DBkWik: Towards Knowledge Graph Creation from Thousands of Wikis

Alexandra Hofmann, Samresh Perchani, Jan Portisch,
Sven Hertling, and Heiko Paulheim

Data and Web Science Group, University of Mannheim, Germany
{ahofmann,sperchan,jportisc}@mail.uni-mannheim.de,
{sven,heiko}@informatik.uni-mannheim.de

Abstract. Popular public knowledge graphs like DBpedia or YAGO are created from Wikipedia as a source, and thus limited to the information contained therein. At the same time, Wikifarms like Fandom contain Wikis for specific topics, which are often complementary to the information contained in Wikipedia. In this paper, we show how the DBpedia approach can be transferred to Fandom to create *DBkWik*, a complementary knowledge graph.

Keywords: Knowledge Graph Creation, Information Extraction, Linked Open Data

1 Motivation

Large-scale open knowledge graph, like DBpedia [3], YAGO [10], and Wikidata [11], play a central role in Linked Open Data as linkage hubs as well as sources of general knowledge [8]. At the same time, the big popular graphs contain very similar information [6]. In particular, while head entities are covered well, long-tail entities are either not contained at all, or described only at a very low level of detail.

While DBpedia and YAGO are created from Wikipedia as a central source of knowledge, there are so-called *Wiki Farms* that host individual Wikis covering special topics, and, in those, also long-tail entities. Among those, *Fandom powered by Wikia*[1] is one of the most popular *Wiki Farms*[2], containing more than 385,000 individual Wikis comprising more than 350 million articles.

In this paper, we introduce *DBkWik*[3], a knowledge graph extracted from Wikia. It applies the DBpedia Extraction Framework to dumps of individual Wikis downloaded from Wikia. We discuss design decisions and challenges, as well as preliminary results.

[1] `http://fandom.wikia.com/`
[2] `http://www.alexa.com/topsites/category/Computers/Software/Groupware/Wiki/Wiki_Farms`
[3] Pronounced *Dee-Bee-Quick*

2 Approach

The DBpedia Extraction Framework takes WikiMedia dumps as input. Hence, the first step of our approach is to collect dumps from the Wikia Web site, as depicted in Fig. 1. However, not all Wikis do have dumps. To date, we have been able to download data dumps of 15,003 Wikis, totaling to 52.4 GB of data (which roughly corresponds to the data dump size of the recent English Wikipedia). The vast majority of those declares English (73.9%) as its language.

As a next step, we execute the DBpedia extraction framework on the extracted dumps. The result is a collection of individual extracted, disconnected knowledge graphs. Since there are no crowd generated mappings to a common ontology, as for the DBpedia ontology, a very shallow schema is generated on-the-fly for each graph. To that end, we declare a class for each infobox type and a property for each infobox key used in the Wiki.

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* [2] for matching the graphs among each other to reduce the complexity.

The resulting knowledge graph is then loaded into a Virtuoso server serving the DBkWik dataset, both as a Linked Data service as well as a SPARQL endpoint.

3 Preliminary Results

As a proof of concept, we have extracted data from 248 Wikis. 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 [1] 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.[4]

For evaluating the matching to DBpedia, we created gold standards manually, selecting eight Wikis randomly, and more than 50 entities from each, totaling in 409 entities annotated with their corresponding DBpedia entity (out of the 409 entities, 20.3% have a corresponding DBpedia entity). We achieve a micro average F1 score of 0.574, as depicted in Table 1.

Likewise, we created a gold standard for mapping classes and properties, using the same set of eight Wikis. The gold standard comprises 27 classes and 161 properties with their DBpedia equivalents. 83.2% of all classes and 44.4% of

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

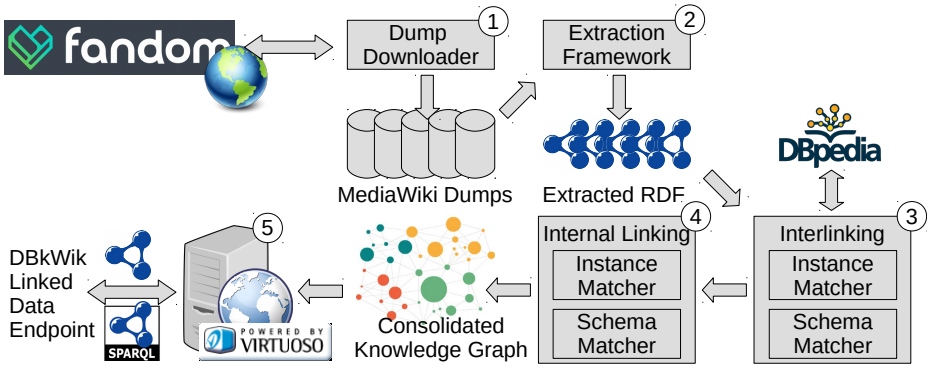

Fig. 1. Schematic depiction of the DBkWik extraction process

all properties have a counterpart in the DBpedia ontology. We achieve a micro average F1 score of 0.917 for classes, and 0.852 for properties. The fourth step, i.e., interlinking the individual graphs using the DBpedia links as blocking keys, has not yet been implemented.

4 Challenges and Future Work

While the current prototype shows a first proof of concept for extracting a knowledge graph from a multitude of individual Wikis, there are still quite a few challenges to be addressed. Those concern the reconciliation of the individual graphs into a consolidated knowledge graph, as well as the interlinking to external datasets.

At the same time, there are a few advantages of the dataset at hand. Since it is extracted from Wikis, both the extracted graph as well as the original Wiki can be exploited for solving those tasks. Hence, both graph-based and text-based techniques can be used and combined.

The main difference to Wikipedia-based knowledge graphs like DBpedia and YAGO is that there are no manual mappings to a central ontology. Hence, the ontology has to be created on the fly. In the current version, classes and properties form only a shallow schema. To make the knowlege graph more valuable, we need to unify them into a common ontology. Here, ontology learning techniques [4] may be applied. Since, as discussed above, both text and graph are available, techniques exploiting both [7] are very promising. Furthermore, it will be interesting to see how links to the DBpedia ontology and the rich axioms therein may be utilized for creating and refining the unified ontology.

Likewise, the interlinking results to DBpedia need improvement, mainly on instance level and in terms of precision. Here, dual approaches utilizing both the graph and the original text representation are considered most promising at the moment. Furthermore, approaches that try to solve the instance and schema

Table 1. Performance of interlinking to DBpedia

	Macro avg.			Micro avg.		
	P	R	F1	P	R	F1
Instances	.449	.976	.574	.482	.957	.641
Classes	1.000	.875	.917	1.000	.917	.957
Properties	.804	.924	.852	.798	.917	.853

matching as a unified problem – where instance equivalences can serve as clues for schema equivalences and vice versa [9] – are worth considering.

Although we have targeted one Wiki hosting platform for this prototype, the creation of the knowledge graph does not need to end there. WikiApiary reports more than 20,000 public installations of MediaWiki[5], all of which could be digested by the framework introduced in this paper.

Last, but not least, many methods for refining knowledge graphs have been proposed in the recent past, e.g., for completing missing information and/or finding errors [5]. Incorporating those into the extraction – with a careful eye on approaches which are scalable – would make the resulting knowledge graph even more valuable.

References

1. Bryl, V., Bizer, C., Paulheim, H.: Gathering alternative surface forms for dbpedia entities. In: Workshop on NLP&DBpedia. pp. 13–24 (2015)
2. 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)
3. Lehmann, J., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web 6(2), 167–195 (2015)
4. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent systems 16(2), 72–79 (2001)
5. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web 8(3), 489–508 (2017)
6. 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)
7. Ristoski, P., Faralli, S., Ponzetto, S.P., Paulheim, H.: Large-scale taxonomy induction using entity and word embeddings. In: International Conference on Web Intelligence (WI) (2017), to appear
8. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: ISWC. pp. 245–260. Springer (2014)
9. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: Probabilistic alignment of relations, instances, and schema. VLDB Endowment 5(3), 157–168 (2011)
10. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
11. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM 57(10), 78–85 (2014)

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