Knowledge gap discovery: A case study of Wikidata

Millenio Ramadizsa[1], Fariz Darari[1,2], Werner Nutt[3] and Simon Razniewski[4]

[1]Faculty of Computer Science, Universitas Indonesia, Indonesia
[2]Tokopedia-UI AI Center of Excellence, Indonesia
[3]Free University of Bozen-Bolzano, Italy
[4]Bosch Center for AI, Germany

Abstract

Society, science, and economy are becoming more and more data-driven, and therefore the study of gaps in knowledge gains importance. The arguably most prominent public source of structured knowledge is Wikidata, which contains impressive amounts of knowledge, but nonetheless comes with surprising gaps.

In this paper we propose a framework for identifying class-level knowledge gaps in Wikidata, based on the concepts of *gap properties*, i.e., properties that mostly exist for prominent entities, but are missing in the tail, and the *gap property ratio.* We conduct an analysis for a varied set of 20 classes, and show that our framework can discover unexpected knowledge gaps, that may guide contributors towards addressing them.

1. Introduction

Society, science, and economy are getting increasingly reliant on data in decision making. While the overall trend is that data and knowledge are vastly and constantly collected, stored, and processed, this does not happen at an equal rate: domains, topics, or subjects receive uneven coverage. These imbalances have fueled a whole research field that uncovers them, most notably in terms of coverage of genders [1, 2, 3], citizenships [4], and individual entity assertions [5].

Whether observed imbalances really reflect a bias of editors or imbalances of the real-world, is often difficult to disentangle [6], and one should therefore be cautious with drawing conclusions. Nonetheless, awareness of knowledge gaps, and their characterization, is a first step towards investigating possible root causes, and is therefore a crucial task.

In this paper, we propose to identify and characterize knowledge gaps of classes in knowledge graphs via the concept of *gap properties*, which are properties, that are frequently present among the "information-richest" entities in a class, but largely absent from the "poor" ones. Gap properties can then be used to identify classes with large gaps (that is, classes, where most properties are gap properties), and (ii) to characterize which properties constitute the imbalances.

Our contributions are three-fold:

1. We introduce the concept of *gap properties* for knowledge graphs, and show how they can be used to identify and characterize class-level knowledge gaps in knowledge graphs.

Wikidata'23: Wikidata workshop at ISWC 2023

2. We perform a case study on 20 Wikidata classes, showing that gap properties provide an intuitive way to understand knowledge imbalances.
3. We analyze the nature of knowledge gaps, finding that on general classes, these mostly relate to generic, logically existing properties, that could in principle be added, while on specific classes, optional achievements stand more out.

2. Related Work

Inequality, bias, and fairness Observing, explaining, and criticizing inequality are fundamental to human society [7, 8]. Notably, the Gini coefficient [8] has become a widespread tool to measure economic inequality in a single number. Inequalities and biases also affect fairness in the information society: it is a widely held belief that inequalities are able to reinforce themselves, e.g., in terms of representations of genders, ethnicities, or educational backgrounds, in various professional roles. As such, society may consider it important to identify inequalities, and to actively tackle them.

Bias and gaps in Wikipedia and Wikidata With the emergence of public world knowledge repositories, the question of their fairness and gaps has emerged. The Wikimedia Foundation considers knowledge gaps a core challenge in their 2030 agenda [9]. A considerable set of studies have focused not only on gender imbalances in Wikipedia [1, 2, 3], but also languages [10], nationalities [4], or individuals [5]. Most interestingly, a study by Abian et al. suggested a methodology to disentangle editor-based gaps from readership-interest-based gaps [6].

