A Unified Framework for Rank-based Evaluation Metrics for Link Prediction in Knowledge Graphs

Charles Tapley Hoyt
Harvard Medical School
Boston, MA, USA
cthoyt@gmail.com

Max Berrendorf
LMU Munich
München, Germany
max.berrendorf@gmail.com

Mikhail Galkin
Mila & McGill University
Montreal, QC, Canada
mikhail.galkin@mila.quebec

Volker Tresp
LMU Munich & Siemens AG
München, Germany
volker.tresp@siemens.com

Benjamin M. Gyori
Harvard Medical School
Boston, MA, USA
benjamin_gyori@hms.harvard.edu

ABSTRACT
The link prediction task on knowledge graphs without explicit negative triples in the training data motivates the usage of rank-based metrics. Here, we review existing rank-based metrics and propose desiderata for improved metrics to address lack of interpretability and comparability of existing metrics to datasets of different sizes and properties. We introduce a simple theoretical framework for rank-based metrics upon which we investigate two avenues for improvements to existing metrics via alternative aggregation functions and concepts from probability theory. We finally propose several new rank-based metrics that are more easily interpreted and compared accompanied by a demonstration of their usage in a benchmarking of knowledge graph embedding models.

CCS CONCEPTS
• Computing methodologies → Ranking.

KEYWORDS
graph machine learning, metrics, ranking

ACM Reference Format:
Charles Tapley Hoyt, Max Berrendorf, Mikhail Galkin, Volker Tresp, and Benjamin M. Gyori. 2022. A Unified Framework for Rank-based Evaluation Metrics for Link Prediction in Knowledge Graphs. In Proceedings of TheWebConf Workshop on Graph Learning Benchmarks 2022 (WWW22). ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXX

1 INTRODUCTION
Knowledge graphs (KGs) are a structured formalism for representing facts about entities \( E \) and their relationships \( R \) as triples of the form \((h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\). KGs are useful for entity clustering, link prediction, entity disambiguation, question answering, dialogue systems, and recommendation systems [33]. They can be constructed under one of two assumptions: under the closed-world assumption (CWA), the non-existence of a triple in the KG implies its falseness and under the open-world assumption (OWA), the non-existence of a triple in the KG neither implies its falseness nor truthiness. Most real-world KGs are constructed under the OWA to reflect their relative incompleteness with respect to true triples and typical lack of triples known to be false.

Link prediction on KGs constructed under the OWA is a popular approach for addressing their relative incompleteness that can be conceptualized as a binary classification task on triples. However, the lack of false triples leads to a positive unlabeled learning scenario [6] which requires negative sampling i.e., randomly labeling some unknown triples as false during the training and evaluation of machine learning models like knowledge graph embedding models (KGEMs). Partially because these techniques introduce bias to classification metrics whose formulation depends on true negatives and false negatives (such as the accuracy, \( F_1 \), and ROC-AUC), the last ten years of KGEM literature has nearly exclusively used the rank-based evaluation metrics: hits at \( k \) (\( H_k \)), mean rank (MR), and mean reciprocal rank (MRR).

Despite their ubiquity, these metrics are not comparable when applied to datasets of different sizes and properties and lack a corresponding theoretical framework for describing their properties and shortcomings. For example, there is recent interest in applying link prediction with KGEMs to real-world tasks in biomedicine such as drug repositioning, target identification, and side effect prediction. However, there are several choices of biomedical KGs for training and evaluation, each formulated with different entities, relations, and source databases [8]. Because the choice of KG can meaningfully impact downstream applications, hyperparameter search and evaluation must also consider and compare different KGs, which, without rank-based evaluation metrics that are comparable across datasets nor theory to help articulate the issue, is not possible.

This work addresses these limitations by making the following contributions: (1) Proposes a theoretical foundation for rank-based evaluation metrics; (2) Proposes and characterizes novel rank-based evaluation metrics with alternative rank transformations and alternative aggregation operations based on special cases of the generalized power mean [10]; (3) Derives probabilistic adjustments for existing and novel rank-based evaluation metrics inspired by [7, 30]. We implemented the proposed novel metrics and made them available as part of the PyKEEN [2] software package.1

1https://github.com/pykeen/pykeen
2 BACKGROUND

2.1 Generating Ranks

Given a set of entities \( \mathcal{E} \), a set of relations \( \mathcal{R} \), a knowledge graph \( \mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \), disjoint testing and validation triples \( \mathcal{T}_{test}, \mathcal{T}_{val} \subseteq \mathcal{K} \), and a KGEM with scoring function \( g : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R} \) (e.g., TransE [9] has \( g(h, r, t) = -\|e_h + r_e - e_t\|_2 \)), evaluation concurrently solves two tasks as described by [22]:

1. In right-side prediction, for each triple \((h, r, t) \in \mathcal{T}_{val}\), we score candidate triples \(\{(h, r, e) \mid e \in \mathcal{E}\}\) using \(g\).
2. In left-side prediction, for each triple \((e, r, t) \in \mathcal{T}_{val}\), we score candidate triples \(\{(e, r, t) \mid e \in \mathcal{E}\}\) using \(g\).

In each task, candidate triples are sorted in descending order based on their scores, and are then assigned a rank based on their 1-indexed sort position. In the (optional) filtered setting proposed by [9], candidate triples appearing in \(\mathcal{T}_{test}\) are removed from the ordering so they do not artificially increase the ranks of true triples \(\mathcal{T}_{val}\) appearing later in the sorted list. A good model results in high scores to true triples and low scores to negatively sampled triples. The ranks are typically aggregated using a rank-based evaluation metric to quantify the performance of the KGEM with a single number.

While the upper bound on an individual rank \(r_i\) is generally \(|\mathcal{E}|\) for both the right- and left-side prediction tasks, the number of candidates may be (considerably) smaller than \(|\mathcal{E}|\), e.g., in the filtered setting [9] or during sampled evaluation [17, 29].

2.2 Aggregating Ranks

While the distribution of raw ranks gives full insight into evaluation performance, it is much more convenient to report aggregate statistics. In this subsection, we introduce and describe three common rank-based metrics reported in applications and evaluation of link prediction: hits at \(k\), mean rank, and mean reciprocal rank.

**Hits at \(k\)**. The hits at \(k\) (\(H_k\)) (Equation 1) is an increasing metric (i.e., higher values are better) that captures the fraction of true entities that appear in the first \(k\) entities of the sorted rank list. Thus, it is tailored towards a use-case where only the top-\(k\) entries are to be considered, e.g., due to a limited number of results shown on a search result page. Because it does not differentiate between cases when the rank is larger than \(k\), a miss with rank \(k + 1\) and \(k + d\) where \(d \gg 1\) have the same effect on the final score. Therefore, it is less suitable for the comparison of different models.

\[
H_k(r_1, \ldots, r_n) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[r_i \leq k] \quad \in [0,1]
\]

where the indicator function \(\mathbb{I}\) is defined as \(\mathbb{I}[x \leq y] = 1\) for \(x \leq y\) and 0 otherwise.

**Mean rank**. The mean rank (MR) (Equation 2) is a decreasing metric (i.e., lower values are better) that corresponds to the arithmetic mean over ranks of true triples. It has the advantage over \(H_k\) that it is non-parametric and better reflects average performance.

\[
MR(r_1, \ldots, r_n) = \frac{1}{n} \sum_{i=1}^{n} r_i \quad \in [1, \infty)
\]

**Mean reciprocal rank**. The mean reciprocal rank (MRR) (Equation 3) is an increasing metric that corresponds to the arithmetic mean over the reciprocals of ranks of true triples. The construction of the MRR biases it towards changes in low ranks without completely disregarding high ranks like the \(H_k\). It can therefore be considered as a soft version of \(H_k\) that is less sensitive to outliers and is often used during early stopping due to this behavior. While it has been argued that MRR has theoretical flaws [13], these arguments are not undisputed [25].

\[
MRR(r_1, \ldots, r_n) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{r_i} \quad \in (0,1)
\]

2.3 Desiderata

After examining the strengths and weaknesses of the three most common rank-based metrics, we outline five desiderata for rank-based metrics that are interpretable and comparable across models/datasets/evaluation tasks in Table 1. Because each of MR, MRR, and \(H_k\) can be defined with the strictly monotonic increasing arithmetic mean as the aggregation function, we describe our desiderata with respect to the transformation \(f\) applied to \(r_i\) within the aggregation (further explored in subsection 3.1).

We first propose the intuitive constraint that the co-domain of the metric is non-negative, which is satisfied by all three metrics. We next propose that the best rank should result in a fixed optimum (ideally, 1) and that worse ranks should asymptotically approach a fixed pessimum (ideally, 0 for unbalanced metrics and -1 for balanced metrics), which are both satisfied by both MRR and \(H_k\) but neither by MR.

We propose that along this gradient, the metric should be strictly anti-monotonic, meaning that, as the performance of the KGEM improves, the predictions for true triples should improve, result in lower ranks \(r_i\), increased values of the transformed ranks \(f(r_i)\), and increased evaluation metrics. This is only satisfied by MRR as MR is an increasing function (i.e., higher ranks result in higher scores) and \(f(r_i)\) for \(H_k\) has only two discrete values (0 and 1), and thus is not strict in terms of monotonicity.

Because the worst rank is bounded based on the number of candidate triples (which itself depends on the dataset and the evaluation procedure) rather than a constant, the same MR from the evaluation on two different datasets of different sizes should not

### Table 1: Desiderata for rank-based metrics

| Property             | Constraint | MR | MRR | \(H_k\) |
|----------------------|------------|----|-----|---------|
| Non-negativity       | \(\forall r \in \mathbb{N} : f(r) \geq 0\) | ✔  | ✔   | ✔       |
| Fixed optimum        | \(f(1) = \text{opt.}\) | ✘  | ✔   | ✔       |
| Asym. pessimum       | \(\lim_{r \to \infty} f(r) = 0\) | ✘  | ✔   | ✘       |
| Anti-monotonic       | \(r > r' \implies f(r) < f(r')\) | ✘  | ✔   | ✘       |
| Size invariant       | \(\exists f \neq n\) | ✘  | ✘   | ✘       |

\[\]
be directly compared. While MRR and H_k are bounded with a constant, the shape of these curves are affected by the same properties of the number of candidate triples and the same issue is applicable. This situation makes the interpretation of results from even large robustness and ablation studies (whose aims are to identify patterns in interaction models, loss functions, regularizations, and other properties of KGE) challenging at best and misleading at worst. Therefore, our final desideratum is that rank-based metrics should be invariant to the number of candidate triples and directly comparable across different datasets and evaluation procedures.

3 AN ANALYSIS OF RANK-BASED METRICS

A general form of a rank-based metric $b$ over ranks $r_1, \ldots, r_n \in \mathbb{N}$ is $b(r_1, \ldots, r_n) = g(f) \in \mathbb{R}$ where $f : \mathbb{N} \mapsto \mathbb{R}$ is a rank transformation function, $\oplus : \mathbb{R}^n \mapsto \mathbb{R}$ is an aggregation operation, and $g : \mathbb{R} \mapsto \mathbb{R}$ is a post-aggregation transformation function.

3.1 Insight and Discovery via Transformations

The MR, MRR, and H_k metrics use the arithmetic mean as an aggregation function with $g(k) = \frac{1}{n} \sum_{i=1}^{n} f(r_i)$ and varying definitions for rank transformation function $f$ and post-aggregation function $g$ as summarized in Table 2. Based on this formulation, we propose that in addition to the colloquial formulation of MRR, we propose that in addition to the colloquial formulation of MRR, $\frac{1}{n} \sum_{i=1}^{n} f(r_i)$, we present alternate aggregations in Table 3. The max rank and min rank would not

4 PROBABILISTIC ADJUSTMENTS

Inspired by the probabilistic adjustments to MR that resulted in the adjusted mean rank (AMR) and the adjusted mean rank index (AMRI) [7], we considered generalizing their derivations and applying them to MRR and H_k. Similar to [7], we assume the ranking tasks $i$ to be independent, and the ranks uniformly discretely distributed over $[1, \ldots, N_i]$, such that $r_i \sim \mathcal{U}(1, N_i)$. Note that $N_i$ may not be constant across ranking tasks $i$ due to filtered evaluation [9].

4.1 Adjustments

**Expectation Adjustment.** The derivation of the AMR motivated normalizing a base metric $b$ by its expectation such that $\frac{b(r_1, \ldots, r_n)}{\mathbb{E}[b]}$. We found that it was only useful for metrics bounded by $[1, \infty]$ (i.e., MR, HMR, GM) whose adjustments were bounded by $[0, 1]$ and for metrics bounded by $[0, 1]$ (i.e., IMR, MRR, IGMR) whose adjustments were unbounded on $(0, \infty)$. The expected value of the adjusted metric is thus 1. Because it is not generally applicable, we do not propose any new metrics using this adjustment.

**Adjusted Index.** The derivation of the AMRI motivated normalizing a base metric $b$ by its expectation then linearly transforming it such that the maximum value maps to 1, the expectation maps to 0, positive values can be considered good, and negative values can be considered bad such that $\frac{b(r_1, \ldots, r_n)}{\max[b]} \triangleq \mathbb{E}[b]$. We use this form to propose the adjusted hits at $k$ (AH_k) and adjusted mean reciprocal rank (AMRR).

Surprisingly, the derivation of AMRI from MR resulted in the same form $\frac{b(r_1, \ldots, r_n)}{\max[b]} \triangleq \mathbb{E}[b]$ for the AMR and the AH_k, despite their different monotonicities (i.e., increasing or decreasing) and co-domains. We note that these characteristics do result in different lower bound behavior, which for AMRI is a constant $-1$ and for the AMR and AH_k is a function of the expectation of the base metric $-\mathbb{E}[b]$. Related derivations can be found in Appendix B.

**z-Adjustment.** We finally propose a novel probabilistic adjustment enabled by the central limit theorem. Because MR, MRR, H_k, and other metrics $b$ are defined as the sum of random variables (despite their several transformations), they have asymptotic Gaussian characteristics. Therefore, we propose using the standardization technique $z = \frac{x - \mu}{\sigma}$ to define a z-scored metric $\frac{b(r_1, \ldots, r_n)}{\sqrt{\mathbb{V}[b]}} \triangleq \mathbb{E}[b]$. We apply this to the MR, MRR, and H_k to respectively define three new metrics: z-mean rank (ZMR), z-mean reciprocal rank (ZMRR), and z-hits at $k$ (ZH_k). Related derivations can be found in Appendix B.
We note that the z-scored metrics can be monotonically mapped with the cumulative distribution function of the standard normal distribution onto the interval $(0, 1)$ to fulfill the desiderata. However, we suggest that z-scores are adequately interpretable and comparable without transformation, as the number of standard deviations below or above of the expected value.

### 4.2 Discussion

Each of the three probabilistic adjustment strategies presented in this work are affine transformations of the base metric with scale and bias constants only dependent on the studied ranking task, but independent of the investigated predictions. Thus, they can be applied to the base metrics after computation of a pre-computed expectation and variance that are appropriate for the dataset, e.g., to make results from existing publications more comparable across datasets and splits. We provide a database of pre-computed expectations and variances for benchmark datasets included in PyKEEN [2] stratified by split (i.e., training, testing, validation), evaluation task (left-hand, right-hand, both), and metric on Zenodo [15].

### 4.3 Case Study

In order to demonstrate the improved interpretability and comparability of our newly proposed adjusted metrics and z-scored metrics, we re-evaluated four KGEMs (ComplEx [32], RotatE [27], TransE [9], TuckER [5]) on four datasets (WN18-RR [11], FB15k-237 [31], Nations, and Kinships [19]) of varying size (from 14 entities to 40k entities, see Appendix Table 4) reusing the optimal hyperparameters reported in [1].

Figure 1 presents a comparison between original metric MRR, its adjusted index AMRR, and its z-adjustment ZMRR. We first observe that the MRR displays an anti-correlation with size of each dataset that is not present for AMRR and ZMRR, disregarding the smallest dataset for which the numerical behavior of the adjustments is slightly erratic. While the original metric suggests that ComplEx performs similarly on WN18-RR (green) and Nations (blue), the adjusted metric shows that the difference is more remarkable. Conversely, the original metric suggests that TuckER performs better on Nations than WN18-RR, while the adjusted metric shows that the difference is more remarkable. Conversely, the original metric suggests that TuckER performs better on Nations than WN18-RR, while the adjusted metric shows that the difference is more remarkable.

Finally, the z-adjusted metric enables direct comparison between the results on different datasets while also giving insight into their significance by normalizing against the expectation and variance of the metric under random rankings. This adjustment reveals that the improved original metrics on the two smaller datasets (Kinships and Nations) were less significant than the results on the two larger datasets (WN18-RR and FB15k-237), despite achieving better unnormalized performance.

All configuration, trained models, results, and analysis presented in this case study are available at https://github.com/pykeen/ranking-metrics-manuscript and archived on Zenodo at [16].

### 5 CONCLUSION

In this article, we motivated and reviewed rank-based evaluation metrics for the link prediction task on KGs before proposing desiderata for metrics with improved interpretability and comparability. We developed a simple theoretical framework for describing rank-based evaluation metrics, investigated their probabilistic properties, and ultimately proposed several new metrics with desired properties based on alternate aggregation functions (i.e., HMR, GMR), alternate transformations (i.e., IMR, IGMR), and probabilistic adjustments (i.e., AMRR, AH$_k$, ZMR, ZMRR, and ZH$_k$). We provide implementations of these metrics in PyKEEN [2] v1.8.0 with closed form solutions for the expectation and variance of the base metrics when possible and numeric solutions for the rest. We leave the remaining derivations of closed forms for the metrics defined with more complicated functions (e.g., GMR) for future work, to enable generation of z-scored metrics for the remaining base metrics.

**Generalization.** While we restricted our description in this work to the evaluation of link prediction on KGs, the discussed approaches are directly applicable to other settings which use rank-based evaluation, e.g., the entity pair ranking protocol [34], entity alignment [12, 21, 28, 35], query embedding [3, 4, 14, 20, 23, 24], unirelational link prediction [18, 36], and relation detection [26].

**Future Work.** Existing evaluation frameworks commonly compute one rank value per evaluation triple and side, then aggregate the ranks. However, real-world KGs often contain hub entities that occur in many triples which may therefore dominate the evaluation. We intend to build on previous work [3, 26, 30], investigating the issue using our novel metrics in a deeper investigation of rank-based evaluation of the link prediction task on knowledge graphs.

### ACKNOWLEDGMENTS

Charles Tapley Hoyt and Benjamin M. Gyori were supported by the DARPA Young Faculty Award W911NF20102551. Max Berrendorf was supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. Mikhail Galkin was supported by the Samsung AI grant held at Mila.
REFERENCES

[1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2020. Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. arXiv:2006.13365 http://arxiv.org/abs/2006.13365

[2] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. Journal of Machine Learning Research 22, 82 (2021), 1–6. http://jmlr.org/papers/v22/20-825.html

[3] Dimitrios Alivianistos, Max Berrendorf, Michael Cochez, and Mikhail Galkin. 2022. Query Embedding on Hyper-Relational Knowledge Graphs. In International Conference on Learning Representations. https://openreview.net/forum?id=4Lw9RTW9w

[4] Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18). AAAI Press, 1811–1818.

[5] Matthias Fey, Jan Eric Lenssen, Christopher Morris, Jonathan Masci, and Nils M. Kriege. 2020. Deep Graph Matching Consensus. In 9th International Conference on Learning Representations, ICLR 2020, Addas Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. https://openreview.net/forum?id=HyvHJKkS

[6] Norbert Fuhr. 2018. Some Common Mistakes In IR Evaluation, And How They Rank-Based Evaluation of Entity Alignment or Link Prediction. In 33rd Annual Conference on Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, December 6-12, 2020, virtual, Hiroko Sato, Mirco Airoldi, Rajat Koner, and Volker Tresp. 2020. Improving Visual Relation Detection using Deep Maps. In 29th International Conference on Pattern Recognition, ICPAR 2020. Virtual Event / Milan, Italy. January 10-15, 2021. IEEE, 3597–3604. https://doi.org/10.1109/ICCPAR.2020.9412945

[7] Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge Graph Completion via Complex Tensor Factorization. J. Mach. Learn. Res. 18 (2017), 130:1–130:38. https://jmlr.org/papers/v18/17-516.html

[8] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Attention-Aware Entity Alignment for Knowledge Graphs. Proc. VLDB Endow. 13, 11 (2020), 2326–2340. http://www.vldb.org/pvldb/vol13/p2326-sun.pdf

[9] Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[10] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Max Berrendorf. 2021. Answering Complex Queries in Knowledge Graphs with Bidirectional Sequence Encoders. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, 4968–4977. https://ojs.aaai.org/index.php/AAAI/article/view/16630

[11] Xian Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational Reflection Entity Alignment. In CIKM ’20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, Mathieu d’Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1095–1104. https://doi.org/10.1145/3340531.3432001

[12] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeny Gabrielovitch. 2015. A review of relational machine learning for knowledge graphs. Proc. IEEE 104, 1 (2015), 11–33.

[13] Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. QueryBox: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In 8th International Conference on Learning Representations, ICLR 2020, Addas Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. https://openreview.net/forum?id=Bljr4s56DS

[14] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeny Gabrielovitch. 2015. A review of relational machine learning for knowledge graphs. Proc. IEEE 104, 1 (2015), 11–33.

[15] Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. QueryBox: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In 8th International Conference on Learning Representations, ICLR 2020, Addas Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. https://openreview.net/forum?id=Bljr4s56DS

[16] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeny Gabrielovitch. 2015. A review of relational machine learning for knowledge graphs. Proc. IEEE 104, 1 (2015), 11–33.
A ADDITIONAL TABLES

A.1 Generalized Hölder Mean

Table 3: Aggregation functions formulated with the generalized Hölder (i.e., power) mean $M_p(x_1,\ldots,x_n) = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$ as defined in [10]. Note that $\lim_{p \to 0} M_p$ asymptotically approaches the geometric mean.

| Name             | $p$       | Definition   |
|------------------|-----------|--------------|
| Max              | $+\infty$ | $\max_i f(r_i)$ |
| Quadratic Mean   | 2         | $\sqrt[2]{\frac{1}{n} \sum_{i=1}^{n} f(r_i)^2}$ |
| Arithmetic Mean  | 1         | $\frac{1}{n} \sum_{i=1}^{n} f(r_i)$ |
| Geometric Mean   | 0         | $\sqrt[n]{\prod_{i=1}^{n} f(r_i)}$ |
| Harmonic Mean    | -1        | $\left(\frac{1}{n} \sum_{i=1}^{n} f(r_i)^{-1}\right)^{-1}$ |
| Min              | $-\infty$ | $\min_i f(r_i)$ |

B DERIVATIONS OF ADJUSTMENTS

B.2 Adjusting the MRR

1. Chance-adjusted MR. The chance-adjusted MR (called adjusted mean rank (AMR) in [7]) is given as:

$$MR^*(r_1,\ldots,r_n) = \frac{MR(r_1,\ldots,r_n)}{E[MR]} = \frac{2}{N(N-1)} \sum_{i=1}^{n} r_i$$

2. Re-indexed Chance-adjusted MR. The authors of [7] introduced a re-indexed variant of the AMR named AMRI that is given as follows:

$$AMRI(r_1,\ldots,r_n) = 1 - \frac{MR(r_1,\ldots,r_n) - 1}{E[MR]} \in [-1, 1]$$

B.3 Adjusting the MRR

The expectation and variance of an inverse-uniform distributed variable $r_i$ with parameters $a = 1$ and $b = N_i$ is given as follows:

$$E\{r_i\} = \frac{N_i + 1}{2}$$

$$\text{Var}\{r_i\} = \frac{N_i^2 - 1}{12}$$

The expectation of the MR metric is given as:

$$E[MR] = \frac{1}{n} \sum_{i=1}^{n} E\{r_i\} = \frac{1}{n} \sum_{i=1}^{n} \frac{N_i + 1}{2} = \frac{N + 1}{2}$$

The variance of the MR metric is given as:

$$\text{Var}\{r_i\} = \frac{\left(\frac{N_i + 1 - 1}{2}\right)^2 - 1}{12} = \frac{N^2 - 1}{12}$$

The expectation of the AMRI is given as:

$$E[AMRI] = \frac{1}{n} \sum_{i=1}^{n} E\{r_i\} = \frac{1}{n} \sum_{i=1}^{n} \frac{N_i}{2} = \frac{N}{2}$$

