Number of instances (b) Average indegree (c) Average outdegree

Fig. 2: Number of instances (a), avg. indegree (b) and avg. outdegree (c) of selected classes. D=DBpedia, Y=YAGO, W=Wikidata, O=OpenCyc, N=NELL. [20]

potential gain most often comes from merging the larger knowledge graphs with NELL. We can therefore conclude that NELL is rather complementary to most of the other KGs under consideration. The most complementary classes, with an average gain of more than 10% across all pairs of knowledge graphs, are political parties and chemical substances. When looking at the overlap relative to the number of existing links, NELL has the weakest degree of interlinking: e.g., for YAGO and NELL, the estimated overlap is more than eight times larger than the number of interlinks. The classes with the weakest degree of interlinking are countries (32 times larger overlap than explicit interlinks), movies (13 times larger), and companies (10 times larger).[6]

[6] Note that it is not necessary that the linking approach is particularly good, as long as we can estimate its quality reasonably well. In our experiments, the agreement about the estimated overlap is rather high, showing an intra-class correlation coefficient (ICC) of 0.969. In contrast, the size of the actual alignments found by the different approaches differs a lot more, showing an ICC of only 0.646.



(a) Overlap as potential gain

(b) Overlap relative to existing links

Fig. 3: Number as potential gain (a) and relative to existing interlinks (b) of selected classes. D=DBpedia, Y=YAGO, W=Wikidata, O=OpenCyc, N=NELL. [20]

6 Summary of the Comparison of DBpedia, YAGO, Wikidata & co.

We have compared the coverage, level of detail, and overlap for 25 popular classes. Some key findings from this comparison include:

- For person data, Wikidata is the most suitable source, containing twice as many instances as DBpedia or YAGO, at a similar level of detail.
- Organizations, such as companies, are best described in YAGO.
- DBpedia contains more places than the other KGs, including almost four times more cities, villages etc. than YAGO.
- While DBpedia and YAGO contain much more countries than Wikidata (due to the inclusion of historic countries, such as the Roman Empire), Wikidata holds the most detailed information about countries.
- Overall, DBpedia contains the largest number of artistic works, although details differ for subclasses: Wikidata contains more music albums and movies, while YAGO contains more songs. The most detailed information about artistic works is provided by YAGO.
- Cars and spacecraft are best covered in YAGO, while DBpedia is the better resource for ships.
- For events, YAGO is the most suitable source, both in terms of coverage and level of detail.
- NELL contains the largest number of chemical substances. The highest level of degree for chemicals, however, is provided in Wikidata.
- YAGO contains the largest number of astronomical objects.

Note that those numbers are not exhaustive, they merely demonstrate the need for a careful analysis of KGs before exploiting them for a project at hand.

In addition to the question which knowledge graph serves a certain task best, another question is whether it makes sense to use *more than one* combined. Here, we have observed that there is often a considerable complementarity. Especially NELL is very complementary to the other KGs, although a lot less rich in details. Thus, the coverage can often be extended significantly by combining different KGs. This, however, requires refinement of the interlinking, since the interlinks are usually incomplete.

When combining multiple knowledge graphs, we observe that, although a lot of interlinks have been established between the public KGs, the estimated overlap is often much higher. In some cases, the estimated overlap exceeds the number of explicitly set links by a factor of more than 20. Hence, for combining KGs, improving the interlinking has to be a key step.

Depending on the task at hand, other aspects may be important as well. Reliability and correctness of the data in different KGs may be crucial for some tasks, for which other studies should be consulted as well, e.g., [9]. Furthermore, timeliness of the data, as discussed above, may be more important for some tasks than for others.

7 New Developments of Knowledge Graphs

From the observations above, we can see that DBpedia, YAGO, and Wikidata have a similar coverage, while OpenCyc and NELL are much smaller in their coverage, and less detailed. Hence, alternatives to the "big three" knowledge graphs are rare. However, for many applications, having detailed information also about long tail instances would be desirable. Examples include, but are not limited to

Recommender systems that also work well on less well-known artists and/or works, [22]

Named entity recognition and linking systems that also recognize long-tail entities, [8]

Data mining applications backed by knowledge graphs [21, 23] that work on domains and/or entities not well covered in DBpedia and others.

Hence, new developments of knowledge graphs should focus on different sets of entities than those which are already well described in the existing ones. In the following, we will briefly discuss two new developments, i.e., *DBkWik* [13] and *WebIsALOD* [12].