The structured format of Wikidata makes statistical analyses especially easy, yet also complicates their interpretation, since gaps may merely stem from technical reasons. A set of tools for helping editors with knowledge gaps in Wikidata have been developed. COOL-WD can be used to display insights about the completeness of subject-property-pairs [11]. ReCoin enables completeness information relative to other entities in the same class [5]. It adds a traffic-light-style status indicator, and lists frequent absent properties. ProWD is a framework and web application tool for profiling the completeness of Wikidata [12, 13]. It notably provides information about group-level knowledge distribution via the Gini index. Our present proposal builds upon the ProWD framework: While the Gini index in ProWD can only provide numeric insights into imbalances, with gap properties, we aim to characterize them.

3. Formal Framework

Knowledge Graphs Knowledge graphs (KGs) organize assertions about entities, such as

Ada Lovelace worked as a computer scientist, or

Ada Lovelace was born on 10 December 1815.

They are stored as triples like (*AdaLovelace*, *occupation*, *ComputerScientist*), or (*AdaLovelace*, *dateOfBirth*, *"10 December 1815"*).

The triples are made up of three kinds of building blocks: items, properties, and literals. In the triples above, *AdaLovelace* and *ComputerScientist* are items, *"10 December 1815"* is a literal, and *occupation* and *dateOfBirth* are properties. Items represent entities, such as people, cities, organizations, and concepts. Literals are scalar values, such as numbers, strings, or dates. Properties are used to assert that entities are related to other entities or atomic values.

Such triples are called statements. They have the generic form (s, p, o), where s plays the role of the subject of the statement, p of the predicate, and o of the object. Subjects are always items, predicates are properties, while objects can be either items or literals. A KG is a set of such triples. We refer the reader to [14] for the Resource Description Framework (RDF), a standard data model for KGs.

In Wikidata, items and literals can be freely created by KG editors. The set of properties, however, is carefully administered. New properties can only be created by a community consensus, since properties constitute the language in which statements are formulated.

Classes KGs have special items that represent classes. Examples in Wikidata are *Human*, *Country*, *Painting*, or *Object*. That an item a is an instance of a class C is expressed by the statement $(a, \mathit{instanceOf}, C)$.

In this paper we apply a more general notion of class. For us, a class is defined by a class item and possibly additional conditions. Technically, this is achieved by means of queries of the form $(?v, P)$, consisting of a set P of triple patterns (i.e., like triples but with the addition of variables) [15], one of which is $(?v, \mathit{instanceOf}, C)$, plus a projection $?v$ onto a single variable. As an example, we can define the class of computer scientists by the query

$$(?v, \{\, (?v, \mathit{instanceOf}, \mathit{Human}), (?v, \mathit{occupation}, \mathit{ComputerScientist}) \,\}).$$

Instantiating that class over the Wikidata graph returns as instances Ada Lovelace and Tim Berners-Lee, among others. We can make such a subclass more specific by adding more triple patterns, defining for instance all computer scientists of a certain nationality.

Information Wealth of Entities Let G be a KG and a an item in G. Moreover, let T_a be the set of triples where a appears as the subject. In a way, the set T_a comprises all the information about (the entity represented by) a that is provided by G: it constitutes the wealth of information about a in G. For example, the wealth of Ada Lovelace in Wikidata comprises all Wikidata triples with Ada Lovelace appearing in the subject position, and that the triples describe a range of Ada Lovelace's properties from image and birth name to audio and plaque image.

To gauge the information wealth of a in G, we can can apply various measures to T_a. A straightforward measure is the cardinality of T_a. Another one is the number of properties occurring in T_a. While the first measure can be skewed by a large number of statements about the same property, the second reflects the variety of information about an entity. It seems natural to expect that often all or at least most true statements about a specific property of an item have been entered into a KG if the property is present. For this reason, we concentrate on the number of distinct properties occurring in T_a, denoted as $\mathit{np}(a)$, as the measure of information wealth of an entity a.

For a given class C, one can study how the values $\mathit{np}(a)$ are distributed for the items a in C. In previous work [13], the distribution of information wealth within classes of items has been analyzed in terms of Gini coefficients. By adopting the Gini coefficient formula for income distribution [16], one may measure the degree of inequality of the information wealth of items in a class (e.g., computer scientist, sovereign state), ranging from 0.0 (i.e., the perfect equality) to 1.0 (i.e., the perfect inequality). However, the Gini coefficient alone does not tell in which way statements of poor items differ from those of rich ones, which is the focus of the present paper.

Properties Associated with Information Richness and Poverty The concepts below are defined with respect to a given class C in a KG G. We consider C as the set of its instances. To keep things simple, we do not mention C explicitly, but assume that it is clear from the context.

We can sort the elements a of C in ascending order with respect to *np*(a). For each integer $k > 0$, we divide C into k *quantiles* with respect to this order, which we denote as $Q_1^{(k)}, \dots, Q_k^{(k)}$. Typical values of k are 4, 5, and 10, where we speak of quartiles, quintiles, or deciles. We want to find out which properties are typical for which quantile, in particular, which properties are typical for the top and bottom quantile $Q_k^{(k)}$ and $Q_1^{(k)}$ for a given k. We often denote the top quantile as *rich* and the bottom quantile as *poor.*

We adapt some vocabulary from the field of association rule mining. We denote the cardinality of a set X as $|X|$.

Let p be a property of the KG G. The *domain* of p is the set of items that appear as subject of some statement with predicate p, that is

$$\mathit{dom}(p) = \{\, a \mid (a, p, o) \in G \text{ for some item or literal } o \,\}.$$

Suppose S is some subset of interest of C, for instance a union of k-quantiles like $S = \mathit{poor} \cup \mathit{rich}$ or simply $S = C$. The *support of* p *relative to* S is the proportion of items in S that are in the domain of p, that is,

$$\mathit{supp}_S(p) = \frac{|\mathit{dom}(p) \cap S|}{|S|}.$$

Let Q be some quantile $Q_i^{(k)}$ of C. We call $Q \to p$ an *association rule* between Q and the property p. For example, *rich* $\to$ *dateOfBirth* is such a rule. The *confidence* in the rule $Q \to p$ is defined as

$$\mathit{conf}(Q \to p) = \frac{|\mathit{dom}(p) \cap Q|}{|Q|}.$$

Clearly, $\mathit{conf}(Q \to p)$ and $\mathit{supp}_Q(p)$ are numerically equivalent.

Finally, we introduce the *lift of the rule* $Q \to p$ *relative to* S, where $Q \subseteq S$. This is defined as

$$\mathit{lift}_S(Q \to p) = \frac{\mathit{conf}(Q \to p)}{\mathit{supp}_S(p)}.$$

Note that the definition of relative lift puts an upper bound on the value $\mathit{lift}_{\mathit{poor} \cup \mathit{rich}}(\mathit{rich} \to p)$. The lift is maximal if all items in the rich quantile have property p and none in the poor quantile. Then $\mathit{conf}(\mathit{rich} \to p) = 1$ while $\mathit{supp}_{\mathit{poor} \cup \mathit{rich}}(p) = 0.5$, so that $\mathit{lift}_{\mathit{poor} \cup \mathit{rich}}(\mathit{rich} \to p) = 2$. The other extreme occurs if no rich item has property p while some poor items do have p. In that case $\mathit{conf}(\mathit{rich} \to p) = 0$, hence $\mathit{lift}_{\mathit{poor} \cup \mathit{rich}}(\mathit{rich} \to p) = 0$.

Gap Properties We are especially interested in properties that occur with some minimum frequency in a class, and are abnormally frequent within the richest decile. Concretely, we call a property p a *gap property* if $supp_{poor \cup rich}(p) \geq 0.1$ and $lift_{poor \cup rich}(rich \rightarrow p) \geq 1.5$. The support threshold is defined as such in order to eliminate a fair number of spurious properties, that is, properties that might appear as gap properties by accident. On the other hand, the lift threshold is determined based on the middle point of the spectrum between total equality (i.e., lift of 1.0) and total inequality (i.e., lift of 2.0).

To be able to compare gaps between classes in a normalized context, we also introduce the *gap property ratio (GPR)*, which we define as the fraction of properties p with $supp_{poor \cup rich}(p) \geq 0.1$ that are gap properties.

4. Experimental Evaluation

Experimental Setup We aim to conduct a gap analysis for a varied set of 20 classes and show that gap properties can provide insights to understand knowledge imbalances for a real-world knowledge graph.

We use an RDF version of a Wikidata dump dated September 30, 2020. More specifically, we take the truthy subset of that dump, focusing on a direct representation of Wikidata assertions without considering qualifiers and references.[1] Moreover, we omit external identifier properties[2] as this allows us to concentrate on properties that describe and link entities from within Wikidata (as opposed to external data sources). Without the removal of such properties, our gap analysis would tend to be heavily biased towards well-known entities in relation to external parties.

As formalized in Section 3, in our experiments we divide our classes of interest into 10 quantiles (that is, deciles) and perform a gap analysis relative to the *poor* and *rich* quantiles.

Our experiment program is developed using Java for the data preprocessing part, and Python for the analysis part. We rely on the Java-based Apache Jena library[3] for processing our RDF data and making it available through SPARQL querying [15]. Once we get hold of the data, we then use our Python program to analyze gap properties. The program is based on SPARQLWrapper[4] for querying RDF, pandas[5] for data analysis, and tqdm[6] for tracking the running progress of the program. Our code for the experiment is available at https://github.com/millerama17/association-analysis.

4.1. Gap Property Analysis

We report on an overview of our gap property analysis over 20 Wikidata classes: human, computer scientist (CS), American CS, German CS, Indonesian CS, football player (FP), Bundesliga FP, Premier League FP, language, painting, painting at the Museum of Modern Art (MoMA), painting at the Louvre, public university, sovereign state, gene, galaxy, taxon, movie, song, and

[1] https://www.wikidata.org/wiki/Wikidata:Database_download

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

[3] https://jena.apache.org/

[4] https://github.com/RDFLib/sparqlwrapper

[5] https://pandas.pydata.org/

[6] https://pypi.org/project/tqdm/

Table 1

Top 10 Most Frequent Gap Properties from 20 Wikidata Classes

Rank	Gap Property	Count
1	image	12
2	Commons category	11
3	name in native language	7
4	award received	6
5	date of death	6
6	family name	6
7	languages spoken, written or signed	6
8	official website	6
9	place of birth	6
10	described by source	5

Table 2

Top 10 Most Frequent Gap Properties from Human-associated Classes: Human, Computer Scientist (CS), American CS, German CS, Indonesian CS, Football Player (FP), Bundesliga FP, Premier League FP

Rank	Gap Property	Count
1	Commons category	7
2	image	7
3	name in native language	7
4	award received	6
5	date of death	6
6	family name	6
7	languages spoken, written or signed	6
8	place of birth	6
9	official website	5
10	place of death	4

Table 3

Top 10 Most Frequent Gap Properties from Non-Human Classes: Language, Painting, Painting at the Museum of Modern Art, Painting at the Louvre, Public University, Sovereign State, Gene, Galaxy, Taxon, Movie, Song, Star

Rank	Gap Property	Count
1	genre	5
2	image	5
3	title	5
4	Commons category	4
5	catalog code	3
6	main subject	3
7	topic’s main category	3
8	coordinate location	2
9	native label	2
10	subclass of	2

star. The selection of the classes is based on the following considerations: human vs. non-human, class vs. subclass, and (in terms of Gini coefficient) equal vs. inequal. The complete list of the gap properties of the 20 classes is available at https://bit.ly/GPR20Classes. Table 1 ranks the 10 most frequent gap properties out of the 20 classes. Furthermore, Table 2 and Table 3 show the 10 most common gap properties for human-associated classes and non-human classes, respectively.

Characterization of gap properties Among the 20 classes examined, a few intriguing observations emerge. In Table 1, "image" (P18) and "Commons category" (P373) are the two most frequent gap properties. Both properties are similar in the sense that they are related to providing images or multimedia files for Wikidata items and could in principle apply to all 20 classes. By being gap properties, it means that the properties "image" and "Commons category" are commonly found in rich Wikidata items but not so in poor ones. The rest of the gap properties in Table 1 are dominated by properties to describe humans as can be confirmed by the content of Table 2 about the top 10 most frequent gap properties from human-associated classes. As for Table 2, we gather a number of remarks. The most notable gap property is "place of birth" (P19), which in principle indeed every human possesses. The designation of "date of death" (P570), "family name" (P734), and "name in native language" (P1559) as gap properties is likely due to the fact that not all humans have passed away, not every culture adheres to the tradition of family names, and some individuals do not have a name variation in their native language.

Table 3 lists top 10 most frequent gap properties for non-human classes. The properties "genre" (P136), "image" (P18), and "title" (P1476) are among the most common gaps. Note that "genre" and "title" (P1476) are gap properties for classes related to creative work like painting (and its subclasses), movie, and song. The gap property of "catalog code" (P528) is exclusive to paintings (and its subclasses). Moreover, "coordinate location" (P625) is found to be a gap property for public university and, interestingly, language.

Gap properties on classes vs. subclasses We also discuss gap properties in the context of class-subclass relationships. Take, for example, the class of computer scientist. There are 11 gap properties of human that are also gap properties of computer scientist, such as "award received" (P166), "country of citizenship" (P27), and "image" (P18). In total, exactly half of all gap properties of computer scientist inherit from those of human. Gap properties of human that are not found in computer scientist include "sport" (P641) and, interestingly, "date of birth" (P569). On the other side, gap properties of computer scientist that are not found in human are, e.g., "Erdős number" (P2021), "doctoral student" (P185), and "field of work" (P101). It is apparent, therefore, that gap properties on more specific subclasses more often are properties that describe specific achievements inside the class, which not necessarily every class member possesses. In contrast, gap properties on more general classes more typically reflect just KG incompleteness.

We now take another example, the class of painting. Out of 11 gap properties of painting at MoMA, as many as 9 are inherited from those of painting. Furthermore, the class of painting at

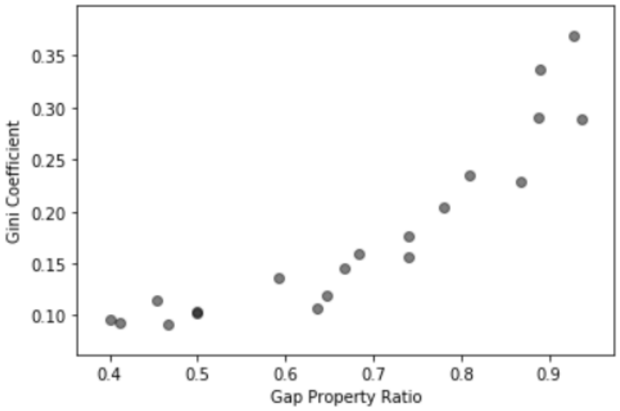

Figure 1: Gap Property Ratio (GPR) vs. Gini Coefficient for 20 Wikidata Classes

the Louvre inherits 7 gap properties from the class of painting; this accounts for 64% of all gap properties in the class of painting at Louvre. Gap properties existing in painting at MoMA but not existing in painting are "copyright holder" (P3931) and "country of origin" (P495), while gap properties found in painting at the Louvre but not found in painting are "Commons category" (P373), "depicts Iconclass notation" (P1257), "exhibition history" (P608), and "movement" (P135). Again, we find that several of these specialized properties do not apply to every item, i.e., they do not necessarily reflect an actionable incompleteness of the investigated knowledge graph.

Gap Property Ratio Figure 1 reveals a clear positive correlation between the Gap Property Ratio (GPR) and the Gini coefficient for each class.[7] Specifically, a higher GPR value corresponds to a higher Gini coefficient value. To quantify this correlation, we apply the Spearman correlation, resulting in the value of 0.95.[8] Given such a high correlation value, it is evident that the GPR can be effectively used to gauge the level of imbalance within a Wikidata class.

Another interesting finding relates to the relationship between classes and their subclasses. Subclasses tend to exhibit lower GPRs and Gini coefficients compared to their superclasses. For instance, the classes "computer scientist" and "football player" possess lower GPR and Gini coefficient values than the broader class of "human". Similarly, more specific classes such as "American CS", "German CS", "Indonesian CS", "Bundesliga FP", and "Premier League FP" show lower GPR and Gini coefficients than their more general classes. This pattern likely arises because general classes encompass a wider array of entities from diverse backgrounds, thus leading to larger gaps. Conversely, filtered or specific classes consist of entities from a more homogeneous group, usually sharing a greater number of common properties.

4.2. Case Studies

We investigate more deeply gaps in the classes of computer scientist and sovereign state. We choose these two classes to showcase an imbalanced class vs. a more balanced one, which we

[7] We refer the reader to https://bit.ly/GPR20Classes for details about the GPR and Gini coefficient of each class.

[8] We choose the Spearman method because it does not rely on linearity nor normality.

Table 4
Top 10 Gap Properties from Computer Scientists and Sovereign States

Rank	Computer Scientist	Sovereign State
1	notable work	coordinates of geographic center
2	name in native language	median income
3	Erdős number	patron saint
4	place of death	seal description
5	doctoral student	category of people buried here
6	Commons category	archives at
7	described by source	compulsory education (maximum age)
8	residence	age of candidacy
9	languages spoken, written or signed	studied by
10	member of	water as percent of area

will later show through our gap property analysis.

Computer Scientist Table 4 lists top 10 gap properties of computer scientists as well as sovereign states, ordered in descending order by the lift values. There are a few noteworthy gap properties from computer scientists. Examples include academic-related properties such as "notable work" (P800), "Erdős number" (P2021), and "doctoral student" (P185). This implies that "poor" computer scientists have no information about, e.g., their notable work. Indeed, the property of "notable work" heavily depends on the popularity of computer scientists (and hence, less popular computer scientists might not have any notable work). "place of death" (P20) and "residence" (P551) are common properties for humans (and hence for computer scientists), yet not all humans have passed away, and not all humans (esp. non-public figures) are willing to share their residence information (due to privacy issues).

The computer scientist class has 22 gap properties out of 27 properties with support ≥ 0.1. The GPR value is therefore 22/27, or approximately 0.81. It has a pretty high GPR which means that most properties of computer scientists are gap properties.

Sovereign State As shown in Table 4, the top-2 gap properties of sovereign states are "coordinates of geographic center" (P5140) and "median income" (P3529), which are actually attributes that all sovereign states should have in the real world. Next, the property of "patron saint" (P417) is however only applicable to countries related to Christianity. Other properties rarely owned by "poor" sovereign states but frequently occurring in "rich" ones include "seal description" (P418),[9] "category of people buried here" (P1791), and "archives at" (P485).

In an absolute number, there are much higher gap properties in sovereign states compared to computer scientists, that is, 60 vs. 22, respectively. This, however, does not mean that the sovereign state class is more imbalanced than computer scientist. On the contrary, due to the larger number of sovereign state properties with support ≥ 0.1, totaling at 146 properties, the GPR of the sovereign state class is indeed pretty low, that is, 60/146, or approximately 0.41 (as opposed to 0.81, the GPR of the computer scientist class).

[9] It is now called "has seal, badge, or sigil".

As lessons learned, imbalances in Wikidata classes can be effectively measured using gap properties and the GPR. Gap properties are useful for identifying which properties constitute the wealth separation between the poor and rich groups in a class, while the GPR provides a tool for comparing the gap level among classes in a normalized context. Findings from our framework may raise the prevasive issue of knowledge gaps, that the problem is real, and actions can be taken by the editors and community to mitigate such an issue. Our framework highlights the essential properties that need data completion, ruling out irrelevant ones, such as "military rank" (P410) in the computer scientist class. Our analysis allows us to understand knowledge gaps and spark initiatives to manage them effectively, promoting accuracy and efficiency in data completion.

5. Conclusions

We have proposed a framework to discover knowledge gaps in Wikidata classes. We have introduced the concepts of gap properties and the gap property ratio (GPR) that can be useful to give insights as to which properties constitute the gaps between the poor and rich groups of a Wikidata class and measure (and compare) the gap level among classes. Our experimental evaluation of gap analysis over 20 classes of Wikidata has shown that knowledge gaps do exist and that awareness to such an issue can be approached scientifically. Especially to Wikidata researchers and contributors, this tool can help them address this phenomenon more swiftly and accurately by identifying the essential properties in Wikidata that constitute imbalances and addressing them accordingly.

Acknowledgement

This work has been partially supported by the project CONFUCIUS, funded by the Free University of Bozen-Bolzano.

References

[1] J. Reagle, L. Rhue, Gender Bias in Wikipedia and Britannica, International Journal of Communication 5 (2011) 1138–1158.

[2] C. Wagner, D. Garcia, M. Jadidi, M. Strohmaier, It's a Man's Wikipedia? Assessing Gender Inequality in an Online Encyclopedia, in: ICWSM, 2015.

[3] F. Tripodi, Ms. Categorized: Gender, notability, and inequality on Wikipedia, New Media & Society 25 (2023) 1687–1707.

[4] Z. Shaik, F. Ilievski, F. Morstatter, Analyzing Race and Citizenship Bias in Wikidata, in: MASS, 2021.

[5] V. Balaraman, S. Razniewski, W. Nutt, Recoin: Relative Completeness in Wikidata, in: WWW (Companion Volume), 2018.

[6] D. Abián, A. Meroño-Peñuela, E. Simperl, An Analysis of Content Gaps Versus User Needs in the Wikidata Knowledge Graph, in: ISWC, 2022.

[7] K. Marx, Das Kapital, Verlag von Otto Meissner, 1867.

[8] C. Gini, Variabilità e Mutabilità: Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche, P. Cuppini, 1912.

[9] M. Redi, I. Johnson, M. Gerlach, L. Zia, Address Knowledge Gaps, Three Years On, Wikimedia Foundation, 2022.

[10] J. M. Dolmaya, Expanding the sum of all human knowledge: Wikipedia, translation and linguistic justice, The Translator 23 (2017) 143–157.

[11] R. E. Prasojo, F. Darari, S. Razniewski, W. Nutt, Managing and Consuming Completeness Information for Wikidata using COOL-WD, in: COLD@ISWC, 2016.

[12] A. Wisesa, F. Darari, A. Krisnadhi, W. Nutt, S. Razniewski, Wikidata Completeness Profiling Using ProWD, in: K-CAP, 2019.

[13] N. H. Ramadhana, F. Darari, P. O. H. Putra, W. Nutt, S. Razniewski, R. I. Akbar, User-Centered Design for Knowledge Imbalance Analysis: A Case Study of ProWD, in: VOILA@ISWC, 2020.

[14] G. Schreiber, Y. Raimond (Eds.), RDF 1.1 Primer, W3C Working Group Note, 24 June 2014. https://www.w3.org/TR/rdf11-primer/.

[15] S. Harris, A. Seaborne (Eds.), SPARQL 1.1 Query Language, W3C Recommendation, 21 March 2013. https://www.w3.org/TR/sparql11-query/.

[16] A. Sen, On Economic Inequality, Oxford University Press, 1997.