Ranking Indicator Discovery from Very Large Knowledge Graphs

Hassan Abdallah
LIFAT, University of Tours
Blois, France
hassan.abdallah@univ-tours.fr

Béatrice Markhoff
CITERES, University of Tours
Blois, France
beatrice.markhoff@univ-tours.fr

Arnaud Soulet
LIFAT, University of Tours
Blois, France
arnaud.soulet@univ-tours.fr

ABSTRACT

Ranking indicators are essential tools for comparing the importance of various entities such as cities or scientists. While extensively used in fields like econometrics and scientometrics, many other domains lack systematic approaches for developing these indicators. In this paper, we introduce a novel method for automatically discovering ranking indicators from very large knowledge graphs. To this end, we formalize the notion of *counting graph pattern* (CG) as a special SPARQL query, and the concept of *ideal ranking indicator* as a CG whose result induces a strict total order on a set of entities. To assess the interestingness of ranking indicators, we employ the *proportion of covered entities* along with an inequality measure, namely the Gini coefficient. We further present Algorithm *Ranking Indicator Pattern Miner (RIPM)*, to efficiently identify interesting ranking indicators for a given field, thanks to pruning techniques for handling the very large search space. Our experimental study shows the effectiveness of our optimizations. It also validates that RIPM extracts *transparent, diverse, and understandable* indicators through a user survey and a comparison with two baselines. This work has significant implications for fields lacking dedicated communities working on ranking tasks, providing a robust tool to automatically produce ranking indicators, and the associated rankings.

PVLDB Reference Format:
Hassan Abdallah, Béatrice Markhoff, and Arnaud Soulet. Ranking Indicator Discovery from Very Large Knowledge Graphs. PVLDB, 18(4): 1183 - 1195, 2024.
doi:10.14778/3717755.3717775

PVLDB Artifact Availability:
The source code, data, and results have been made available on Zenodo: https://zenodo.org/records/14181263

1 INTRODUCTION

Ranking indicators are very common to compare the importance of different entities (like cities or scientists) based on targeted criteria (in the case of cities, the comparison may concern the economy or culture), in particular, to drive public policy [5] or scientific policy [50]. Even if caution is called for [8, 27, 34], the use of numerical indicators objectively establishes a ranking, making it easy to compare entities. Some fields, such as economics or epistemology, even have their own community dedicated to the construction of these ranking indicators, with econometrics and scientometrics. For instance, bibliometric indicators help measure the impact and influence of scientific research, leading to the identification of emerging trends and influential research areas. Unfortunately, many fields do not have a comparable community, even though the needs have been identified. For example, it would be interesting to produce ranking indicators in artistic fields to study the influence of artists and their works to answer questions as simple as "How to compare painters?". In fact, some flawed rankings are available on websites, such as those shown in Table 1 (a)-(c). These three rankings of the 10 most important painters are ultimately very different, with only four painters in common (highlighted in bold). It is not their diversity of viewpoint that's troublesome, but rather their lack of explainability. Indeed, they are based on subjective data (a), on aggregated subjective data (b), or on unavailable usage data (c). More generally, there is a high demand from knowledge workers, such as journalists, for understandable rankings to help them in decision-making [16, 49].

The vast knowledge graphs available on the Web, like DBpedia [4], YAGO [42] or Wikidata [47] could serve as an essential knowledge source to produce ranking indicators. Typically, Wikidata contains several hundred million entities linked by billions of facts. Firstly, these crowdsourced knowledge graphs, that can be considered as mirrors of reality, are publicly available, which guarantees the *transparency* of the ranking indicators. Interestingly, crowdsourcing also makes it possible to describe entities with sufficiently varied relationships to ensure the *diversity* of the ranking indicators. Of course, crowdsourcing also means that they are not exempt from shortcomings regarding correctness, completeness, and representativity, but there are methods for assessing these quality criteria [25, 36, 39]. Secondly, knowledge graphs are data with semantics [6] that enable the construction of *understandable* ranking indicators. Entities are very varied in nature (objects, concepts, people, events, and so on) and they are linked by relationships with a specific meaning. Let us come back to the example of painters, with Table 2 that provides two distinct rankings extracted from Wikidata, based on two distinct criteria: the first considers their number of art exhibitions and the second considers their number of paintings. The two rankings are very different as there is no intersection in the top 10 painters, although both are understandable and relevant for ranking painters. To the best of our knowledge, there is no method for automatically discovering transparent, diverse, and understandable ranking indicators for a given field.

The major challenge in the automatic discovery of ranking indicators is to design a generic method that works for *any* field but returns indicators meaningful to *each* field. Indeed, knowledge graphs like Wikidata are trans-disciplinary, covering a wide range of fields: the arts, biology, linguistics, history, geography, religion and so on. Firstly, a ranking indicator relevant for a target field must be based on quantities that reflect the characteristics of this

Proceedings of the VLDB Endowment, Vol. 18, No. 4 ISSN 2150-8097.
doi:10.14778/3717755.3717775

Table 1: Four rankings of most important painters

(a) theartwolf.com

#	Painter
1	**Picasso**
2	di Bondone
3	**da Vinci**
4	Cézanne
5	**Rembrandt**
6	Velázquez
7	Kandinsky
8	**Monet**
9	Caravaggio
10	van Eyck

(b) ranker.com

#	Painter
1	**Rembrandt**
2	van Gogh
3	Caravaggio
4	Michelangelo
5	**da Vinci**
6	**Monet**
7	Vermeer
8	Raphael
9	**Picasso**
10	Velázquez

(c) artcyclopedia.com

#	Painter
1	**Picasso**
2	van Gogh
3	**da Vinci**
4	**Monet**
5	Dali
6	Matisse
7	**Rembrandt**
8	Warhol
9	O'Keeffe
10	Michelangelo

(d) Wikidata pop.

#	Painter
1	Tagore
2	Shi
3	Rubens
4	J. M. W. Turner
5	N. Bose
6	van der Heijden
7	Gavarni
8	Daumier
9	Diederen
10	J. I. Kraszewski

Table 2: Ranking indicators on Wikidata

(a) Art exhibitions

#	Painter	Nb.
1	Picasso	315
2	Matisse	241
3	Miró	197
4	Klee	166
5	J. Johns	154
6	Léger	145
7	Rauschenberg	143
8	Braque	134
9	Ernst	127
10	Cézanne	122

(b) Paintings

#	Painter	Nb.
1	N. Bose	3,511
2	Rubens	2,043
3	K. Palsa	1,940
4	Munch	1,797
5	van Dyck	1,674
6	Teniers the Y.	1,360
7	Sluijters	1,267
8	Diederen	1,208
9	P.-A. Renoir	1,199
10	Monet	1,141

field, unlike some information retrieval measures that work identically for all entities [10, 11, 13]. For example, simply counting the number of facts associated with a painter (see Table 1 (d)) is too vague to rank the painter, as these facts encompass a varied range of knowledge, some of which may not be related to the field of painting (e.g., spouses or children). In this ranking, which measures popularity more than anything else, Rabindranath Tagore, better known as a poet and philosopher, finds himself ranked as the best painter. Secondly, the heterogeneity of fields makes it illusory to build offline a set of ranking indicators for all possible sets of entities. For this reason, the extraction of a ranking indicator must depend on a user query targeting the entities to be ranked. This raises the problem of on-demand extraction with limited response time, which we propose to perform from a public SPARQL endpoint directly, to avoid downloading a costly dump and to guarantee data freshness. But, as these endpoints do not use preemption [26], they implement fair use policies that prevent the execution of overly complex queries and limit the frequency of querying. Moreover, the need to simplify queries is all the greater given the sheer volume of data involved, and the need to maintain response times that are acceptable to the end-user.

This paper overcomes the previous challenges to automatically discover which related items to count for entities, so that they can be compared according to different complementary meaningful criteria. More specifically, our contributions are as follows:

- We formalize the notion of *ranking indicators* from knowledge graphs. We define the shape of graph patterns, named *counting graph patterns*, corresponding to SPARQL queries counting items. These queries enable the building of the corresponding rankings from the knowledge graph for the user.
- We show that a ranking indicator should cover a large proportion of entities (tending towards a strict total order), and should lead to an unequal distribution. We measure the unequality using the Gini coefficient, that we approximate using a complex network model for the sake of efficiency.
- We detail an efficient algorithm for extracting ranking indicators from a SPARQL public endpoint, named *Ranking Indicator Pattern Miner*. Despite the sheer number of possible ranking indicators in the search space, this is made possible by restricting their syntax and by pruning relationships whose Gini coefficient is too low.

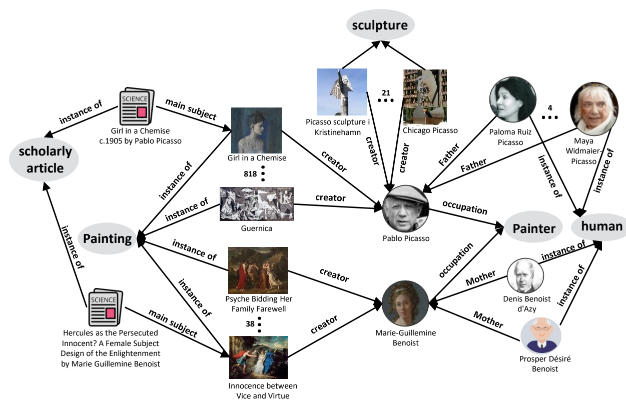

Figure 1: An example of a small KG extracted from Wikidata

- We show with extensive experiments on Wikidata, DBpedia, and YAGO that our method is effective for discovering ranking indicators thanks to our Gini approximation, which is faster than the exact calculation with a minor loss of precision. On several occupations, we also show the diversity of the ranking indicators extracted, as well as the agreement with the best indicators chosen by human raters. Finally, we compare these indicators with an information retrieval measure and with ChatGPT4.