7.1 DBkWik

The reason for the strong similarity of the big public knowledge graphs, i.e., DBpedia, YAGO, and Wikidata, is that they are either extracted from or strongly

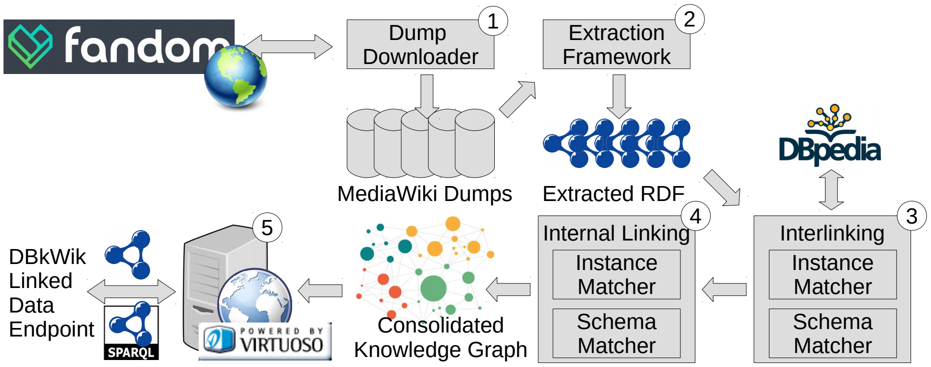


Fig. 4: The DBkWik framework for extracting a knowledge graph from a Wikifarm [13]

oriented at Wikipedia. Hence, their coverage is very close to that of Wikipedia – and to that of each other.

At the same time, there are thousands of Wikis on the Web. *Fandom powered by Wikia*[7] is one of the most popular *Wiki Farms*[8], containing more than 385,000 individual Wikis comprising more than 350 million articles. WikiApiary reports more than 20,000 public installations of the MediaWiki framework, which also underlies Wikipedia[9].

Since those Wikis are technically very similar to Wikipedia, the same tool stack which is used to create a knowledge graph like DBpedia can also be applied to extract a knowledge graph from any other Wiki as well. With *DBkWik*, we have shown that the extraction of a joint knowledge graph from many Wikis is technically feasible. Fig. 4 shows the process.

While for DBpedia, mappings from infobox definitions in Wikipedia to a common ontology are collected in a crowd-sourced process, for DBkWik, neither a common ontology nor such mappings exist. In contrast, the ontologies (for each Wiki) need to be created on the fly in DBkWik.

To create a unified knowledge graphs from those individual graphs, we have to reconcile both the instances (i.e., perform instance matching) as well as the schemas (i.e., perform schema matching). Since pairwise matching of the individual graphs would not be feasible due to its quadratic complexity, we follow a two-step approach: the extracted Wikis are first linked to DBpedia (which is linear in the number of Wikis). The links to DBpedia are then used as *blocking keys* [7] for matching the graphs among each other to reduce the complexity.

As a proof of concept, we have, so far, extracted data from 248 Wikis from Wiki dumps from the Fandom Wiki farm, using the DBpedia Extraction Frame-

[7] `http://fandom.wikia.com/`

[8] `http://www.alexa.com/topsites/category/Computers/Software/Groupware/Wiki/Wiki_Farms`

[9] `https://wikiapiary.com/wiki/Statistics`

work.[10] The resulting dataset comprises 4,375,142 instances, 7,022 classes, and 43,428 (likely including duplicates). Out of those, 748,294 instances, 973 classes, and 19,635 properties are mapped to DBpedia. To match the knowledge graph to DBpedia, we use string matching on labels using surface forms [3] for entities, manually filtering out non-entity pages like list pages, and simple string matching for classes and properties. The resulting knowledge graph encompasses a total of 26,694,082 RDF triples.[11]

7.2 WebIsALOD

While Wikis are fairly easy to process, mainly since the tool stacks for creating Wikipedia-based knowledge graphs already exist, the ultimate goal of knowledge graph creation would be to create a knowledge graph from the entire Web. In [25], we have focused on a particular generic relation for information extraction, i.e., the *hypernymy* relation. That relation holds both between classes (e.g., *industrial metal band* is a hypernym of *band*), as well as for instance-class relations (e.g., *industrial metal band* is a hypernym of *Nine Inch Nails*).

The approach sketched in [25] uses Hearst like patterns to identify hypernymy relations. For example, the pattern *X, such as Y* can be used to infer a hypernymy relation between X and Y (e.g., in the sentence fragment *Industrial metal bands, such as Nine Inch Nails*. The original approach uses more than 50 such patterns to extract hypernymy relations from the *Common Crawl*[12], a large-scale open crawl from the Web. The result of this extraction is the IsADB, a database of 400 million hypernymy relations.

