



# Automated assessment of knowledge hierarchy evolution: comparing directed acyclic graphs

Guruprasad Nayak<sup>1</sup> · Sourav Dutta<sup>2</sup>  · Deepak Ajwani<sup>2</sup> · Patrick Nicholson<sup>2</sup> · Alessandra Sala<sup>2</sup>

Received: 1 May 2018 / Accepted: 23 November 2018 / Published online: 17 December 2018  
© Springer Nature B.V. 2018

## Abstract

Automated construction of knowledge hierarchies from huge data corpora is gaining increasing attention in recent years, in order to tackle the infeasibility of manually extracting and semantically linking millions of concepts. As a knowledge hierarchy evolves with these automated techniques, there is a need for measures to assess its temporal evolution, quantifying the similarities between different versions and identifying the relative growth of different subgraphs in the knowledge hierarchy. In this paper, we focus on measures that leverage structural properties of the knowledge hierarchy graph to assess the temporal changes. We propose a principled and scalable similarity measure, based on *Katz similarity* between concept nodes, for comparing different versions of a knowledge hierarchy, modeled as a generic directed acyclic graph. We present theoretical analysis to depict that the proposed measure accurately captures the salient properties of taxonomic hierarchies, assesses changes in the ordering of nodes, along with the logical subsumption of relationships among concepts. We also present a linear time variant of the measure, and show that our measures, unlike previous approaches, are tunable to cater to diverse application needs. We further show that our measure provides *interpretability*, thereby identifying the key structural and logical difference in the hierarchies. Experiments on a real DBpedia and biological knowledge hierarchy showcase that our measures accurately capture structural similarity, while providing enhanced scalability and tunability. Also, we demonstrate that the temporal evolution of different subgraphs in this knowledge hierarchy, as captured purely by our structural measure, corresponds well with the known disruptions in the related subject areas.

**Keywords** Knowledge hierarchy matching · DAG similarity · Concept tracking · Semantic subsumption · Taxonomy evaluation

---

✉ Sourav Dutta  
sourav.dutta@nokia-bell-labs.com

Extended author information available on the last page of the article

## 1 Introduction

The current revolution in Semantic Web, Natural Language Processing and user modeling has witnessed the transition from documents and keywords to knowledge, entities and relationships. This has led to the advent of large knowledge repositories such as DBpedia (Lehmann et al. 2015), Freebase (Bollacker et al. 2008), and YAGO (Suchanek et al. 2008)—forming the backbone of semantic search, personalization, recommendation and textual entailment (Fensel 2005; Maedche and Staab 2002; Harabagiu et al. 2003; Geffet and Dagan 2005). Knowledge hierarchies, or taxonomies, represent concepts and relations as hierarchies expressing semantic connections via parent–child or hypernym–hyponym edges, enabling easy navigation across concepts and information linking. These hierarchies typically represent information in the form of Directed Acyclic Graphs (DAGs).

### 1.1 Motivation

Traditionally, taxonomies have been created and curated manually by domain experts. However, with the enormous expanse of data available world wide, automated techniques to construct huge taxonomies have been proposed (Liu et al. 2012; de Knijff et al. 2011; Vedula et al. 2018). Modern linked open data repositories like the DBpedia taxonomy<sup>1</sup> (over Wikipedia pages) contains over 6 million nodes (representing concepts) and 64 million directed edges (capturing relations) organized as a DAG.

As the knowledge hierarchies evolve, there is a need for *taxonomy evaluation*, i.e., comparing hierarchies and quantifying their similarity. Linked knowledge repositories are conceptual models representing information from diverse sources, and may disagree among each other on certain aspects. Assessing such disagreements in the evolution process is crucial for information veracity and integration (Pesquita et al. 2009). Further, identifying the source of disagreement across the different hierarchies might provide interesting insights into the evolution or construction of the hierarchies (i.e., interpretability). Thus, it would help for translating semantically related query onto similar hierarchies (David et al. 2010).

Traditionally, the generated hierarchies are presented to domain experts for manual evaluation and curation. However, such techniques are practically infeasible for modern taxonomies comprising millions of concepts and relations. Thus, there is a need for automated techniques to assess the evolution of very large knowledge hierarchies that capture logical semantic subsumption.