The remainder of this paper is organized as follows. Section 2 provides definitions related to KGs, that are used to define the concept of ranking indicators in Section 3. In Section 4, we show how to effectively evaluate the interestingness of a ranking indicator with the proportion and the Gini coefficient. In Section 5, we present the key optimizations leading to RIPM. Section 6 presents the results of applying RIPM to Wikidata, DBpedia, and YAGO. Section 7 discusses related work and finally, Section 8 concludes the paper.

2 PRELIMINARIES

We rely on the notations and definitions provided by [20, 33].

Knowledge graph Considering distinct infinite sets I and L (IRIs [14] and literals, respectively), a knowledge graph $\mathcal{K} \subseteq I \times I \times (I \cup L)$ is a set of facts. Each fact is a triple $\langle \mathsf{s}, \mathsf{p}, \mathsf{o} \rangle \in \mathcal{K}$, where s, p and o denote respectively the *subject*, the *predicate* (or relationship) and the *object*. For example, ⟨Guernica, creator, Picasso⟩ is a fact meaning that P. Picasso created the painting Guernica. Figure 1 provides a small excerpt of Wikidata with 6 relationships (e.g., instance of or creator) describing several kinds of entities (e.g., persons or paintings).

Basic graph pattern We now consider knowledge graph querying by introducing an infinite set V for the variables. A *triple pattern t* is a triple $\langle \mathsf{s}, \mathsf{p}, \mathsf{o}\rangle \in (I \cup V) \times (I \cup V) \times (I \cup V \cup L)$ allowing variables in any position. For instance, ⟨?item, creator, M.-G. Benoist⟩ is a triple pattern to return subjects of the relationship creator for M.-G. Benoist. Thereby, evaluating this triple pattern to the knowledge graph shown in Figure 1 returns the 38 items created by her (including the painting Psyche Bidding Her Family Farewell). A *basic graph pattern* (or simply graph pattern) combines triple patterns by conjunction: $(P_1 \wedge P_2)$ forms a basic graph pattern if P_i are triples or basic graph patterns themselves. Assuming that P_1 and P_2 are respectively the basic graph pattern ⟨?p, creator, ?ent⟩ and ⟨?item, main subject, ?p⟩, then $P_1 \wedge P_2$ forms a basic graph pattern that queries for all paintings $?p$ created by painters $?ent$ that are main subject of scholarly articles $?item$, along with these articles for each painting. We denote the set of variables in P as $var(P)$. We denote $P(?v, i)$ the function that replaces $?v$ by the IRI i in the graph pattern P. Using the same previous example, $var(P_1 \wedge P_2)$ returns three variables: $?item$, $?p$, and $?ent$. Consequently, $P(?ent,$ P. Picasso$)$ replaces the variable $?ent$ to restrict the query to ask only for paintings and scholarly articles corresponding to Picasso.

Evaluation We now formalize the evaluation of a basic graph pattern P on a knowledge graph $\mathcal{K}$. A mapping μ is expressed as a partial function $\mu : V \rightarrow (I \cup L)$. The triple obtained by replacing the variables in a triple t according to μ is denoted as $\mu(t)$. The domain of μ, represented by $\mathrm{dom}(\mu)$, is the subset of V where μ is defined. Two mappings, μ_1 and μ_2, are considered compatible if, for every variable $?x \in \mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2)$, $\mu_1(?x) = \mu_2(?x)$. In simpler terms, μ_1 and μ_2 are compatible when μ_1 can be expanded with μ_2 to create a new mapping, and vice versa. It is important to observe that two mappings with non-overlapping domains are inherently compatible. Additionally, the empty mapping $\mu_\emptyset$ (with an empty domain) is compatible with any other mapping. Let Ω_1 and Ω_2 be sets of mappings. The join of Ω_1 and Ω_2 is defined as $\Omega_1 \bowtie \Omega_2 = \{\mu_1 \cup \mu_2 \,|\, \mu_1 \in \Omega_1, \mu_2 \in \Omega_2$ and μ_1, μ_2 are compatible mappings$\}$. The evaluation of a basic graph pattern P over a knowledge graph $\mathcal{K}$, denoted by $[[P]]_{\mathcal{K}}$, is recursively defined as follows:

(1) If P is a triple pattern t, then $[[P]]_{\mathcal{K}} = \{\mu \,|\, \mathrm{dom}(\mu) = \mathrm{var}(t)$ and $\mu(t) \in \mathcal{K}\}$.
(2) If P is $(P_1 \wedge P_2)$, then $[[P]]_{\mathcal{K}} = [[P_1]]_{\mathcal{K}} \bowtie [[P_2]]_{\mathcal{K}}$.

Continuing with the previous example, by applying $P = P_1 \wedge P_2$ to the knowledge graph $\mathcal{K}$ depicted in Figure 1, in $[[P]]_{\mathcal{K}}$ two scholarly articles and their corresponding painting are returned: the "Girl in a Chemise c.1905 by P. Picasso" article and the *Girl in a Chemise* painting for P. Picasso, and the "Hercules as the Persecuted Innocent? A Female Subject Design of the Enlightenment by Marie Guillemine Benoist" article and the *Innocence between Vice and Virtue* painting for M.-G. Benoist. We define $val(P, ?v) = \{\mu(?v) \;:\; \mu \in [[P]]_{\mathcal{K}}\}$ which returns the set of IRIs that match the variable $?v$ of the graph pattern P in $\mathcal{K}$, and $val_d(P, ?v)$ which ensures the returned results are devoid of duplication. For instance, by applying P to the graph in Figure 1, $val(P, ?ent)$ will several times include the painters P. Picasso and M.-G. Benoist while $val_d(P, ?ent)$ will contain exactly one of each of these painters.

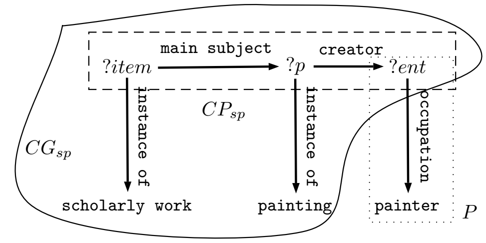

Figure 2: The counting graph pattern CG_{sp} with its path CP_{sp} with a range pattern P

3 RANKING INDICATORS

This section formalizes the problem of ranking indicator discovery. More precisely, we introduce a specific form of basic graph pattern, named *counting graph pattern*, that associates a number of items with each entity (see Section 3.1). Then, we define a ranking indicator as a counting graph pattern inducing an (almost) total order over the entities to be ranked (see Section 3.2).

3.1 Counting graph patterns

For an entity, a ranking indicator corresponds to a count of the items associated with it. For example, it may be interesting to count the number of scholarly works devoted to an artist's paintings. On the one hand, the items counted must be linked to the corresponding entity by a path of relationships (here, the relationships main subject and creator). On the other hand, not all items need to be counted, restrictions should be added (for example, restricting creations to only paintings via ⟨$?item$, instance of, painting⟩). We formalize these intuitions by introducing the notions of *counting path pattern* (Definition 3.1) and *counting graph pattern* (Definition 3.2). The following definition introduces the notion of *counting path pattern* (which is the simplest form of property path [22]):

Definition 3.1 (Counting path pattern). A counting path pattern CP is a conjunction of triple patterns $\{\langle ?v_1, p_1, ?v_2\rangle \wedge \langle ?v_2, p_2, ?v_3\rangle \wedge \cdots \wedge \langle ?v_{n-1}, p_n, ?v_n\rangle\}$ such that $?v_i = ?v_j \Rightarrow i = j$ for $i, j \in \{1, \ldots, n\}$.

This definition means that a counting path pattern is a set of triple patterns forming an acyclic path. For example, $CP_{sp} = \langle ?item,$ main subject$, ?p\rangle \wedge \langle ?p,$ creator$, ?ent\rangle$ is the counting path pattern that links $?item$ with $?ent$ by the fact that the item's main subject is something created by $?ent$. We define $var_{item}(CP)$ (resp. $var_{entity}(CP)$) as the function returning the unique variable appearing only as subject (resp. object) within a counting path pattern CP. Note that Definition 3.1 ensures that $var_{item}(CP) \neq var_{entity}(CP)$. With the counting path pattern CP_{sp}, we get $var_{item}(CP_{sp}) = ?item$ and $var_{entity}(CP_{sp}) = ?ent$. Thereafter, without loss of generality, we choose to denote $var_{item}(CP)$ and $var_{entity}(CP)$ by $?item$ and $?ent$ respectively.

Clearly, the evaluation of the counting path pattern CP_{sp} will mix paintings and sculptures for $?p$ and scholarly works and novels for $?item$. Consequently, this leads to a ranking that confuses items of various kinds. In order to count only scholarly works devoted to an artist's paintings, we need to use more complicated patterns than paths. To this end, we define the notion of *counting graph patterns* as follows:

Definition 3.2 (Counting graph pattern). A counting graph pattern, denoted by CG, is a basic graph pattern such that there exists one and only one counting path pattern $CP \subseteq CG$ with $var(CP) = var(CG)$ and no variable as relationship. The unique counting path pattern CP of CG is denoted by $\widetilde{CG}$.

In the following, $\mathcal{CG}$ denotes the set of counting graph patterns. Definition 3.2 means that triple patterns of CG not belonging to $\widetilde{CG}$ are of the form $\langle ?v, p, o\rangle$ or $\langle s, p, ?v\rangle$, where $?v \in var(\widetilde{CG})$, $s \in I$, $p \in I$, $o \in (I \cup L)$. With Definition 3.2, it is now possible to target the counting precisely. For example, the counting graph pattern $CG_{sp} = CP_{sp} \wedge \langle ?item, \texttt{instance of}, \texttt{scholarly work}\rangle \wedge \langle ?p, \texttt{instance of}, \texttt{painting}\rangle$ can be used to count scholarly articles about paintings of creators. Figure 2 depicts CG_{sp} (solid line) with its counting path pattern CP_{sp} (dashed line).

Now we can specify how to associate a score with an entity, based on a counting graph pattern:

Definition 3.3 (Counting for an entity). The counting of the entity e for the counting graph pattern CG, denoted by $\#CG(e)$, is defined as: $\#CG(e) = |val(CG(?ent, e), ?item)|$.

Let us consider the ranking based solely on the number of paintings (as in the right part of Table 2) with the counting graph pattern $CG_p = \langle ?item, \texttt{creator}, ?ent\rangle \wedge \langle ?item, \texttt{instance of}, \texttt{painting}\rangle$. Then, by applying it to the knowledge graph illustrated in Figure 1, we get $\#CG_p(Picasso) = 818$ corresponding to the number of paintings created by P. Picasso. Applying the previous counting graph pattern CG_{sp}, it gives that $\#CG_{sp}(Picasso) = 1$ since in this graph, there is only one painting created by P. Picasso which is declared to be the subject of a scholarly article. With a counting graph pattern, it is easy to define a partial order on entities: the ranking induced by CG, denoted by $\leq_{CG}$, over the entities $val_d(CG, ?ent)$ is defined as $e_1 \leq_{CG} e_2 \Leftrightarrow \#CG(e_1) \leq \#CG(e_2)$.

In this way, it is possible to induce a ranking for any counting graph pattern, but not all of them are relevant. Typically, it would be possible to count the number of children to rank painters, but this is not very meaningful. We have computed that there are over 10 thousand possible counting graph patterns in Wikidata for painters alone. For this purpose, we explain in the next section which properties a counting graph pattern needs to satisfy in order to be meaningful for ranking entities.

3.2 Range pattern and ideal ranking indicator

User-specified range pattern. Our goal is to automatically find good ranking indicators specific to a user-specified subset of entities (e.g., painters or teachers). For example, it seems natural to rank painters according to their number of paintings, but this indicator makes no sense for ranking teachers. For this purpose, we propose that the end-user targets the entities to be ranked with a basic graph pattern, named *range pattern*:

Definition 3.4 (Range pattern). A range pattern is a basic graph pattern P such that $var(P) = \{?ent\}$ is a singleton. The entities in P are the valuations of its unique variable: $val_d(P, ?ent)$. When the context is clear, we use only P to denote these valuations.

A range pattern P characterizes the entities on which the end-user wants to build a ranking, based on a counting graph pattern CG such that $var_{entity}(CG) = var(P)$. The user can specify an occupation or a class, as in our experiments (see Section 6), but it is also possible to add other, more specific filters to the range pattern (e.g., citizenship or period for artists). This paves the way for a high degree of interactivity, so that ranking indicators can be designed to best meet the desired objective. For instance, in Figure 3, the end-user uses the range pattern $P = \{\langle ?ent, \texttt{occupation}, \texttt{painter}\rangle\}$ for the question "How to compare painters in Wikidata?".

Intuitively, a ranking is an order on a set: the range pattern P specifies this set, while the order is defined by the counting graph pattern CG, more precisely by $\#CG(e)$, for each entity e in P. For instance, in Figure 2, $P = \{\langle ?ent, \texttt{occupation}, \texttt{painter}\rangle\}$ selects the painters among all the objects of the `creator` property in CG_{sp}. Evaluated on the graph in Figure 1, this range pattern returns $val_d(P, ?ent) = \{\texttt{P. Picasso}, \texttt{M.-G. Benoist}\}$ (which is a very small subset of the real 291, 617 painters stated in Wikidata).

Relevant ranking indicator. There are numerous works in the literature proposing measures for ranking entities (see Section 7), but to the best of our knowledge, there is no formal definition of what a ranking indicator is. Intuitively, a ranking relies on a numerical function that outputs a different value for two different entities. For example, it is easier to distinguish two painters by comparing their number of paintings (different in 84% of cases in Wikidata) than their number of children (different in only 8% of cases). This intuition echoes several works in the field of multi-criteria decision making, highlighting the interest of having numerical indicators leading to a total order on the objects to be ranked [31]. Following this direction, we claim that *ideally*, a ranking indicator RI for the entities of P would allow to obtain the count $\#RI$ that induces a strict total order on the entities of a range pattern P.

Definition 3.5 (Ideal ranking indicator). Given a range pattern P, an *ideal ranking indicator* RI for entities in P is a counting graph pattern such that $var_{entity}(RI) = var(P)$ and $\#RI(e_1) \neq \#RI(e_2)$ for all entities $(e_1, e_2) \in P \times P$.

This definition means that the order $\leq_{RI}$ induced by an ideal ranking indicator RI must be a strict total order. In practice, some entities may be ex aequo, especially for low-count entities (e.g., although the number of paintings is discriminating, the order is far from total, with 21,834 painters having a single painting, 9,714 having two paintings and so on). For this reason, there is little chance of identifying ideal ranking indicators in real-world knowledge graphs. Then, we aim at automatically finding ranking indicators that allow us to rank as many entities as possible within a given set of entities. Problem 1 formalizes this intuition:

Problem 1. *Given a knowledge graph $\mathcal{K}$ and a range pattern P, we aim at finding the ranking indicator RI for the entities in P maximizing the number of entity pairs in P that are strictly comparable:*

$$\arg\max_{RI \in \mathcal{CG}} |\{(e_1, e_2) \in P \times P \ : \ \#RI(e_1) \neq \#RI(e_2)\}|$$

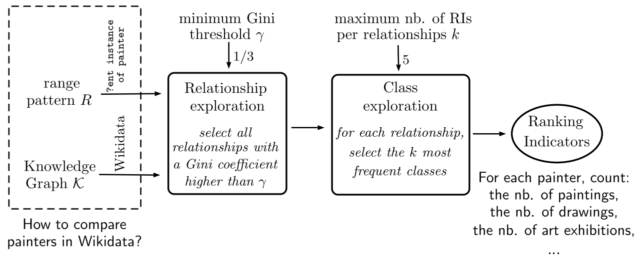

Figure 3: Overview of Algorithm RIPM

Since we want to answer this problem on demand, there are two major challenges to address:

(1) **Costly interestingness evaluation:** Calculating the number of pairs of incomparable entities is costly to compute from a public SPARQL endpoint since you need to have the count $\#RI(e)$ for each entity e in P.
(2) **Huge search space:** The search space for counting graph patterns $\mathcal{CG}$ is very large making naive enumeration of individual patterns impossible.

To address these two challenges, (1) we reformulate the interestingness of a ranking indicator to simplify its computation, by relying on the Gini coefficient (see Section 4), and (2) we prune the search space by successively exploring the set of relationships and classes (see Section 5), as depicted by Figure 3.

4 FAST INTERESTINGNESS EVALUATION

In order to assess the quality of a RI indicator, Problem 1 proposes to calculate the number of entity pairs that are comparable with RI requiring the score $\#RI(e_i)$ for each entity $e_i \in P$, which is very costly. Rather than calculating this number of pairs, this section shows how it is possible to evaluate the interest of a ranking indicator with only 3 values: the number of entities to be ranked (i.e., $n_P = |val_d(P, ?ent)|$), the number of entities with a non-zero score (i.e., $n_e = |val_d(CG, ?ent)|$) and the total number of items (i.e., $n = |val(CG, ?item)|$). To do this, we decompose the interestingness of a ranking indicator with two measures: the *proportion* of entities that can be ranked (see Section 4.1) and the *Gini coefficient* to measure the dispersion between these entities (see Section 4.2).

4.1 Proportion

Intuitively, a ranking for entities in P is all the more interesting as it covers many entities (i.e., where $\#RI(e_i) > 0$), because this enables more entities to be compared with each other. For example, the number of children does not allow us to distinguish many painters, as 96% of painters have no children stated in Wikidata. We introduce the notion of proportion to take this factor into account:

Definition 4.1 (Proportion). Given a range pattern P, the proportion of entities in P covered by a ranking indicator RI such that $var_{entity}(RI) = var(P)$ is defined as $Prop(RI) = n_e/n_P$ where $n_e = |val_d(RI, ?ent)|$ and $n_P = |val_d(P, ?ent)|$.

As $val_d(RI, ?ent) \subseteq val_d(P, ?ent)$, the proportion is necessarily between 0 and 1. For example, for P representing painters, the proportion for the number of paintings is 61% (= 178, 126/291, 617) while the proportion for the number of sculptures is only 7% (= 19, 576/291, 617). In order to have a ranking covering more painters, it is more interesting to count the number of paintings than the number of sculptures. The fact that a painter has both paintings and sculptures, as Picasso in Figure 1, distinguishes him from others, as *M.-G. Benoist* who has paintings but no sculptures. Nevertheless, it is not the best criteria for not only comparing them but for *ranking* them as painters.

Increasing the proportion is necessary to ensure that as many pairs of entities as possible are strictly comparable with each other. The below property states that for an ideal ranking indicator the proportion must be close to 1:

Property 1 (Proportion maximization). *Given a knowledge graph $\mathcal{K}$, the proportion of an ideal ranking indicator RI approaches 1 when the number of entities approaches infinity:*

$$Prop(RI) \xrightarrow[|P|\to\infty]{} 1$$

Due to the lack of space, we omit the proofs of properties, but they are provided with the code, sources and results. Property 1 considers the extreme case of ideal ranking indicators, which requires the proportion to be maximized. More generally, a very high proportion is needed to guarantee that the ranking is discriminating (i.e., with many pairs of strictly comparable entities).

4.2 Gini coefficient

Definition. Ideally, a ranking indicator should lead to a total order where e_i is ranked in i-th place. This means that the difference between the two counts of entities e_i and e_j would be at least $j - i$ items (corresponding to the number of ranks separating them): $\#RI(e_i) - \#RI(e_j) \geq j - i$ for $i \leq j$. Besides, the greater the gap $\#RI(e_i) - \#RI(e_j)$ between two entities, the more significant the ranking would be, since the best ranked would be ranked more highly. In other words, we are looking for a very unequal distribution between the entities in P. To measure this level of inequality of a distribution, we resort to a measure of concentration [9]. We opt for the Gini coefficient [18] which is a measure of concentration regularly used in economics to estimate income inequalities, but also in bibliometrics [35, 37][1]. More precisely, the Gini coefficient is a statistical measure evaluating the level of inequality of the distribution of a variable in a population:

Definition 4.2 (Gini coefficient). The Gini coefficient of a ranking indicator RI for n_e entities $\langle e_1, e_2, \ldots, e_{n_e}\rangle$ in P where $\#RI(e_i) \geq \#RI(e_{i+1})$ is calculated as follows :

$$Gini(RI) = \frac{2 \times \sum_{i=1}^{n_e}(n_e - i + 1) \times \#RI(e_i)}{n_e \times \sum_{i=1}^{n_e} \#RI(e_i)} - \frac{n_e + 1}{n_e}$$

It increases between 0 and 1 with the level of inequality, where 0 means perfect equality and 1, perfect inequality. The Gini coefficient can also be represented graphically as twice the area between the identity and the Lorenz curve. Figure 4 represents the Lorenz curve for CG_p in blue (i.e., for k entities among the painters, the ordinate represents $\sum_{i=1}^{k} \#RI(e_i)$ in percentage). In this representation, perfect equality corresponds to the identity (solid black line) and the strongest inequality to the curve $\{(0, 0), (1, 0), (1, 1)\}$.

[1]The same methodology can be applied with other inequality measures such as the Atkinson index [3] or the generalized entropy index [38].

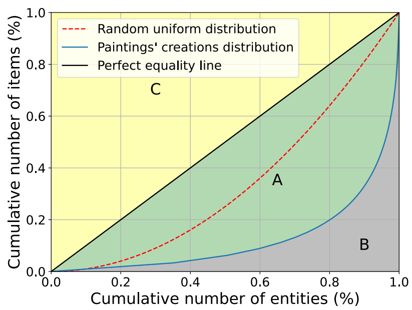

Figure 4: Lorenz curve of the distribution derived from CG_p

Fast Gini coefficient computation. Using Definition 4.2 to compute the Gini coefficient is resource-intensive, particularly when aiming to identify ranking indicators for fields extensively represented in knowledge graphs (e.g., painters). This method requires retrieving all entities e_i (e.g., P. Picasso) and, for each one, counting the number of items $\#RI(e_i)$. Besides, in order to find ranking indicators, we need to consider a lot of counting graph patterns (because the search space $\mathcal{CG}$ is large). Instead of measuring all the counts, it would be better to directly approximate the area A of Figure 4 using a model of the data. For this purpose, we benefit from our previous work [2] showing that the distribution of items for entities follows a power law of exponent $\alpha = 1 + \frac{1}{1-n_e/n}$. Based on this distribution, we prove the main important theoretical result of this paper by approximating the Gini coefficient:

Theorem 4.3 (Gini approximation). *Given a ranking indicator RI, the Gini coefficient of RI is approximated by the following formula:*

$$\widetilde{Gini}(n_e, n) = \frac{1 - n_e/n}{1 + n_e/n}$$

where $n_e = |val_d(RI, ?ent)|$ *and* $n = |val(RI, ?item)|$.

Proof. Let us consider that the distribution of items for entities follows a power law of exponent $\alpha = 1 + \frac{1}{1-n_e/n}$. [28] demonstrates that the fraction W of the wealth in the hands of the richest P of the population is $P^{(\alpha-2)/(\alpha-1)}$. We benefit from this result on the Lorenz curve (see Figure 4) by transposing the ordinate axis as W and the abscissa axis as P. We can then calculate the area C (above the identity which is equal to 0.5) and the area A (between the identity and the Lorenz curve) with the following integral:

$$A + 0.5 = \int_0^1 W\, dP = \int_0^1 P^{(\alpha-2)/(\alpha-1)}\, dP$$

Evaluating this integral, we get $A = 1/(4\alpha - 6)$. Geometrically, the Gini coefficient $Gini$ is equal to $A/(A + B)$ where B is the area below the Lorenz curve. Since the Lorenz curve ranges from 0 to 1 on both axes, we have $A+B = 0.5$ leading to $Gini = 2\times A$. Therefore, we get $Gini = 1/(2\alpha - 3)$. Finally, by injecting the exponent $\alpha = 1+\frac{1}{1-n_e/n}$ in this formula, we prove that Theorem 4.3 is correct.

□

Theorem 4.3 is central to effectively assessing the value of a ranking indicator with little loss of quality compared with an assessment that would cover the whole distribution. For instance, for the ranking indicator CG_p computed on Wikidata, $Gini(CG_p) = 0.763$ is approximated by $\widetilde{Gini}(62724, 672581) = 0.829$. In practice, this approximation is frequently good and it reduces considerably the execution times as shown in the experimental section (see Section 6.2).

4.3 Problem reformulation

Maximizing the proportion and the Gini coefficient tends to improve the quality of the ranking. Interestingly, these two measures can be calculated by using only the 3 values of n_P, n_e and n, making it inexpensive to evaluate the interestingness of a ranking indicator in order to select the most interesting ones, even on very large-scale knowledge graphs. We therefore propose to reformulate Problem 1 as follows:

Problem 2. *Given a knowledge graph $\mathcal{K}$ and a range pattern P, we aim at finding the ranking indicator RI for the entities in P maximizing* $Gini(RI) \times Prop(RI)$*:*

$$\arg\max_{RI \in \mathcal{CG}} Gini(RI) \times Prop(RI)$$

Starting from a range pattern P, Problem 2 focuses on only one ranking indicator RI whose interestingness maximizes $Gini(RI) \times Prop(RI)$ (for maximizing at the same time the Gini and the proportion). As mentioned in Introduction, it is important not to limit ourselves to a single ranking. The following section shows how to apply this problem formulation to select k ranking indicators based on different relationships.

5 OPTIMIZING RANKING INDICATOR EXTRACTION

Before presenting Algorithm RIPM, we explain how to reduce the search space $\mathcal{CG}$ in a two-step process as shown in Figure 3, by exploring only one promising relationship at a time (see Relationship exploration in Section 5.1) and by syntactically restricting ourselves in order to get the k most frequent classes of items (see Class exploration in Section 5.2).

5.1 Relationship exploration

Unfortunately, the number of candidate counting graph patterns in $\mathcal{CG}$ is very high, making it impossible to enumerate them naively to select the right ranking indicators. To cope with this problem, we propose restricting ourselves to patterns (i) whose path length is 1 and (ii) whose Gini is greater than a minimum threshold γ.

At first sight, restricting ourselves to counting graph patterns whose path length is 1 could appear as a strong limitation, as longer paths allow us to target relevant information (e.g., the counting graph pattern CG_{sp} requires two relationships: `main subject` and `creator`). However, it is possible to apply our approach repeatedly to recombine patterns thanks to the following property:

Property 2 (Concatenation). *Given two counting graph patterns CG_1 and CG_2 such that $var_{entity}(CG_1) = var_{item}(CG_2)$ and $var_{item}(CG_1) \neq var_{entity}(CG_2)$, $CG_1 \wedge CG_2$ is also a counting graph pattern.*

Property 2 essentially derives from the fact that putting two paths end-to-end results in a new path. For example, our approach will return for painters the ranking indicator CG_p and for paintings the ranking indicator $CG_s = \langle ?item, \texttt{main subject}, ?ent\rangle \wedge$

$\langle ?item,$ `instance of`, `scholarly work`$\rangle$. It is then possible to concatenate them to obtain the more complex ranking indicator $CG_{sp} = CG_s \wedge CG_p$ (by renaming the entity of CG_s and the item of CG_p with the new variable name $?p$).

Limiting ourself to a single relationship greatly reduces the search space, especially as it is possible to avoid exploring unpromising ones. As already mentioned, the relationship `father`, which is not interesting for ranking painters, has a Gini coefficient equal to 0.269. As a reminder, an ideal ranking indicator has a proportion close to 1. Similarly, the following property states that an ideal indicator must have a Gini coefficient greater than 1/3:

Property 3. *Given a knowledge graph $\mathcal{K}$, the Gini coefficient of an ideal ranking indicator RI for a range pattern P is greater than* $1/3$ *when the number of entities in P approaches infinity.*

This property follows from the fact that the Gini coefficient of an ideal ranking indicator is at least greater than that of a uniform distribution. In Figure 4, the Lorenz curve corresponding to the uniform distribution is shown in red dashed line. The curve corresponding to the CG_p ranking indicator is clearly under it.

Once again, we want to extract ranking indicators that are not necessarily ideal. Nevertheless, it is clear that this theoretical threshold is relevant for eliminating less relevant relationships, which motivates the introduction of a minimum Gini threshold γ in RIPM (see Figure 3 and Algorithm 1).

5.2 Class exploration

Using only relationships often leads to insufficiently precise rankings. For example, counting the number of creations of an entity remains vague. If we want to explore Picasso's contribution to art, it makes sense to count paintings and sculptures separately. More generally, we consider items subjects of the single relationship, but only those belonging to a specific class. To achieve this, we must specify the class to which the items should belong. Since a class encompasses items that share certain characteristics, these characteristics are also likely to be shared with the entities of the target field. To do this, we choose to consider, for each candidate relationship, the top k most frequent classes for its subjects (i.e., classes that encompass the greatest number of subjects). This method is based on the intuition that for an entity type, the most frequent classes should logically be the most relevant ones. Note that maximizing frequency means maximizing n, which also maximizes the Gini coefficient (see Definition 4.3). For instance, considering the creator relationship with painters as a range pattern, painters have created more paintings than photographs, with 673,198 and 2,524 respectively in Wikidata. The variable k indicates how many classes need to be considered per relationship, and it is configurable based on user needs. As with the path-length constraint of the previous section, this does not call into question the generality of our method. Indeed, as the k classes are extracted in parallel, it is possible to merge two patterns with a union. In the case of Picasso, for example, we can find out his art work by combining paintings and sculptures.

5.3 Algorithm RIPM

Ranking Indicator Pattern Miner, or RIPM (see Algorithm 1), generates from a knowledge graph $\mathcal{K}$ the set of (length 1) ranking indicators $\mathcal{RI}$ for entities in P, with at most k item's classes per relationship having a Gini coefficient greater than γ. Note that it is quite easy for users to set the number of rankings k they want for each relationship selected by the Gini threshold γ ($k = 5$ in our experiments). For the threshold γ, the user can set it to 1/3 by following Property 3, or slightly below, to be sure of not missing anything.

Algorithm 1 Ranking Indicator Pattern Miner

Require: A knowledge graph $\mathcal{K}$, a range pattern P where $var(P) = \{?ent\}$, a minimum Gini threshold γ, a maximum number of counting patterns per relationship k
Ensure: The set $\mathcal{RI}$ of ranking indicators with Gini and support for P (with at most k patterns per relationship)

1: $\mathcal{RI} \leftarrow \emptyset$
2: $GP \leftarrow \langle ?item, ?r, ?ent\rangle \wedge P$
3: $n_P = |val_d(P, ?ent)|$
4: **for** $rel \in val_d(GP, ?r)$ **do** ▷ Relationship exploration
5: $\quad CG \leftarrow GP(?r, rel)$
6: $\quad n_e \leftarrow |val_d(CG, ?ent)|$
7: $\quad n \leftarrow |val(CG, ?item)|$
8: $\quad$ **if** $\widetilde{Gini}(n_e, n) \geq \gamma$ **then**
9: $\quad\quad CG \leftarrow CG \wedge \langle ?item,$ `instance of`$, ?class\rangle$
10: $\quad\quad classSet \leftarrow \emptyset$
11: $\quad\quad$ **for** $class \in val_d(CG, ?class)$ **do** ▷ Class exploration
12: $\quad\quad\quad classSet \leftarrow classSet \cup \{(class, |val_d(CG(?class, class), ?item)|)\}$
13: $\quad\quad$ **end for**
14: $\quad\quad$ Keep the k most frequent classes from $classSet$
15: $\quad\quad$ **for** class $\in ClassSet$ **do** ▷ Ranking indicator evaluation
16: $\quad\quad\quad RI \leftarrow CG(?class, class)$
17: $\quad\quad\quad n_e^{class} \leftarrow |val_d(RI, ?ent)|$
18: $\quad\quad\quad n^{class} \leftarrow |val(RI, ?item)|$
19: $\quad\quad\quad \mathcal{RI} \leftarrow \mathcal{RI} \cup \{(RI, \widetilde{Gini}(n_e^{class}, n^{class}), n_e^{class}/n_P)\}$
20: $\quad\quad$ **end for**
21: $\quad$ **end if**
22: **end for**
23: **return** $\mathcal{RI}$

In lines 1-3 of Algorithm 1, we initialize several variables: the set of ranking indicators $\mathcal{RI}$ is set to the empty set, a basic graph GP is created to find the compatible relationships with entities in the range pattern P, and n_P gets the number of entities in P. Subsequently, the main loop (lines 4-22) iterates over all relationships $val_d(GP, ?r)$, replacing the variable $?r$ in GP with the current relationship rel. Lines 6 and 7 respectively compute the number of entities and items. Proceeding with only relationships that have a Gini coefficient greater than γ, a simple triple pattern is concatenated onto the current CG, containing a restriction on the variable $?item$ to be an instance of a $?class$ (line 9). The next task in lines 10-14 is to identify the most frequent classes to rank the entities in P. For each class $class$, line 12 counts the number of items and appends the result to $classSet$. We then select the k most frequent classes from this list (line 14). To evaluate their interest, the loop (lines 15-20) iterates over these k classes, replacing the variable $?class$ with the value $class$ leading to the ranking indicator RI (line 16). We calculate its Gini coefficient, as well as its proportion (line 19). For this purpose, the number of entities n_e^{class} (resp. items n^{class}) covered by the ranking indicator RI is computed at line 17 (resp. line 18). Note that the proportion also requires the initial

Table 3: Main statistics of the three KGs

KG	DBpedia	Wikidata	YAGO4
#rel. $\lvert\mathcal{R}\rvert$	15,100	1,484	85
#facts	1,082,635,010	2,404,397,928	282,056,110
#subjects	245,113,095	431,875,708	48,957,049
#objects	93,130,240	136,870,611	19,214,906

number n_P of entities in P computed at line 3. Thus, we append the computed measures along with their corresponding ranking indicator in the list $\mathcal{RI}$. Finally, the algorithm returns the final list of ranking indicators with their Gini and proportion at line 23. Thus, RIPM has a complexity of $O(r \cdot (c \log c + k))$, assuming that r is the total number of possible relations (line 4), c is the total number of possible classes (line 11), and k is the number of selected most frequent classes.

6 EXPERIMENTAL STUDY

Our experiments aim to evaluate the following aspects: the pruning influence of the minimum Gini threshold (Section 6.1), the Gini coefficient approximation (Section 6.2), the diversity of the extracted ranking indicators (Section 6.3), the agreement between the Gini-proportion interestingness and human raters (Section 6.4) and finally, a comparison of RIPM with two baselines: ranking by popularity and ChatGPT4 (Section 6.5).

Protocol Experiments were performed on Wikidata [47], DBpedia [4], and YAGO [32] via their public SPARQL endpoints [2] in June 2024. Table 3 indicates the main statistics of these three crowd-sourced KGs: number of relationships $|\mathcal{R}|$, number of facts, number of distinct subjects, and number of distinct objects. We use a personal machine equipped with an Intel Core i7-8650U CPU 1.90GHz with 32GB of RAM. RIPM is implemented in Python language with multithreading to parallelize SPARQL query executions. The source code, data, results, and all materials needed to reproduce the findings, is publicly available on the Git repository https://scm.univ-tours.fr/habdallah/RIPM/ and Zenodo https://zenodo.org/records/14181263.

We apply RIPM with the number of classes k set to 5 and the minimum Gini threshold γ set to 0.1[3] (except for Section 6.1) on two sets of range patterns:

- occupations: The top 100 largest occupations in Wikidata, DBpedia, and YAGO are used (e.g., $\langle ?ent, occupation, painter \rangle$). Leading to 21,235, 19,249, and 3,402 ranking indicators for Wikidata, DBpedia, and YAGO, respectively.
- classes: The classes appearing in the ranking indicators discovered for occupations were in turn used as range patterns. For instance, $\langle ?ent, \texttt{instance of}, \texttt{painting} \rangle$ is considered because it occurs in CG_p extracted above.

Interestingly, it is then possible to combine patterns extracted with occupations and with classes using Property 2 to generate 1,231,352, 2,459,452, and 19,750 potential ranking indicators for Wikidata,

[2] query.wikidata.org ; dbpedia.org/sparql ; yago-knowledge.org/sparql/query

[3] We use 0.1 instead of 1/3 (see Property 3) to evaluate the algorithm's behavior more widely (see Section 6.2), and to obtain lower-quality rankings useful for the human survey (see Section 6.4).

Table 4: Statistics for occupations and classes

KG	Type	#*P*	#*RI*	#Filtered RI	Avg. #RI per *P*	#Dist. rel.	Avg. dist. rel. per *P*
Wikidata	occ.	100	21,235	1,576	212.4	269	71.4
	classes	1,304	45,401	1,054	35.2	748	12.9
	all	1,404	66,636	2,630	48.0	818	17.1
DBpedia	occ.	100	19,249	606	192.5	357	40.2
	classes	153	84,267	7,155	550.8	3,946	119.2
	all	253	103,516	7,761	409.2	3,962	87.9
YAGO	occ.	100	3,402	173	34.4	30	7.9
	classes	145	2,050	75	14.2	45	3.9
	all	245	5,452	248	22.4	46	5.5

DBpedia, and YAGO, respectively. In Table 4, the third and fourth two columns provide the number of range patterns and the total number of ranking indicators for each knowledge graph.

Baselines To the best of our knowledge, no method exists in the literature that is directly comparable to RIPM. Therefore, we have compared the results of our approach with two other baselines:

(1) **Popularity**: An information retrieval baseline that ranks entities based on the total number of facts associated with them, without distinguishing between fact types [13]. As the number of facts is the same regardless of the range pattern, it reflects a general popularity (see Table 1 (d) for painters where the highest-ranked entity is a philosopher).
(2) **ChatGPT4**: We use this famous and widely used large language model [29] (i) to rank given indicators from best to worst (see Section 6.4) and (ii) to generate relevant ranking indicators of the form $\langle ?item, PROP, ?ent \rangle \wedge \langle ?item, \texttt{instance of}, CLASS \rangle$ (see Section 6.5).

6.1 Impact of the minimum Gini threshold

We first measure the impact of the minimum Gini threshold γ on the pruning of the search space. The primary objective of this threshold in RIPM is to optimize computation by pruning relationships that are evidently not relevant to ranking a specific range pattern (see line 8 of Algorithm 1). Figure 5 plots the percentage of remaining relationships after pruning (averaging accross all range patterns) with γ ranging from 0 to 0.9. This experiment was conducted on Wikidata, DBpedia, and YAGO. As expected, the curves generally show a downward slope since increasing the threshold restricts the condition for the relationships to continue in the algorithm. Interestingly, at the threshold of a uniform distribution, which equals 1/3 (see Property 3), 45%, 50%, and 42.5% of relationships are pruned on average for Wikidata, DBpedia, and YAGO, respectively. This drastic reduction of the search space shows the interest of this optimization.