In [12], we have provided the resulting dataset as a Linked Data knowledge graph, enriched with rich provenance metadata, confidence scores computed using a machine learning approach, and interlinks to DBpedia and YAGO. The final resulting dataset consists of the original 400M hypernymy relations, together with a confidence score and metadata, as well as 2,593,181 instance links to DBpedia and 23,771 class links to YAGO. All in all, the dataset consists of 5.4B triples.

In order to obtain a first content profile, we analyzed the fraction of instances which are linked to and typed in DBpedia, and analyzed the type hierarchy in DBpedia to estimate the distribution of those entities. That resulting distribution is depicted in Fig. 5. We can observe that about half of the information is about persons and organizations. Places, works, and species make up for 18%, 12%, and 5%, respectively, while the rest is a mix of other types.

There are various challenges for the WebIsALOD dataset. Examples of ongoing and future work include the learning of better scoring models and the induction of a type hierarchy, where the latter also includes the subtask of automatically distinguishing *subclass of* and *instance of* relations. Further, we aim at extracting relations from pre- and post modifiers of the terms. For example,

[10] `https://github.com/dbpedia/extraction-framework`

[11] `http://dbkwik.webdatacommons.org`

[12] `https://commoncrawl.org`

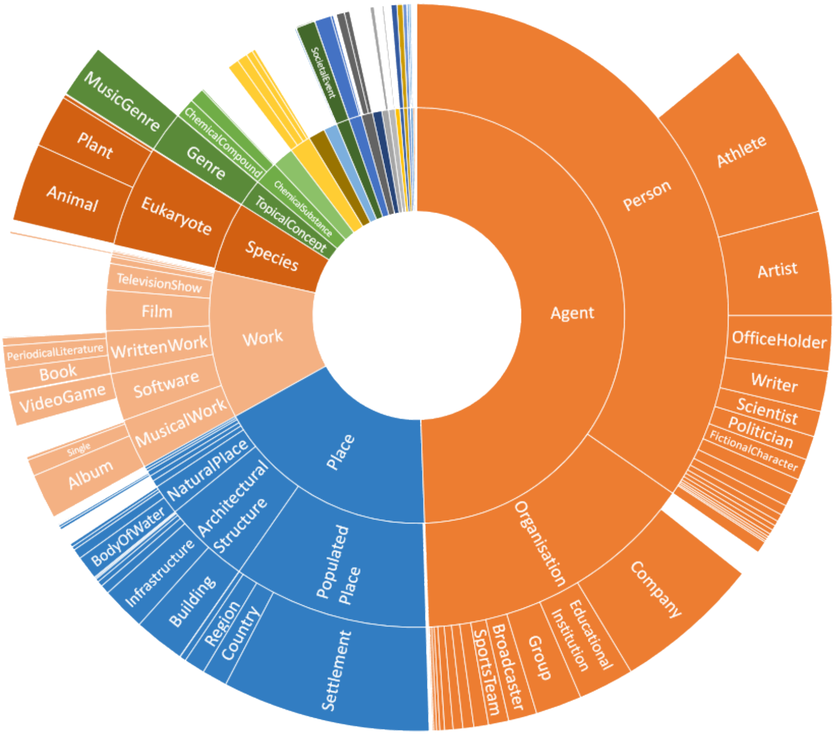


Fig. 5: Type breakdown of the instances in the WebIsALOD dataset [12]

in the hypernymy relation between *Industrial metal band* and *Nine Inch Nails*, *Industrial metal* is a pre-modifier for the head noun *Band*. Hence, we could infer two additional axioms here: in general, the head noun is a hypernym of the compound, i.e., *Band* is a hypernym for *Industrial metal band*. Second, using the information that *Industrial metal* is also a genre, we can heuristically create the axiom that *Industrial metal* is the *genre* of *Nine Inch Nails*, similar to the approach sketched in [10].

Another crucial issue is the identification of homonyms in the dataset. Given the two assertions *Bauhaus is a goth band* and *Bauhaus is a German school*, it is clear that the subjects are two disjoint instances, while *Bauhaus is a goth band* and *Bauhaus is a post-punk band* are not. Identifying such homonyms is an ongoing effort. Here, we will rely both on clustering related hypernyms, as well as linking the type hierarchy to upper ontologies, like it is done for DBpedia [18].

8 Conclusion

In this paper, we have given an in-depth look at knowledge graphs on the Semantic Web. We have seen that, although they are often conceived as comparable, there are measurable differences between DBpedia, YAGO, and Wikidata. Furthermore, we have shown how to estimate the actual overlap between knowledge graphs.