In building a learning system to capture semantic subsumption and evaluating knowledge hierarchies, there are various features that can be leveraged. Traditional features based on syntactic analysis and natural language processing can aid in identifying similar nodes. For instance, one can consider various distance measures on the string description of the nodes in the knowledge hierarchies to determine if they refer to the same node or not. However, features based on syntactic approximations induce considerable errors and do not provide enough accuracy to assess the evolving knowledge hierarchies on their own. Similarly, word embeddings can capture semantic similarities but it is non-trivial to determine

---

<sup>1</sup> Hierarchical ordering of DBpedia concepts and Wikipedia categories based on *Simple Knowledge Organization System (SKOS)* : broader relationships between the corresponding Wikipedia pages is henceforth referred to as the DBpedia taxonomy.

semantic subsumption just based on embeddings. Specifically, it is not clear how to determine whether two nodes  $X$  and  $Y$  are siblings or  $X$  is a parent and  $Y$  a child or  $Y$  is a parent and  $X$  a child, based on word embeddings alone. Ideally, in addition to the semantic information coming from the syntactic and embedding features, we would like to leverage structural properties of the entire knowledge hierarchy graph to assess the changes. This has the potential to not just quantify how much change has happened in the knowledge hierarchy, but also to understand which regions of the hierarchy are undergoing more changes relative to others, enabling us to gain deeper insight into the evolution of knowledge hierarchies.

In this paper, we explore if there is a scalable, tunable and interpretable graph similarity measure that can be leveraged to quantify the changes in the knowledge hierarchy. To this end, we identify a set of key characteristic properties that a graph similarity measure indicating semantic subsumption should possess. We then show that a measure based on a group Katz similarity satisfies these requirements and is a good indicator of logical subsumption change happening in a knowledge hierarchy. Then, we go to the extreme and assess how well we can identify the relative growth of different subject areas in the knowledge hierarchies merely based on our graph structure comparison measure, independent of any syntactic or embedding feature. We show that the proposed measure is able to identify key changes in the organization of popular subject areas in the DBpedia taxonomy and these changes match well with external events.

## 1.2 State-of-the-art

The literature hosts a variety of techniques for assessing the quality of taxonomies and evaluating their similarities with other structures, and can be broadly categorized as (Brank et al. 2006):

- Traditionally, the generated hierarchies are presented to domain experts for manual evaluation and curation. However, such techniques are practically infeasible for modern taxonomies comprising millions of concepts and relations.
- Qualitative indicators compare taxonomies based on structural overlap, accuracy, consistency and rigidity (Volker et al. 2010). However, differences in structure and scope might stem from input data, domain of interest and granularity of application. Further, they fail to capture transitive closure or logical subsumption of hyponym relations.
- Automated measures for evaluating taxonomies against a gold standard taxonomy pose a rigid domain dependency, and are hence unable to provide a generic concept of similarity. For example, the concept “*chiaroscuro*” is categorized under *picture*, *image*, *icon* (along with concept “collage”) in WordNet, while it falls under *perspective and shading technique* category in Art and Architecture Thesaurus (with “collage” in *image making processes and techniques*) (Velardi et al. 2012).
- Similarity measures for tree structures or undirected graphs (McVicar et al. 2016; Koutra et al. 2016) are unable to model the structural complexity (like multiple parents) within DAGs or consider the semantic coherence of parent–child links that characterizes hierarchical knowledge sources.

Further, none of the above approaches are scalable in practice for comparing huge taxonomies, and also do not provide tunability to characterize the degree of similarity for different scenarios. Also, most existing measures require many hours or days of computation on a

typical workstation, even for small sub-graphs of DBpedia taxonomy. In contrast, we aim for a measure that is significantly faster and scalable.

### 1.3 Contributions

This paper proposes a novel similarity measure for comparing non-specialized hierarchically linked knowledge structures in a principled and scalable fashion. We show that our proposed measure exhibits the essential properties to capture structural similarity as well as logical coherence and subsumption in directed acyclic hierarchies. We also demonstrate that the measure is extremely scalable for massive taxonomies and tunable across diverse applications. In a nutshell, the contributions of this paper are:

- A generalized and scalable measure, based on weighted connected path count, to automatically compute the similarity between taxonomies (and directed acyclic graphs), capturing both the structural similarity and the logical subsumption of concepts. Particularly, we focus on the scenario of comparing and evaluating taxonomies in an automated fashion.
- Theoretical analysis to demonstrate that the performance of our similarity measure conforms to commonsense and intuitive properties.
- A linear time variant of the measure for practical applications.
- Detailed experimental evaluations on the desired properties of a similarity measure, and the novel concept of *interpretability*.
- Experimental validation on large real linked knowledge repositories to showcase improved scalability and tunability of our measure.

Furthermore, the proposed technique can be generalized beyond taxonomies as a measure for computing similarities between Directed Acyclic Graphs (DAGs). The similarity measures proposed in this paper have been implemented in C and the code for the same is available at <https://github.com/guruprasadnk7/DAGSimilarityKatz>.

## 2 Related work

Traditional works on taxonomy construction and comparison relied on *domain experts* to evaluate the output hierarchies (Adams 1972) or test efficacy through the application performance. However, increase in the amount and diversity of information necessitates the need for automated evaluation methods (Dellschaft and Staab 2006). Evaluating a taxonomy with respect to a given reference for semantic web has been proposed (Dellschaft and Staab 2006; Brank et al. 2005; Velardi et al. 2013). Automatically learned ontology can be compared against a gold standard by transforming concepts and their properties into term distributions for pairwise concept similarity between the hierarchies (Zavitsanos et al. 2011; Brank et al. 2006) using Rand index cluster analysis (Rand 1971). However, such techniques converge to 1 as the number of clusters increases (Fowlkes and Mallows 1983). Taxonomies are also evaluated based on the *lexical content*, i.e., by comparing terms across taxonomies. The degree of edge overlap between the taxonomies is a common similarity measure (Bordea et al. 2016; Maedche and Staab 2002) to compute precision, recall and F1-score. However, these methods consider differences in the parent–child structural

relationships only, and does not take into account all connected pair of concepts (Guarino and Welty 2002).

*Concept hierarchies* capture the relative position of concepts between taxonomies for comparison (Dellschaft and Staab 2006; Brank et al. 2006; Velardi et al. 2013), while *semantic similarity* measures consider the depth of the least common ancestor to judge the goodness of node addition in existing hierarchies (Jurgens and Pilehvar 2016; Wu and Palmer 1994). However, this only provides “in-vitro comparison” of taxonomy enrichment, and none of the above measures capture the relational subsumption of concept linking in hierarchies. Measuring the similarities in language parse trees have been studied based on edge overlap by Cai and Knight (2013).

Among state-of-the-art measures for taxonomy comparison (Bordea et al. 2016, 2015), the most commonly used is the *Fowlkes-Mallows* (FM) measure (Fowlkes and Mallows 1983; Wagner and Wagner 2007; Velardi et al. 2013). It computes a hierarchical graph partitioning (based on vertex cuts with possible overlaps) of concepts. The taxonomies are then compared based on the agreement on the allocation of every vertex-pair to the same partition, with associated path lengths assigned weights. However, the FM measure suffers from 4 key drawbacks—(1) *dependence on hierarchy of vertices to capture graph structure (uniquely defined only for trees) and hence weights assignment is ad-hoc*; (2) *not scalable as it involves vertex partition at each level of the hierarchy*; (3) *not tunable to suit the sensitivity needs of different applications*; and (4) *lacks interpretability*.

A parallel body of related research includes *tree* and *graph similarity* measures. However, it mostly involves capturing graph similarities with unlabeled nodes and undirected edges (Foggia et al. 2014; Elghawalby and Hancock 2008). Recent graph embedding approaches (Goyal and Ferrara 2017, 2018) have been proposed for comparison based on high-dimensional representation, but such techniques suffer from scalability issues. Supervised methods using *guided walk* on graph (Levin et al. 2016) have been studied to obtain similarities for recommendations—but requires large training data.

Shervashidze et al. (2009) proposed a scalable graph kernel approach based on the frequency of subgraph patterns called *graphlets*. Recently, DeltaCon (Koutra et al. 2016), a scalable similarity measure was proposed for undirected labeled graphs using pairwise node similarities, along with a faster approximate version using belief propagation (Koutra et al. 2011). Kendall–Tau based distance measures were also proposed (Brandenburg et al. 2012) for comparing partial orderings. However such approaches consider undirected graphs or unlabeled nodes, and fail to model the structure of linked hierarchies (e.g., taxonomies) represented as DAGs with labeled nodes.

Recently, a similarity measure for DAGs comparing partial rankings, based on the edge set of the DAGs, assigns different weights to different kinds of edge differences was proposed (Malmi et al. 2015). However, it ignores paths between nodes (uses direct relations) thus failing to capture multi-hop semantic subsumption present in hierarchies. Other distance metrics like *tree edit* (Bille 2005) and *graph edit* distances (Gao et al. 2010) were also explored. A graph similarity approach comparing taxonomies by defining edit distance on taxonomies was proposed (McVicar et al. 2016). However, the input taxonomies were assumed to be trees, which is not generally true (Bordea et al. 2016; Velardi et al. 2013; Kozareva and Hovy 2010). Also, it suffers from quadratic time complexity and hence is not scalable.

Similarity measures like *Tversky’s parameterized ratio model* (Cross et al. 2013) for comparing ontologies are sensitive to parameter setting and/or involve computation of Information Content (IC) of concepts, making it practically infeasible for large hierarchies. Other approaches in this area (David et al. 2010; d’Aquin 2009) and

entity-disambiguation (Hulpuş et al. 2015) consider mutual agreement scores or centrality based edge weights in DAGs, but do not capture the logical subsumption over longer paths.

### 3 Properties for similarity measure

Knowledge repositories and taxonomies model *cotopy* of concepts (i.e., super and sub-concepts) as directed parent–child relationships. Hence, a measure to compare such hierarchies should be sensitive to the explicit structural similarity, directedness of semantic relations, and the implicit logical subsumption of concepts. An automated comparative measure, to be principled and practically sound, should ideally demonstrate the following *qualitative properties* to approximate the reasoning and commonsense of domain experts.

(1) *Sensitivity to concept hierarchy* The key characteristic of linked data is the hierarchical relations among concepts, i.e., *concept hierarchy*. Specifically, a directed parent–child edge depicts two related concepts, with the parent representing a broader category. For example, in Fig. 1a, the edge *Mammals* → *Felines* models the relation that *Felines* are a sub-type of *Mammals*.

On the other hand, an oppositely directed edge (from *Felines* to *Mammals*), although structurally similar (ignoring edge direction), denotes a completely different and possibly wrong relation. Similarity measure should be aware of this asymmetry to accurately capture concept hierarchy. Hence, direct use of similarity measures designed for undirected graphs, would implicitly assume symmetry and fail to account for the strict ordering of concepts in a hierarchy.

(2) *Proximity of least common ancestor* Given two structurally different taxonomies, it may not be possible to completely characterize the effects of the differences on their similarity. However, analytically we can argue about the behaviour that a similarity measure should demonstrate for varying degrees of dissimilarity.

Consider two structurally different taxonomies, *A* and *B*, shown in Fig. 1b to be compared with taxonomy *X* (Fig. 1a). Assume taxonomy *A* to link concept *Domestic Cats* as a sub-type of *Bovines*, while *B* connects *Domestic Cats* to *Reptiles*. Although both taxonomies structurally differ in only one edge, intuitively, taxonomy *A* is semantically and logically more similar (than *B*) to *X*, as the concept *Domestic Cats* is still categorized as a sub-type of *Mammals* in *A*, while it falls under *Reptiles* in *B*. A similarity measure should capture such multi-hop transitive relations.

The above notion of semantic similarity (with structural differences) is captured by the *Least Common Ancestor* (LCA) of the “differing” concept nodes (Resnik 1995). