SectionLinks: Mapping Orphan Wikidata Entities onto Wikipedia Sections

Natalia Ostapuk[1], Djellel Difallah[2], and Philippe Cudré-Mauroux[1]

[1] University of Fribourg, Fribourg, Switzerland {firstname.lastname}@unifr.ch
[2] New York University, New York, USA djellel@nyu.edu

Resource Type: Dataset
Permanent URL: http://doi.org/10.5281/zenodo.3840622

Abstract. Wikidata is a key resource for the provisioning of structured data on several Wikimedia projects, including Wikipedia. By design, all Wikipedia articles are linked to Wikidata entities; such mappings represent a substantial source of both semantic and structural information. However, only a small subgraph of Wikidata is mapped in that way – – only about 10% of the sitelinks are linked to English Wikipedia, for example. In this paper, we describe a resource we have built and published to extend this subgraph and add more links between Wikidata and Wikipedia. We start from the assumption that a number of Wikidata entities can be mapped onto Wikipedia sections, in addition to Wikipedia articles. The resource we put forward contains tens of thousands of such mappings, hence considerably enriching the highly structured Wikidata graph with encyclopedic knowledge from Wikipedia.

Keywords: Wikidata · Wikipedia · Linked Data.

1 Introduction

Knowledge Graphs (KGs) provide a rich, structured, and multilingual source of information useful for a variety of applications that require machine-readable data. KGs are leveraged in search engines, natural language understanding, and virtual assistants, to name but a few examples. A KG is usually represented as a graph of vertices denoting entities and connected with directed edges depicting their relationships. KGs can be constructed automatically using information extraction techniques, or semi-automatically, as is the case with Wikidata[3], a KG built and maintained by a community of volunteers. Wikidata has the advantage of being curated by humans and of being tightly integrated with multiple Wikimedia projects (e.g., Wikipedia, Wikimedia Commons, and Wiktionary). For example, every Wikipedia article

[3] https://www.wikidata.org/

across all languages has a corresponding and unique language-independant Wikidata entity. This mapping between Wikipedia and Wikidata is beneficial for both projects. On one hand, it facilitates information extraction and standardization of Wikipedia articles across languages, which can benefit from the standard structure and values of their Wikidata counterpart, e.g., for populating infoboxes. On the other hand, Wikipedia articles are routinely updated, which in turn keeps Wikidata fresh and useful for online applications.

However, the Wikipedia editorial guidelines require that an entity be notable or worthy of notice to be added to the encyclopedia, which is not the case of Wikidata. Hence, only a fraction of Wikidata entities has a corresponding article in any language. We refer to the remaining entities, without an article, as *orphans*. In the absence of a textual counterpart, orphans often suffer from incompleteness and lack of maintenance.

Our present work stems from the observation that a substantial number of orphan entities are indeed available in Wikipedia, but not at the page level; orphan entities can be described within existing Wikipedia articles in the form of sections, subsections, and paragraphs of a more generic concept or fact. Interestingly, even a short section describing an orphan Wikidata entity can carry useful information that could enrich the entity with additional facts and relationships. Such pieces of information are unfortunately buried inside long articles without direct relevance to the main subject. Instead, we propose to establish a fine-grained mapping between Wikidata orphan entities and Wikipedia (sub)-sections.

Our main contribution is a dataset of such mapping between Wikidata and Wikipedia sections that we created using several algorithmic methods, ranging from string matching to graph inference.

2 Related Work

To the best of our knowledge, we are the first to come up with a resource providing fine-grained mappings between Wikipedia and Wikidata; our mappings come in addition to the existing links that Wikipedia provides to Wikidata through section anchors (see Section 3).

A similar effort of matching entities to Wikipedia articles was made by Tonon et al. in [18]. The paper addresses the problem of constructing a knowledge graph of Web Entities and mapping it onto DBpeadia with Wikipedia articles acting as DBpedia entries.

Our effort is not directly related to link prediction [10,12], which typically operates in a homogeneous domain (e.g., when trying to infer new links in a given social network or knowledge graph), while we operate across two heterogeneous domains (i.e., Wikidata and Wikipedia). It is however related to Ad-hoc Object Retrieval techniques [13,17], which retrieve target entities

based on keyword or natural language queries, as well as to Entity Linking [16,4,1,11], which attempts to link mentions in Web text to their referent entities in a knowledge base.

A special case of Entity Linking is Wikipedia Linking, which aims at discovering links between Wikipedia documents. This task was broadly studied within the Wiki track of INEX[4] conference [7,5,6]. Participants were invited to establish links between Wikipedia articles both at the page and text level (i.e. detect an anchor point in the text of the source document and a best entry point in the text of the target). The task of linking documents at the text level is of particular interest to us as it is a general case of linking a document to a section and closely relates to the main topic of this paper. A number of interesting approaches were developed both for identifying link source and target pages [8] and detecting the best entry point inside the text of the target [3].

Our work is also directly related to information extraction [15] and KG construction [14] efforts. In that context, a number of systems have recently been proposed to extract information, often in the form of triples, from structured or unstructured content and link it to a semi-structured representation like a knowledge graph. DeepDive [19], for instance, is a well-known tool that employs statistical learning, inference and declarative constructs written by the user to build a knowledge base from a large collection of text. FRED [2] is a machine-reading tool that automatically generates RDF/OWL ontologies and linked data from multilingual natural language text. MapSDI [9] is a recent rule-based mapping framework for integrating heterogeneous data into knowledge graphs. It takes as input a set of data sources and mapping rules to produce a consolidated knowledge graph. None of those tools is readily applicable to our problem of linking Wikipedia sections to Wikidata entities, however.

3 Relevance and Use Cases

In Wikidata, entities are characterized by a unique identifier (a sequential integer prefixed with Q), multilingual labels, descriptions, and aliases when available. Each entity may have multiple statements to express a property or a relationship with another entity. An entity can have Sitelinks[5] referencing other Wikimedia projects. These are hyperlinks that establish an identity mapping between the entity and, for instance, a Wikipedia page. Thanks to Sitelinks, Wikidata is often utilized as a hub for multilingual data, connecting a given concept to articles written in a dozen languages.

To understand Wikidata's Sitelinks coverage, we collected the number of label entities per language. We focus on the 15 languages having +1 Million

[4] International Workshop of the Initiative for the Evaluation of XML Retrieval
[5] https://www.wikidata.org/wiki/Help:Sitelinks

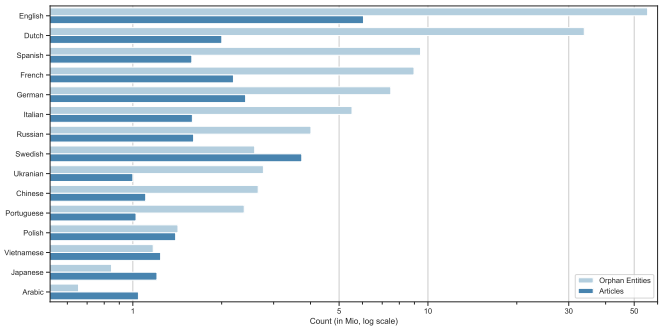

Fig. 1. Language statistics and gaps in Wikidata.

Wikipedia articles (see Section 4). We examined the number of orphan entities (defined above in Section 1) having a label in each language, as shown in Figure 1, which we contrast to the number of available Wikipedia articles. We see that the gap between the number of orphans and articles is much higher for languages having more labels. In fact, English Wikipedia, the largest and most active project of all Wikis, links to only about 10% of all Wikidata entities having an English label. This discrepancy signals a necessity to close this gap using alternate methods.

This work aims to identify potential orphan entity textual content that may exist within Wikipedia in the form of sections. This content can be linked using anchor links to article sections. Currently, Wikidata does not support using anchors links as a Sitelink, i.e. linking to a specific section of a page.