The variance of the AMRI is given as:

$$\text{Var}\{AMRI\} = \frac{\left(\frac{N_i}{2}^{-1}\right)^2 - 1}{12} = \frac{N^2 - 1}{12}$$
The variance of the MRR metric is given as:

\[ \text{Var} \left[ \text{MRR} \right] = \text{Var} \left[ \frac{1}{n} \sum_{i=1}^{n} r_i^{-1} \right] = \frac{1}{n} \sum_{i=1}^{n} \text{Var} \left[ r_i^{-1} \right] \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{N_i - 1} \ln \frac{N_i}{N_i - 1} \approx \frac{1}{N} \ln N - \frac{1}{N - 1} \]  (13)

B.2.1 Chance-adjusted MRR. The chance-adjusted MRR is given as:

\[ \text{MRR}^*(r_1, \ldots, r_n) = \frac{\text{MRR}(r_1, \ldots, r_n)}{\mathbb{E} [\text{MRR}]} = \frac{N - 1}{N \ln N} \sum_{i=1}^{n} r_i^{-1} \]  (14)

B.3 Adjusting the Hits at \( k \)

The expectation of \( H_k \) is derived first by deriving the expectation of the discrete indicator function \( f(x) = \mathbb{1} [x \leq k] \) (Equation 15) then applying it in full under the assumption that \( N_i = N \) for all \( i \) (Equation 17). The expectation of \( r_i \) is given as:

\[ \mathbb{E} \left[ \mathbb{1} [r_i \leq k] \right] = \frac{k}{N_i} = \frac{k}{N} \]  (15)

The variance of \( r_i \) is given as:

\[ \text{Var} \left[ \mathbb{1} [r_i \leq k] \right] = \mathbb{E} \left[ \mathbb{1} [r_i \leq k] \right] \times (1 - \mathbb{E} \left[ \mathbb{1} [r_i \leq k] \right]) \]

\[ = \frac{k}{N_i} \times \left( 1 - \frac{k}{N_i} \right) = \frac{k(N_i - k)}{N_i} \]

\[ = \frac{k}{N} \times \left( 1 - \frac{k}{N} \right) = \frac{k(N - k)}{N} \]  (16)

The expectation of the \( H_k \) metric is given as:

\[ \mathbb{E} [H_k] = \mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1} [r_i \leq k] \right] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E} \left[ \mathbb{1} [r_i \leq k] \right] \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \frac{k}{N_i} = \frac{k}{N} \]  (17)

The variance of the \( H_k \) metric is given as:

\[ \text{Var} [H_k] = \text{Var} \left[ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1} [r_i \leq k] \right] = \frac{1}{n} \sum_{i=1}^{n} \text{Var} \left[ \mathbb{1} [r_i \leq k] \right] \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \frac{k(N_i - k)}{N_i} = \frac{k(N - k)}{N} \]  (18)

B.3.1 Chance-adjusted \( H_k \). The chance-adjusted \( H_k \) is given as:

\[ H_k^*(r_1, \ldots, r_n) = \frac{H_k^*(r_1, \ldots, r_n)}{\mathbb{E} [H_k]} = \frac{1}{k} \sum_{i=1}^{n} \mathbb{1} [r_i \leq k] \]  (19)

B.3.2 Re-indexed Chance-adjusted \( H_k \). Combining the facts that ranks are 1-indexed and the \( H_k \in [0, 1] \), the \( H_k \) can be adjusted as in Equation 20. A negative value of the \( AH_k \) corresponds to performance below random, zero corresponds to random performance, and 1 to optimal performance. The adjustment for \( H_k \) is affine with respect to a dataset’s filtering constant, so it can be applied to results after evaluation.

\[ AH@k = \frac{H_k - \mathbb{E}[H_k]}{1 - \mathbb{E}[H_k]} \in \left[ -\frac{\mathbb{E}[H_k]}{1 - \mathbb{E}[H_k]}, 1 \right] \]  (20)

Note the lower bound was calculated by inserting \( \min H_k \) as the value for \( H_k \), which is 0.

B.4 Remaining Adjustments

Identifying a closed-form expectation for the geometric mean rank (GMR) is difficult because of the inclusion of a product. Further, identifying closed-form expectations for harmonic mean rank (HMR), inverse geometric mean rank (IGMR), and inverse mean rank (IMR) come from the difficulty of introducing inverses. For these, we implemented a simple workflow to numerically estimate the adjustment constant in PyKEEN that can be applied as an affine transformation after the fact.