6.2 Evaluation of the Gini coefficient approximation

In this section, we check whether our Gini approximation is sufficiently accurate in practice, whether it is faster, and whether it contributes to reducing the number of queries needed. To evaluate our approximation of the Gini coefficient, we focused on the rankings that concern at least 500 entities (see column "# filtered

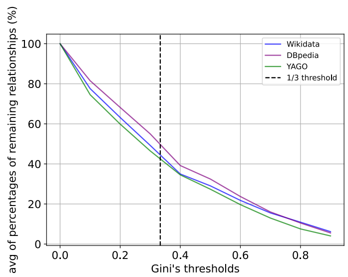

Figure 5: Proportion of relevant relationships with γ

Table 5: Gini evaluation

KG	Type	MAE	MSE	Time-Gain (%)	Queries-Avoided (%)	Avg. response time (sec.)
Wikidata	occ.	0.062	0.005	90.3	34.4	0.100
	classes	0.078	0.012	88.5	36.7	0.426
	all	0.068	0.008	88.9	36.1	0.404
DBpedia	occ.	0.137	0.031	97.3	27.0	0.041
	classes	0.072	0.012	99.2	22.0	0.060
	all	0.077	0.014	99.1	22.5	0.053
YAGO	occ.	0.069	0.007	78.9	35.3	0.041
	classes	0.064	0.009	63.9	28.4	0.093
	all	0.068	0.008	74.3	32.4	0.074

RI" in Table 4). For example, 1,576 rankings verify this for Wikidata occupations.

First, for each filtered ranking indicator, we compare the differences of Gini coefficients in their values. To measure the error between the two distributions, we utilized two common statistical metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). Here, MAE computes the average absolute difference between the Gini coefficient and our approximation: MAE = $\frac{1}{n}\sum_{i=1}^{n}|Gini_i - \widetilde{Gini}_i|$ where n represents the number of ranking indicators. Similar to MAE, MSE calculates the average squared difference: MSE = $\frac{1}{n}\sum_{i=1}^{n}(Gini_i - \widetilde{Gini}_i)^2$. However, it places greater focus on larger errors due to the squaring operation leading to greater penalization of larger errors. Overall, we can see that the values are very low for Wikidata, DBpedia, and YAGO for both MAE (with 0.068, 0.077, and 0.068, respectively) and MSE (with 0.008, 0.014, and 0.008, respectively), since the range of the Gini coefficient is between 0 and 1. Table 5 shows these measures also after separating occupations and classes. Furthermore, Figure 6-a, b, and c plot the ranking indicators for Wikidata, DBpedia, and YAGO respectively, considering $\widetilde{Gini}$ as the x-axis and $Gini$ as the y-axis. Additionally, we plot the identity line, indicating where the Gini coefficient and its approximation are equal. We see that the behavior is the same for occupations (in blue) and classes (in red). Given the low MAE and MSE, it is no surprise that the points generally follow the identity line. For some points (below the identity), the approximation is disappointing (especially for DBpedia), as the underlying model proposed in [2] has an average accuracy of 72.5%.

We now evaluate the runtime performance of the Gini coefficient approximation. For this purpose, we consider the time gain achieved by employing the $\widetilde{Gini}$: Time-Gain = $\frac{T-\tilde{T}}{T}$ where T (resp. $\tilde{T}$) is the average computation time using the $Gini$ (resp. $\widetilde{Gini}$). This measure calculates the percentage difference in computation time between the two methods. A positive value indicates a reduction in computation time (i.e., time gain) achieved by the $\widetilde{Gini}$ compared to the $Gini$. Interestingly, Figure 7-a, b, and c depicts points each representing the total number of seconds required to generate possible rankings for a specific range pattern for Wikidata, DBpedia, and YAGO respectively, where the computation times of $\widetilde{Gini}$ and $Gini$ are respectively on the abscissa axis and on the ordinate axis. It is clear that the $Gini$ requires significantly more time compared to $\widetilde{Gini}$ since most points are above the identity line. Consequently, Time-Gain of the approximation is 88.9% for Wikidata, 99.1% for DBpedia, and 74.3% for YAGO. For example, in extreme cases, for Wikidata the standard Gini requires 47.35 minutes to discover rankings for the Business class while our approximation only requires 0.5 minutes. Additionally, Table 5 presents the average time in seconds to obtain the first response, i.e., the time to return the first ranking indicator for a range pattern. The times are 0.404 seconds for Wikidata, 0.053 seconds for DBpedia, and 0.074 seconds for YAGO. These results indicate a very fast initial response, emphasizing the potential for developing an efficient, and automatically interactive, on-demand application. Lastly, we evaluate the optimization in the number of queries. To compute the percentage of queries avoided when using the approximation rather than the standard formula, we use the following formula: Queries-Avoided = $\frac{Q-\tilde{Q}}{Q}$, where Q (resp. $\tilde{Q}$) is the average needed query number using the $Gini$ (resp. $\widetilde{Gini}$). Overall, 36.1%, 22.5%, and 32.4% of queries needed using the standard formula are avoided by using the approximation for Wikidata, DBpedia, and YAGO, respectively. In conclusion, $\widetilde{Gini}$ enables us to design an on-demand application returning rankings due to the simultaneous minimization of the number and complexity of queries required for calculations.

6.3 Evaluation of the diversity

This section focuses on Wikidata to assess whether our approach introduces diversity or, on the contrary, tends to always produce the same ranking indicator and to reduce a set of entities to a single ranking (the results of DBpedia and YAGO are available in the supplementary materials). For the same range pattern, we first observe in Table 4 that the ranking indicators obtained are numerous: 212.4 ranking indicators on average for occupations and 35.2 on average for classes. We also note that the relationships used to count vary (globally and for the same range pattern), bringing diversity to the way entities are compared. Among the 818 relationships, the 3 most used in the rankings indicators are `part of` (P361) with 2,723 ranking indicators, `main subject` (P921) with 2,664 ranking indicators and `depicts` (P180) with 1,673 ranking indicators.

Now we want to measure the impact of these different patterns on the ranking of entities for the same range pattern. For this experiment, we selected the 8 occupations among the largest occupations in Wikidata involving ranking indicators with several relationships. For each occupation, Table 6 gives the number of extracted ranking indicators. We retained the 4 best ranking indicators w.r.t $Gini \times Prop$ while ensuring covering different relationships. For each ranking indicator, we then considered the first 500 entities of the induced ranking. Note that the below observations remain true

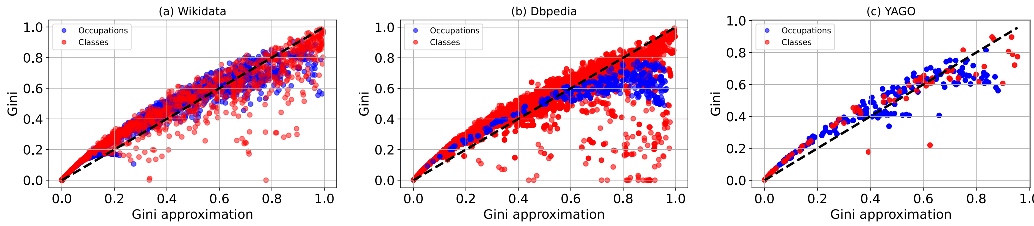

Figure 6: Comparison of the values between the Gini coefficient $Gini$ and its approximation $\widetilde{Gini}$

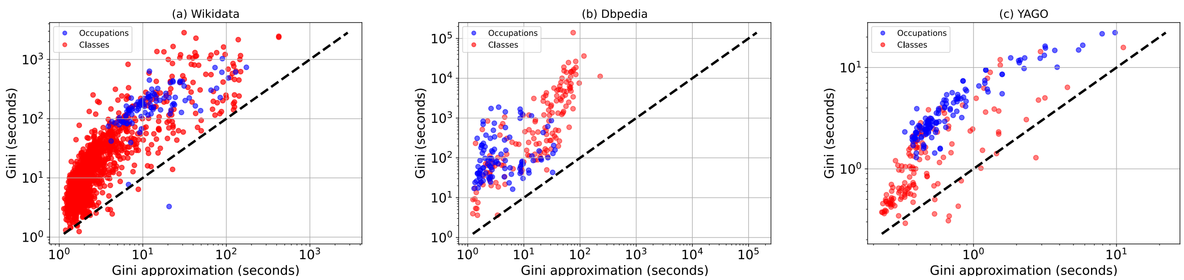

Figure 7: Comparison of the computation time needed between the Gini coefficient $Gini$ and its approximation $\widetilde{Gini}$

Table 6: Ranking diversity of RIPM and Popularity

		Pair with maximum inter.		All pairwise comparisons		Ranking by popularity	
Occupation	**#RIs**	**Inter. (%)**	τ	**Avg. inter.(%)**	**Avg.** τ	**Inter. (%)**	τ
Writers	519	30.8	0.332	6.7	0.030	0.306	0.313
Univ. teachers	390	13.8	0.019	3.7	0.094	0	-
Singers	291	63.8	0.635	15.9	-0.046	0.344	0.182
Journalists	380	27.2	0.246	8.8	-0.109	0.306	0.358
Poets	341	29.4	0.094	13.0	0.051	0.356	0.390
Actors	388	29.4	0.180	7.5	-0.168	0.166	0.255
Painters	389	19.2	0.129	5.2	0.115	0.604	0.516
Composers	345	15.4	0.055	7.9	0.049	0.344	0.180

when thresholds 4 and 500 are varied. To measure the degree of agreement between two different rankings, we use Kendall's tau, denoted by τ (also known as Kendall's rank correlation coefficient). Kendall's tau quantifies the resemblance between two rankings by assessing the proportion of pairwise agreements and disagreements between the ranks. It is a non-parametric measure, meaning it does not assume any specific distribution of the data, and it is suitable for ordinal data. It can be computed using the following formula: $\tau = \frac{C-D}{C+D}$ where C is the number of concordant pairs, and D is the number of discordant pairs. Kendall's tau ranges from -1 (i.e., perfect disagreement) to 1 (i.e., perfect agreement). In practice, for a pair of rankings, we apply the Kendall's tau to the intersection (i.e., the common entities appearing in both rankings).