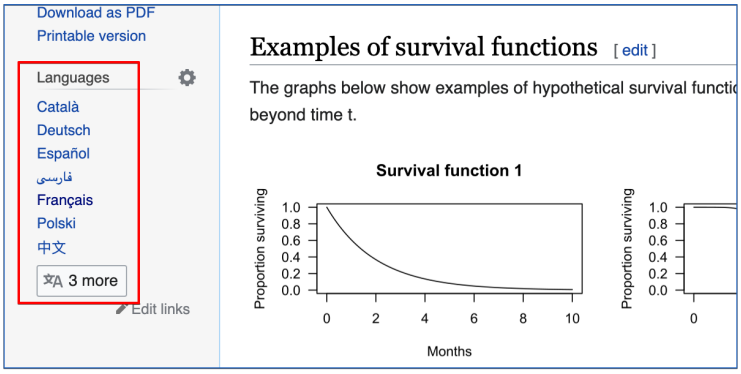

Fig. 2. A list of Wikidata sitelinks on a Wikipedia page. The French link points to: `https://fr.wikipedia.org/wiki/Analys_de_survie#Fonction_de_survie`

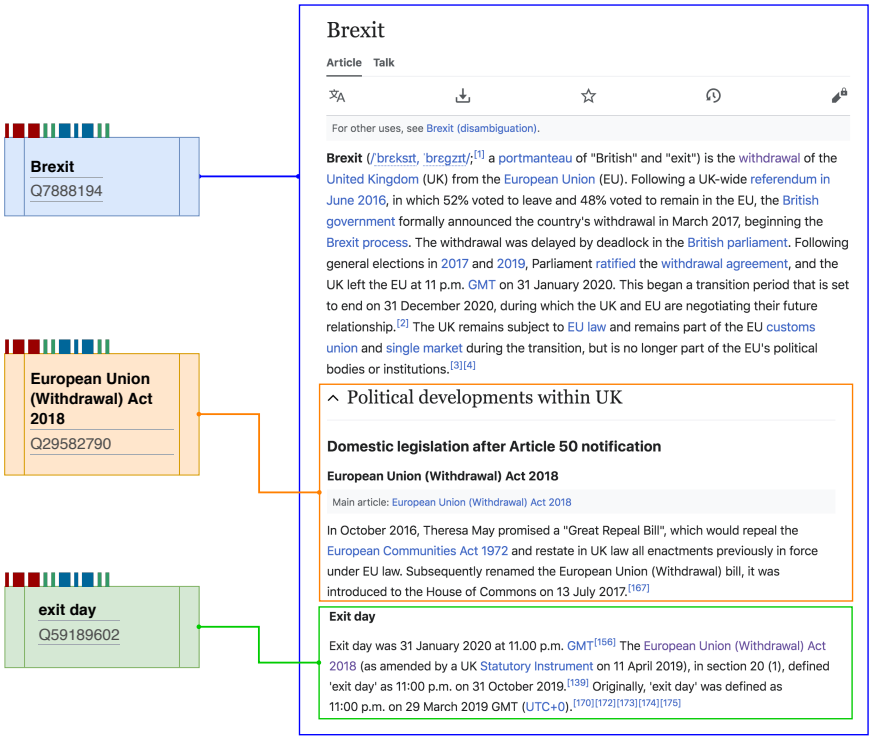

Fig. 3. An example where linking orphan entities from Wikidata (Q29582790 and Q59189602) to their corresponding section in Wikidata has the potential to greatly improve both sources by extracting key data from text.

It is worth noting that inter-language Wikipedia can perform this operation, for example, the Wikidata entity Q2915096 contains a Sitelink to the English Wikipedia page *Survival_function*, and all the other Wikidata sitelinks are listed on this page in the left column (Figure 2). A link to a section can be added to this list and thus can be mapped to the source Wikidata entity, as is the case for the French language. Unfortunately, this is done inconsistently and provides only an indirect mapping to Wikidata, and also assumes that at least one language has a dedicated Wikipedia page for the entity. Our proposed resource fills this important gap by building an external resource to map Wikidata orphans to Wikipedia, without entering a sitelink.

Figure 3 illustrates what we want to achieve. It depicts the Wikipedia entry for *Brexit*. While the *Brexit* entity (Q7888194) from Wikidata correctly links to that page, two related Wikidata entities are orphans: *European Union (Withdrawal) Act 2018* (Q29582790) and *exit day* (Q59189602). Linking those two entities to their corresponding sections in Wikipedia, as

shown in the figure, would provide important information and context to Wikidata and greatly improve a number of key downstream tasks such as ad-hoc object retrieval, joint embeddings, or question answering.

4 The Dataset

We developed two different algorithms to derive mappings from Wikidata entities to Wikipedia sections. We ran both our algorithms on 15 languages and obtained tens of thousands of new links in the process (see section 4.2 for details). The two resulting datasets complement each other (i.e. contain sitelinks for different sets of entities) and are both available as part of our resource. The rest of this section describes our methods and results in detail, and provides performance numbers and illustrative examples to better assess the usefulness of our resource.

4.1 Data Generation Pipeline

We consider a bipartite graph $\mathcal{G}$ whose vertices consist of two disjoint subsets: $\mathcal{D}$, representing Wikidata entities that are missing a Wikipedia link, and $\mathcal{P}$, representing Wikipedia page sections. Our goal is to correctly match as many vertices as possible from $\mathcal{D}$ to $\mathcal{P}$ (i.e., to create as many correct links as possible between Wikidata entities and Wikipedia section). To help with this task, we use existing labels and statements available from each entity, as well as the section titles that we collect from the 15 Wikis. We proceed in four steps:

Candidates selection: the first step is to identify candidates, both from Wikidata ($\mathcal{D}$ vertices) and Wikipedia ($\mathcal{P}$ vertices), in order to create the matching graph $\mathcal{G}$.

Key generation: then, we create a key (or a set of keys) to represent each vertex in $\mathcal{D}$ and $\mathcal{P}$.

Matching: at this stage, we create candidate links between by matching keys in $\mathcal{D}$ with keys in $\mathcal{P}$.

Filtering: finally, as the matching step may result in many false positive links, we consider a postprocesing step where each resulting link is vetted against a set of roles or conditions.

We describe two different instantiations of our data generation pipeline below: one considering a strict *all-to-all* matching algorithm, and a second, graph-based algorithm that takes into account the neighborhood of each candidate.

All-to-All Matching Algorithm Our first approach considers a *complete* bipartite graph, where each Wikidata entity in $\mathcal{D}$ is matched to all Wikipedia sections in $\mathcal{P}$. Since we do not apply any restriction on the candidate targets (Wikipedia pages) and since the number of matches grows quadratically, the key comparison method and the filtering functions both have to be very strict, otherwise the algorithm would return a lot of false positive matches. We achieve this with the requirement that a Wikipedia key has to comprise all tokens from both the page title and the section title; as such, a Wikipedia key is specific enough to guarantee with a high probability that it refers to the same object as a corresponding Wikidata entity.

Candidates selection First, we identify all orphan Wikidata entities, i.e. all entities that have a label in a given language but do not have a sitelink to a corresponding Wikipedia page or section. Orphans are further filtered by type to exclude service pages like categories or templates, as well as some types which have homonymous labels but rarely match any Wikipedia section (for example, an entity of the type *painting* with the label *The Crucifixion* match the Wikipedia section describing the crucifixion of Jesus, which is irrelevant to this object).

Key generation We consider a set of keys for each Wikidata candidate in $\mathcal{D}$. This set of keys consists of its label and all its aliases for a given language. For example, for an entity `Q63854053` the set of keys will be *{"spun silk", "noil", "silk noil"}*. To generate keys for Wikipedia page section in $\mathcal{P}$, we concatenate the page title with the section title. After all keys are generated, we split each key into tokens, remove punctuation and stop-words, sort tokens in alphabetic order and concatenate them back together. We used stop word lists provided by the NLTK package[6] in this context.

Matching The output of the key generation step consist of two key-value tables: one for Wikidata entities, where the keys are as described above and each value is an entity id, and another for Wikipedia sections, with $(page\_title-section\_title)$ pairs as values. These two tables are then joined by key and grouped by QID (Wikidata entity id). This operation was performed on a Hadoop cluster.

Filtering The last step of the pipeline is result filtering. As mentioned above, this approach considers all possible matches and hence may bring up a lot of false positives, therefore the filtering function we use is strict also: we keep only those QIDs for which exactly one Wikipedia section was found. In more formal terms, this step of the algorithm checks the output of the groupBy operation and filters out records which grouped more than one value per QID. Figure 4 outlines the overall pipeline of our first approach.

[6] `https://www.nltk.org`

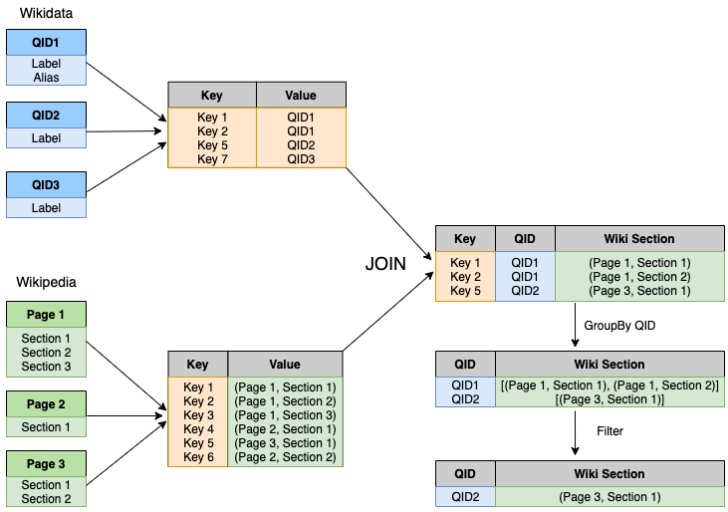

Fig. 4. All-to-all matching pipeline

Neighbors Matching Algorithm Although the above algorithm demonstrates a good performance (over 80% precision for English Wikipedia), a manual analysis reveals that it has a relatively low recall. The reason is that in many cases, when a Wikipedia section describes an object its title is self-sufficient, i.e. it stands on its own and does not depend on the title page to identify the object. Hence, matching a Wikidata label strictly with a combination of a page and section title results in a lot of false negatives as the page title introduces redundancy. On the other hands, matching with section titles only will significantly drop the precision in general. To tackle this problem, we restrict the set of candidate Wikipedia sections for each Wikidata entity by leveraging the Wikidata graph structure.

Candidates selection Our candidate selection algorithm in this case is based on the assumption that a Wikipedia page that is "semantically" related (e.g., to a subclass relation) to a Wikidata entity in $\mathcal{D}$ is more likely to contain a section relevant to that entity.

We introduce a second condition to further restrict the candidates as follows: a candidate in $\mathcal{P}$ should be related to one and only one source entity in $\mathcal{D}$ for a particular edge type. For example, consider an orphan entity *badminton racket* and a triple [(badminton), (uses), (badminton racket)]. Here, *(badminton)* is a good candidate, because it is linked to *(badminton racket)* with the relation *[uses]*. On the other hands, in the triple [(Sofia Shinas), (occupation), (singer)], *(singer)* is *not* an interesting candidate for *(Sofia Shinas)*, as many entities have an occupation *singer*.

As such, we developed the following algorithm pipeline for selecting candidate Wikipedia sections using a graph-based approach:

- Identify an orphan Wikidata entity;
- Collect its neighbor entities following all incoming and outgoing edges;
- Filter out neighbors that do not have a Wikipedia sitelink;
- Filter out neighbors with non-unique edges;
- Extract Wikipedia sitelinks from the remaining neighbors;
- Consider sections of the resulting Wikipedia pages as candidates for matching.

This algorithms yields excellent results in practice as we describe below in Section 4.3.

Key generation As we significantly limited the set of candidate Wikipedia sections in $\mathcal{P}$, we consider a different way of constructing the keys. First, we do not always consider the tokens from the page title for the keys in $\mathcal{P}$ (although they may be included). Second, in addition to removing stop words and punctuation, we consider a third postprocessing step by stemming the key tokens, i.e. we remove all affixes that mostly carry morphological information and only keep just their root (e.g. words *works*, *worked*, *working* are all reduced to *work*). Finally, we remove disambiguation tokens from Wikipedia page titles: when a title is ambiguous, a disambiguation word or phrase can be added in parenthesis. For example, titles *Mercury (element)*, *Mercury (planet)* and *Mercury (mythology)* are all reduced to *Mercury*.

Matching The matching step is similar to the one in our first algorithm, but instead of running a join of two tables we process each Wikidata entity in $\mathcal{D}$ individually. If one of the Wikidata keys exactly match a Wikipedia section key, we consider the section as a potential sitelink for this entity.

Filtering Due to the various manipulations we consider on the keys, we may end up with situations where different Wikipedia sections have the same key that matches a Wikidata key. For a example, an article *Rotterdam Metro* includes two sections: *Line D* and *Lines*. After stemming and stop words removal, both section titles are reduced to *line*. If we consider a Wikidata entity with a label *Line D* (which is also reduced to *line*), we get two potential matches. In that case, we consider the edit distance between the Wikidata label and the section title (in their original forms) and pick the closest match to break the tie.

Table 1. Datasets size per language

	All-to-all matching	**Neighbors matching**	**Final**
Arabic	672	1752	1792
Chinese	38	848	882
Dutch	2699	5230	5644
English	9834	25469	30351
French	6573	10436	13098
German	3923	9140	10443
Italian	8512	27775	32262
Japanese	215	4105	4229
Polish	663	678	1271
Portuguese	399	1188	1407
Russian	2844	3404	4675
Spanish	3081	6754	7939
Swedish	2552	3035	3343
Ukrainian	309	290	571
Vietnamese	8164	108	8244
Total	50478	100212	126151

4.2 Resource Description

We ran our methods using the dumps of 15 Wikipedias obtained from April 2020.[7] While the Wikidata graph dump was obtained from February 2020.[8]

The final resource contains 126,151 sitelinks for 109,734 unique entities for 15 languages obtained with the two methods described above. The subset of languages we initially considered were chosen according to the following criteria:

1. Number of articles in the corresponding Wikipedia (over 1 million)[9]
2. Number of Wikipedia active users (over 1000)

We plan to run algorithms on more languages in the future. We report below the full list of languages we considered as well as detailed statistics on the datasets (see Table 1).

4.3 Evaluation Results

To estimate the precision of our resource, we randomly sampled several hundreds matches from each dataset and manually evaluated them as either true or false. For instance, one algorithm matched the entity *Q49001814*

[7] https://dumps.wikimedia.org/
[8] https://dumps.wikimedia.org/wikidatawiki/
[9] https://meta.wikimedia.org/wiki/List_of_Wikipedias

Table 2. Datasets precision

	All-to-all matching	**Neighbors matching**	**Final**
Arabic	0.99	1.0	0.99
English	0.82	0.81	0.82
French	0.85	0.92	0.89
Russian	0.88	0.85	0.87

(*Timber Dam*, the name of the dam in Montana, USA) to the section *Timber dams* of the Wikipedia page *Dam*, which describes a type of dams made of timber. This match was labelled as a false positive. An example of a true positive match is a mapping of the entity *Q334415* (*security camera*) onto the page *Surveillance*, section *Cameras*. We labelled each sample this way and then divided the number of true positive matches by the sample size to get a precision value. We then generalized from the sample observations to the whole dataset using linear extrapolation in order to estimate the dataset precision. Table 2 reports our results.

We evaluated 12 samples – one sample per algorithm plus the joint results, for 4 different languages (Arabic, English, French, Russian). Each sample contains around 200 mappings. This number was chosen empirically, as we observed that 200 random examples were enough to stabilize the metric, and increasing the sample size did not change the resulting value significantly. Overall we manually labelled 2400 mappings.

Our evaluation aims to demonstrate that the overall accuracy of the resource is high enough that it can be used for many tasks that do not require a perfect dataset (for example, most deep learning algorithms are robust to errors in the training set). We cannot provide evaluation results for all languages unfortunately, as we decided to focus on those languages we felt comfortable with only.

5 Availability and Reusability

Our resource is available in JSON and RDF formats and comply with the Wikibase data model.[10] To keep the resource compact and as easy to process as possible, we only publish the sitelinks discover using our methods.

In the JSON representation, an entity contains two fields: *id* (the unique identifier of an entity) and *sitelinks* (links to Wikipedia pages). Each sitelink record comprises three fields: *site*, *title* and *url*. A section title is appended to the page title separated with # symbol. Such a compound title is then URL-encoded and added to the URL path. Following the Wikidata guidelines, each entity is encoded as a single line.

[10] `https://www.mediawiki.org/wiki/Wikibase/DataModel`

The RDF dump is serialized using the Turtle format and stores nodes describing Wikipedia links. Section titles are added in the same manner as described above . [11]

The resource is published on the Zenodo platform under CC BY 4.0 license.[12]. The canonical citation is available on the Zenodo page. The source code is also available on our github repository to help maintain and generate newer releases in the future.[13].

6 Conclusion and Future Work

We presented a dataset that extends Wikidata orphan entities with Sitelinks referencing Wikipedia sections for the 15 most prominent languages in Wikipedia. To generate this resource, we employed string matching and graph processing methods that leverage multilingual labels and the graph structure to find corresponding sections in Wikipedia. Since our methods use heuristics, we compute the accuracy of a subset of the data using manual judgment. This piece of information can be useful to inform downstream application on how to use the data. For instance, for entities with an English label, we identified 9,834 links with 82% accuracy when using exact label matching, and 25,469 links when using graph-based method alone with 81% accuracy.

We believe that using this resource can improve both sources in terms of completeness and freshness, as well as diminish the information gap that persists between Wikipedia-based entities and tail-entities. For example, one could build targeted information extraction tools and automatically curate entities that do not have a dedicated Wikipedia article using our resource. As future work, we plan to incorporate embedding-based similarity scores into our mapping method and perform a comprehensive evaluation of the obtained results in terms of both precision and recall. We also envision building a section recommendation system that can be offered to Wikidata editors for relevance judgment.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 683253/GraphInt).

[11] For the detailed description of the Wikidata RDF format refer to: `https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format`

[12] `http://doi.org/10.5281/zenodo.3840622`

[13] `https://github.com/eXascaleInfolab/WikidataSectionLinks`

References

1. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st International Conference on World Wide Web. p. 469–478. WWW '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2187836.2187900, https://doi.org/10.1145/2187836.2187900
2. Gangemi, A., Presutti, V., Recupero, D.R., Nuzzolese, A.G., Draicchio, F., Mongiovì, M.: Semantic web machine reading with FRED. Semantic Web **8**(6), 873–893 (2017). https://doi.org/10.3233/SW-160240, https://doi.org/10.3233/SW-160240
3. Geva, S., Trotman, A., Tang, L.X.: Link discovery in the wikipedia. Shlomo Geva, Jaap Kamps, Andrew Trotman p. 326 (2009)
4. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: A graph-based method. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 765–774. SIGIR '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2009916.2010019, https://doi.org/10.1145/2009916.2010019
5. Huang, D.W.C., Geva, S., Trotman, A.: Overview of the INEX 2008 link the wiki track. In: Geva, S., Kamps, J., Trotman, A. (eds.) Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers. Lecture Notes in Computer Science, vol. 5631, pp. 314–325. Springer (2008). https://doi.org/10.1007/978-3-642-03761-0_32, https://doi.org/10.1007/978-3-642-03761-0_32
6. Huang, D.W.C., Geva, S., Trotman, A.: Overview of the INEX 2009 link the wiki track. In: Geva, S., Kamps, J., Trotman, A. (eds.) Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers. Lecture Notes in Computer Science, vol. 6203, pp. 312–323. Springer (2009). https://doi.org/10.1007/978-3-642-14556-8_31, https://doi.org/10.1007/978-3-642-14556-8_31
7. Huang, D.W.C., Xu, Y., Trotman, A., Geva, S.: Overview of INEX 2007 link the wiki track. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers. Lecture Notes in Computer Science, vol. 4862, pp. 373–387. Springer (2007). https://doi.org/10.1007/978-3-540-85902-4_32, https://doi.org/10.1007/978-3-540-85902-4_32
8. Itakura, K.Y., Clarke, C.L.A.: University of waterloo at INEX2007: adhoc and link-the-wiki tracks. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers. Lecture Notes in Computer Science, vol. 4862, pp. 417–425. Springer (2007). https://doi.org/10.1007/978-3-540-85902-4_35, https://doi.org/10.1007/978-3-540-85902-4_35

9. Jozashoori, S., Vidal, M.: Mapsdi: A scaled-up semantic data integration framework for knowledge graph creation. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) On the Move to Meaningful Internet Systems: OTM 2019 Conferences - Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, October 21-25, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11877, pp. 58–75. Springer (2019). https://doi.org/10.1007/978-3-030-33246-4_4, https://doi.org/10.1007/978-3-030-33246-4_4
10. Liben-Nowell, D., Kleinberg, J.M.: The link prediction problem for social networks. In: Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003. pp. 556–559. ACM (2003). https://doi.org/10.1145/956863.956972, https://doi.org/10.1145/956863.956972
11. Lin, T., Mausam, Etzioni, O.: Entity linking at web scale. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction. p. 84–88. AKBC-WEKEX '12, Association for Computational Linguistics, USA (2012)
12. Martínez, V., Berzal, F., Talavera, J.C.C.: A survey of link prediction in complex networks. ACM Comput. Surv. **49**(4), 69:1–69:33 (2017). https://doi.org/10.1145/3012704, https://doi.org/10.1145/3012704
13. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th International Conference on World Wide Web. pp. 771–780. WWW '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1772690.1772769, http://doi.acm.org/10.1145/1772690.1772769
14. Qiao, L., Yang, L., Hong, D., Yao, L., Zhiguang, Q.: Knowledge graph construction techniques. Journal of Computer Research and Development **53**(3), 582–600 (2016)
15. Sarawagi, S.: Information extraction. Foundations and trends in databases **1**(3), 261–377 (2008)
16. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2015). https://doi.org/10.1109/TKDE.2014.2327028, https://doi.org/10.1109/TKDE.2014.2327028
17. Tonon, A., Demartini, G., Cudré-Mauroux, P.: Combining inverted indices and structured search for ad-hoc object retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 125–134. SIGIR '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2348283.2348304, http://doi.acm.org/10.1145/2348283.2348304
18. Tonon, A., Felder, V., Difallah, D.E., Cudré-Mauroux, P.: VoldemortKG: Mapping schema.org and Web Entities to Linked Open Data. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46547-0_23, https://exascale.info/assets/pdf/voldemort.pdf, http://www.slideshare.net/eXascaleInfolab/voldemortkg-mapping-schemaorg-and-web-entities-to-linked-open-data

19. Zhang, C., Ré, C., Cafarella, M.J., Shin, J., Wang, F., Wu, S.: Deepdive: declarative knowledge base construction. Commun. ACM **60**(5), 93–102 (2017). https://doi.org/10.1145/3060586, https://doi.org/10.1145/3060586