Despite their commonalities, one characteristic shared by the big knowledge graphs is their focus on head entities. We have introduced two prototypes of works in progress – i.e., *DBkWik* and *WebIsALOD* – which also encompass long tail entities. Although still in their infancy, those new knowledge graphs can grow to become a strong complement for the established ones.

Acknowledgements

The author would like to thank Daniel Ringler for his contributions to measuring knowledge graphs, Alexandra Hofmann, Jan Portisch, and Samresh Perchani for their work on DBkWik, Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, and Simone Ponzetto for their contributions to the original WebIsA database, and Sven Hertling for his works on DBkWik and WebIsALOD.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International journal on semantic web and information systems 5(3), 1–22 (2009), `http://dx.doi.org/10.4018/jswis.2009081901`
2. Blanco, R., Cambazoglu, B.B., Mika, P., Torzec, N.: Entity Recommendations in Web Search. In: The Semantic Web–ISWC 2013. LNCS, vol. 8219, pp. 33–48 (2013)
3. Bryl, V., Bizer, C., Paulheim, H.: Gathering alternative surface forms for dbpedia entities. In: Workshop on NLP&DBpedia. pp. 13–24 (2015)
4. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr, E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 101–110 (2010)
5. Dong, X.L., Murphy, K., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Strohmann, T., Sun, S., Zhang, W.: Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion. In: 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 601–610 (2014)
6. Ellefi, M.B., Bellahsene, Z., Breslin, J., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: Rdf dataset profiling - a survey of features, methods, vocabularies and applications. Semantic Web (2017)
7. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering 19(1), 1–16 (2007)
8. van Erp, M., Mendes, P.N., Paulheim, H., Ilievski, F., Plu, J., Rizzo, G., Waitelonis, J.: Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job. In: LREC. vol. 5, p. 2016 (2016)
9. Färber, M., Ell, B., Menne, C., Rettinger, A., Bartscherer, F.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. Semantic Web (to appear) (2016)
10. Heist, N., Paulheim, H.: Language-agnostic relation extraction from wikipedia abstracts. In: International Semantic Web Conference (2017)

11. Hellmann, S., Stadler, C., Lehmann, J., Auer, S.: Dbpedia live extraction. On the Move to Meaningful Internet Systems: OTM 2009 pp. 1209–1223 (2009)
12. Hertling, S., Paulheim, H.: Webisalod: Providing hypernymy relations extracted from the web as linked open data. In: International Semantic Web Conference (2017)
13. Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: Dbkwik: Towards knowledge graph creation from thousands of wikis. In: International Semantic Web Conference (Posters and Demos) (2017)
14. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web Journal 6(2) (2013)
15. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
16. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web 8(3), 489–508 (2017)
17. Paulheim, H., Bizer, C.: Type Inference on Noisy RDF Data. In: The Semantic Web–ISWC 2013, LNCS, vol. 8218, pp. 510–525. Springer, Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-41335-3_32
18. Paulheim, H., Gangemi, A.: Serving dbpedia with dolce–more than just adding a cherry on top. In: International Semantic Web Conference. pp. 180–196. Springer (2015)
19. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From freebase to wikidata: The great migration. In: Proceedings of the 25th International Conference on World Wide Web. pp. 1419–1428 (2016)
20. Ringler, D., Paulheim, H.: One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In: 40th German Conference on Artificial Intelligence (2017)
21. Ristoski, P., Bizer, C., Paulheim, H.: Mining the web of linked data with rapidminer. Web Semantics: Science, Services and Agents on the World Wide Web 35, 142–151 (2015)
22. Ristoski, P., Mencía, E.L., Paulheim, H.: A hybrid multi-strategy recommender system using linked open data. In: Semantic Web Evaluation Challenge. pp. 150–156. Springer (2014)
23. Ristoski, P., Paulheim, H.: Semantic web in data mining and knowledge discovery: A comprehensive survey. Web semantics: science, services and agents on the World Wide Web 36, 1–22 (2016)
24. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the Linked Data Best Practices in Different Topical Domains. In: International Semantic Web Conference. LNCS, vol. 8796 (2014)
25. Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.: A large database of hypernymy relations extracted from the web. In: Language Resources and Evaluation Conference, Portoroz, Slovenia (2016)
26. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: 16th international conference on World Wide Web. pp. 697–706 (2007)
27. Vrandečić, D., Krötzsch, M.: Wikidata: a Free Collaborative Knowledge Base. Communications of the ACM 57(10), 78–85 (2014)