For each occupation, Table 6 presents the average intersection and Kendall's tau for all pairwise comparisons within the four selected ranking indicators and the results for the pair with the maximum intersection. We note that the ranking indicators do not cover the same entities in the top 500. The intersection between two rankings is always below 63.8%, and on average below 15.9%. This clearly shows the complementarity of the different rankings of the entities involved. For example, counting the number of films and the number of television series does not rank the same actors (with only 3.4% as an intersection). We also observe that the entities common to two rankings are ranked in distinct orders. Indeed, even if Kendall's tau are mostly positive, they are always below 0.635 and on average below 0.115. There are even 3 occupations where the values are negative, indicating strong divergences. For example, the actors who have written the most screenplays are not the ones who have acted in the most films leading to -0.091 as Kendall's tau between these two rankings. Once again, these results demonstrate the diversity of the extracted ranking indicators.

Table 6 also presents results regarding the ranking by popularity baseline. For each occupation, we compare the ranking of Popularity with that induced by the best ranking indicator identified by RIPM (highlighted in bold in Table 7). We retain the top 500 entities from each ranking and compute the intersection percentage and Kendall's tau of common entities (see the last two columns). This results in an average entity intersection percentage of 0.303% and an average Kendall's tau of 0.313, indicating low overlap and low agreement respectively. This result shows that confusing all facts leads to very different rankings from those obtained with our indicators based on precise semantics.

Additionally, Table 7 (resp. Table 8) provides the three best ranking indicators (w.r.t $\widetilde{Gini} \times Prop$) discovered per occupation (resp. class). These two tables show the specificity and the understandability of the ranking indicators produced by the method for each range pattern (occupation or class). In particular, our method automatically discovers the ranking indicators used in scientometrics. For example, on one hand, it indicates that to rank a university teacher, we can count the number of times they were a doctoral advisor of a PhD student, the number of their scholarly articles,

Table 7: Best discovered ranking indicators for occupations

Occupation	Property	Class	Gini	Prop.
Writer	**author**	**version, edition or translation**	**0.683**	**0.101**
	author	literary work	0.646	0.094
	main subject	encyclopedia article	0.338	0.051
Univ. teacher	**doctoral advisor**	**human**	**0.949**	**0.159**
	author	scholarly article	0.688	0.085
	author	review article	0.768	0.034
Singer	**performer**	**album**	**0.698**	**0.145**
	performer	single	0.698	0.083
	performer	musical work/composition	0.637	0.071
Journalist	**author**	**version, edition or translation**	**0.723**	**0.058**
	author	literary work	0.593	0.048
	screenwriter	film	0.540	0.023
Poet	**author**	**version, edition or translation**	**0.797**	**0.082**
	author	literary work	0.797	0.075
	author	poem	0.936	0.026
Actor	**cast member**	**film**	**0.755**	**0.370**
	cast member	television series	0.464	0.166
	cast member	television film	0.395	0.068
Painter	**creator**	**painting**	**0.829**	**0.308**
	creator	print	0.863	0.027
	creator	drawing	0.861	0.024
Composer	**composer**	**musical work/composition**	**0.792**	**0.094**
	composer	film	0.786	0.090
	producer	album	0.759	0.040

Table 8: Best discovered ranking indicators for classes

Classes	Property	Class	Gini	Prop.
Position	**position held**	**human**	**0.825**	**0.790**
	subclass of	position	0.951	0.011
	office contested	public election	0.884	0.003
Scientific journal	**cites work**	**scholarly article**	**0.930**	**0.071**
	main subject	scholarly article	0.981	0.005
	has written for	human	0.714	0.002
Business	**employer**	**human**	**0.589**	**0.073**
	owned by	United States patent	0.942	0.028
	owned by	business	0.284	0.023
River	**mouth of the watercourse**	**river**	**0.564**	**0.053**
	mouth of the watercourse	stream	0.551	0.008
	crosses	road bridge	0.464	0.006
Association football clubs	**member of sports team**	**human**	**0.954**	**0.462**
	participating team	sports season	0.755	0.112
	winner	sports season	0.596	0.126
Sports season	**participant in**	**human**	**0.857**	**0.040**
	participant in	national sports team	0.654	0.007
	participant in	association football club	0.738	0.006
Musical group	**performer**	**album**	**0.621**	**0.274**
	performer	single	0.667	0.076
	performer	musical work/compo.	0.560	0.045
University	**educated at**	**human**	**0.983**	**0.557**
	employer	human	0.971	0.545
	part of	academic department	0.678	0.068

or their review articles. On the other hand, to rank a university, we can count the number of humans educated there, the number of humans they employ, or the number of their academic departments which indicates the size of the university, also the number of academic journals they published, which provides insight into their scientific quality has been considered as a ranking indicator (not among the top three). Of course, other ranking indicators are also discovered, 390 (resp. 190) ranking indicators are discovered for university teachers (resp. university), but here we mention only the best three. In some cases, certain indicators may be too general (e.g., the best ranking for poets is based on the number of editions rather than the number of poems) or too specific (e.g., the number of "association football club" participation to rank "sports season"). For this reason, it is important to offer users several rankings, so that they can choose the one they prefer.

Table 9: Agreement of interestingness on the user survey and baseline comparison

Occupation	Users τ	Users-RIPM τ	ChatGPT-RIPM τ	RIPM		Popularity		ChatGPT4	
				Gini	Prop.	Gini	Prop.	Gini	Prop.
Writers*	0.610	1	1	0.683	0.101	0.780	0.807	0.347	0.01
Univ. teachers	0.454	1	1	0.949	0.159	0.822	0.728	0.785	0.161
Singers	0.236	1	1	0.698	0.145	0.759	0.903	0.606	0.144
Journalists	0.252	1	1	0.723	0.058	0.749	0.861	0.803	0.009
Poets*	0.555	1	0.333	0.797	0.082	0.816	0.830	0.868	0.026
Actors*	0.587	1	1	0.755	0.370	0.719	0.808	0.655	0.374
Painters*	0.360	1	1	0.829	0.308	0.792	0.806	0.753	0.307
Composers*	0.454	0.333	1	0.792	0.094	0.773	0.845	0.639	0.071
Average	0.438	0.917	0.917	0.778	0.165	0.776	0.824	0.682	0.138

6.4 Interestingness evaluation based on a user survey

There is no method for extracting ranking indicators automatically (see Section 7). Due to this absence, we chose to conduct a user survey using a questionnaire on the occupations selected in the previous section, which are fairly well-recognized and then, understandable. For each occupation, we presented three ranking indicators expressed in natural language corresponding to 3 levels of interest (best/medium/bad) w.r.t our approach. We order the ranking indicators according to the measure $Gini \times Prop$. Then, the *best* ranking indicator is selected (see Table 7), the *medium* one is drawn from the 45th to the 55th percentile of the ordered list and the *bad* one is drawn from the last 10% of the ordered list. For instance, when ranking a painter, our protocol considers factors such as the number of paintings they have created (best ranking), the number of prints derived from their work (medium ranking), or the number of art museums named after them (bad ranking penalized by its low cover of entities). We selected a group of 19 raters of current university students or graduates in the field of computer science. Next, each rater ordered the three ranking indicators associated with the specific occupation, based on their opinion about their relevance for ranking entities having the occupation.

As shown in Table 9, firstly, we calculated Kendall's tau for the rankings provided by each pair of raters for each occupation. We then computed the average τ per occupation, which ranged from 0.236 to 0.610, with a total average of 0.438 across occupations, indicating moderate agreement [23]. We also identified the ground truth from the user ratings using majority voting. In other words, for each occupation, we selected the ranking that occurred most frequently among the users. Note that (*) in Table 9 means that the majority voting is statistically significant. More precisely, the first choice of the population would be the same as that of our study with a probability greater than 99% based on Bennett's inequality [24]. Besides, the last choice is significantly the worst for the whole population, with a probability greater than 85%. For three occupations (university teacher, singer and journalist), the first choice is not completely certain, but increasing the sample size would not be enough to remove the measured intrinsic disagreement between first and second choice. For example, the choice for university teachers is subjective: 11 (resp. 6) raters considered that the number of doctoral students (resp. PhD defense committee member) was more important. We computed the τ between this ground truth and the results of RIPM (see the third column). Interestingly, we observe that our approach has a perfect agreement with the ground truth (i.e., $\tau = 1$), except for composers ($\tau = 0.333$). The average τ across the occupations is very high with 0.916. This implies agreement between the results of our method and the subjective opinion of humans, indicating the efficacy of our approach in discovering relevant rankings. Finally, we used ChatGPT4 to evaluate the 3 ranking indicators. It agrees with the ground truth, with an accuracy comparable to RIPM. This indicates the meaningfulness of the $Gini \times Prop$ measure w.r.t ChatGPT4.

6.5 Comparison with baselines

We queried ChatGPT4 to find 3 ranking indicators directly for the 8 occupations, providing it with the form of the expected pattern to guide it. 12 out of the 24 results provided were hallucinations, with erroneous classes and relationships that made it impossible to construct the ranking. For example, for journalists, it proposed the relationship “affiliation” with the class “media compagny” (Q378427), but this Wikidata identifier corresponds to “literary Awards”. Table 9 shows, for each occupation, the Gini and Proportion of the best ranking indicator generated by RIPM, by Popularity, and by ChatGPT4 (after correcting its hallucinations except in 2 cases where the proposed class did not exist in Wikidata). First, RIPM has a Gini coefficient as good as that of Popularity. The latter has a higher proportion value than RIPM because it mixes all ranking criteria together, resulting in broader coverage of more entities. But, as already illustrated with painters, some criteria may not be relevant for ranking the specific domain leading to incorrect ranking of entities. Second, we observe that RIPM outperforms ChatGPT4 with a Gini of 0.778 compared to 0.682. Regarding the proportion, RIPM (= 0.165) also surpasses ChatGPT4 (= 0.138), indicating that the quality of indicators discovered by RIPM is better than those generated by ChatGPT4 (even after correcting its errors).

7 RELATED WORK

The semantics introduced by knowledge graphs deserve to be more directly usable by humans. A great deal of work has been done to facilitate the exploration and analysis of knowledge graphs. Our contribution clearly differs from these in its focus on the ranking task. However, we can cite some approaches that are comparable to ours in their reliance on specific graph patterns. For example, in [15] the authors define the notion of facet as a property path that can serve as a filter for graph exploration. They propose an algorithm for selecting the most relevant facets for a given set of entities. In a similar spirit, authors of [48] define graph patterns to measure how important, or central, nodes are in a graph, with the aim of generating snippets from keywords. As the objectives are different, the graph patterns we define are obviously different too: they are intended to compute ranking indicators.

Our approach to discovering ranking indicators in knowledge graphs is inspired by various proposals from the science of measurements in the field of information (informetrics). Pioneers are in the field of bibliometrics [35, 37], providing measures that are often used in a reductive way for evaluation purposes [17]. Furthermore, the rise of altmetrics offers a new dimension of impact measurement, capturing the broader societal and online engagement with scholarly work [7, 44]. Yet these measures can also be used to assess the importance of certain scientific results and the influence of certain authors, for an epistemological or sociological analysis of sciences. Compared with these approaches, our contribution has a much broader scope, since it does not apply solely to information specifically related to science, along with the automatic discovering and extracting of measures.

As several authors have shown [41, 43, 45], bibliometrics has adapted to the Web, evolving into Webometrics. Webometrics is presented more generally in [45] as the quantitative study of Web-related phenomena, initiated at the end of the previous century, and focusing on the analysis of web page links or search engine logs. More generally, interest in ranking methods and tools is particularly strong when the objective is information retrieval [21, 30], and many measures have been devised in this field including [10–13]. However, in line with the baseline Popularity, these proposals only consider graphs whose links are all identical and carry no semantics other than the fact of existing or not (with rare exceptions, e.g. [40]). In contrast, our contribution consists of evaluating the different types of relationships to determine which ones should be considered for a ranking task, and we show how diverse the choices are to perform a ranking task. Finally, while existing works on rankings (e.g., [8, 16, 34]) study *predefined* ranking functions, Algorithm RIPM automatically discovers fresh, transparent, diverse, and understandable ranking functions from knowledge graphs.

8 CONCLUSION

This paper presents the first method for automatically discovering ranking indicators from knowledge graphs. We have formalized the notion of ranking indicator, which is a counting graph pattern that covers the entities to be ranked as widely as possible, inducing a distribution with high inequality. We present an efficient algorithm for extracting ranking indicators from a public SPARQL endpoint using a Gini approximation as inequality measure and several prunings of the search space. Extensive experiments show the algorithm’s speed, the diversity of the ranking indicators extracted, and their high relevance to the target domain in agreement with human raters. The ability to extract transparent, diverse, understandable ranking indicators on demand paves the way for the analysis of domains that do not already have indicators [49].

In future work, we would like to explore combinations of ranking indicators in order to build finer ones to go beyond the countings proposed in this paper. More precisely, it would be possible to construct indicators comparable to the h-index used in bibliometrics [19] (which combines both the number of citations and the number of papers). For example, for painters, it is possible to combine the number of paintings with the number of items referring to each of these paintings. Furthermore, multicriteria analysis [1, 46] can be performed by combining and weighting ranking indicators. For instance, in the case of painters, a combination of the number of paintings and the number of art exhibitions can be used. Finally, we also plan to implement these indicators in an online tool to enable any knowledge worker to interactively analyze any field of Wikidata.

ACKNOWLEDGMENTS

We thank Louise Parkin who recently joined our team for her rigorous proofreading and numerous suggestions for improvement.

REFERENCES

[1] Ash Mohammad Abbas. 2011. Weighted indices for evaluating the quality of research with multiple authorship. *Scientometrics* 88, 1 (2011), 107–131.

[2] Hassan Abdallah, Béatrice Markhoff, and Arnaud Soulet. 2024. A Complex Network Model for Knowledge Graphs' Relationships. *Preprint.* (2024).

[3] Anthony B Atkinson et al. 1970. On the measurement of inequality. *Journal of economic theory* 2, 3 (1970), 244–263.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *international semantic web conference.* Springer, 722–735.

[5] Robert D Behn. 2003. Why measure performance? Different purposes require different measures. *Public administration review* 63, 5 (2003), 586–606.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american* 284, 5 (2001), 34–43.

[7] Lutz Bornmann. 2014. Do altmetrics point to the broader impact of research? An overview of benefits and disadvantages of altmetrics. *Journal of informetrics* 8, 4 (2014), 895–903.

[8] Zixuan Chen, Panagiotis Manolios, and Mirek Riedewald. 2023. Why not yet: Fixing a top-k ranking that is not fair to individuals. *Proceedings of the VLDB Endowment* 16, 9 (2023), 2377–2390.

[9] Frank Alan Cowell. 2011. *Measuring inequality.* Oxford University Press.

[10] Renaud Delbru, Stephane Campinas, and Giovanni Tummarello. 2012. Searching web data: An entity retrieval and high-performance indexing model. *Journal of Web Semantics* 10 (2012), 33–58.

[11] Renaud Delbru, Nickolai Toupikov, Michele Catasta, Giovanni Tummarello, and Stefan Decker. 2010. Hierarchical link analysis for ranking web data. In *Extended Semantic Web Conference.* Springer, 225–239.

[12] Abdelmoneim Amer Desouki, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. Ranking on very large knowledge graphs. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media.* 163–171.

[13] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. 2004. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management.* 652–659.

[14] Martin Dürst and Michel Suignard. 2005. *Internationalized resource identifiers (IRIs).* Technical Report.

[15] Leila Feddoul, Sirko Schindler, and Frank Löffler. 2019. Automatic Facet Generation and Selection over Knowledge Graphs. In *International Conference on Semantic Systems.* Springer, 310–325.

[16] Abraham Gale and Amélie Marian. 2020. Explaining monotonic ranking functions. *Proceedings of the VLDB Endowment* 14, 4 (2020), 640–652.

[17] Eugene Garfield. 1998. From citation indexes to informetrics: Is the tail now wagging the dog? *Libri* 48, 2 (1998), 67–80.

[18] Corrado Gini. 1936. On the measure of concentration with special reference to income and statistics. *Colorado College Publication, General Series* 208 (1936), 73–79.

[19] Jorge E Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences* 102, 46 (2005), 16569–16572.

[20] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)* 54, 4 (2021), 1–37.

[21] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.

[22] Egor V Kostylev, Juan L Reutter, Miguel Romero, and Domagoj Vrgoč. 2015. SPARQL with property paths. In *The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I 14.* Springer, 3–18.

[23] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[24] Andreas Maurer and Massimiliano Pontil. 2009. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740* (2009).

[25] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckeschmidt. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal* 33, 1 (2024), 131–161.

[26] Thomas Minier, Hala Skaf-Molli, and Pascal Molli. 2019. SaGe: Web preemption for public SPARQL query services. In *The World Wide Web Conference.* 1268–1278.

[27] Jerry Z Muller. 2018. *The tyranny of metrics.* Princeton University Press.

[28] Mark EJ Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary physics* 46, 5 (2005), 323–351.

[29] OpenAI. 2024. *ChatGPT4 (Aug 8 version).* https://chat.openai.com Large language model.

[30] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[31] M Pavan and R Todeschini. 2008. Total-order ranking methods. *Data handling in Science and Technology* 27 (2008), 51–72.

[32] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. YAGO4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17.* Springer, 583–596.

[33] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2009. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)* 34, 3 (2009), 1–45.

[34] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2022. Fairness in rankings and recommendations: an overview. *The VLDB Journal* (2022), 1–28.

[35] Allan D Pratt. 1977. A measure of class concentration in bibliometrics. *Journal of the American Society for information Science* 28, 5 (1977), 285–292.

[36] Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, and Fabian Suchanek. 2024. Completeness, Recall, and Negation in Open-World Knowledge Bases: A Survey. *Comput. Surveys* 56, 6 (2024), 1–42.

[37] Ronald Rousseau. 1994. Similarities between informetrics and econometrics. *Scientometrics* 30, 2 (1994), 385–388.

[38] Anthony F Shorrocks. 1980. The class of additively decomposable inequality measures. *Econometrica: Journal of the Econometric Society* (1980), 613–625.

[39] Arnaud Soulet, Arnaud Giacometti, Béatrice Markhoff, and Fabian M Suchanek. 2018. Representativeness of knowledge bases with the generalized Benford's law. In *International Semantic Web Conference.* Springer, 374–390.

[40] David Stuart. 2012. FOAF Within UK Academic Web Space: A Webometric Analysis of the Semantic Web'. *Social Information Research (Library and Information Science, Volume 5). Emerald Group Publishing Limited* (2012), 173–191.

[41] David Stuart. 2014. *Web metrics for library and information professionals.* Facet Publishing.

[42] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web.* 697–706.

[43] Mike Thelwall. 2008. Bibliometrics to webometrics. *Journal of information science* 34, 4 (2008), 605–621.

[44] Mike Thelwall, Stefanie Haustein, Vincent Larivière, and Cassidy R Sugimoto. 2013. Do altmetrics work? Twitter and ten other social web services. *PloS one* 8, 5 (2013), e64841.

[45] Mike Thelwall, Liwen Vaughan, and Lennart Björneborn. 2005. Webometrics. *Annual review of information science and technology* 39, 1 (2005), 81–135.

[46] Evangelos Triantaphyllou and Evangelos Triantaphyllou. 2000. *Multi-criteria decision making methods.* Springer.

[47] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[48] Xiaxia Wang, Jinchi Chen, Shuxin Li, Gong Cheng, Jeff Z Pan, Evgeny Kharlamov, and Yuzhong Qu. 2019. A framework for evaluating snippet generation for dataset search. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18.* Springer, 680–697.

[49] Gerhard Weikum. 2021. Knowledge graphs 2021: A data odyssey. *Proceedings of the VLDB Endowment* 14, 12 (2021), 3233–3238.

[50] Alvin M Weinberg. 2000. Criteria for Scientific Choice (Minerva, I (2),(1962), 158–171). *Minerva* 38, 3 (2000), 253–266.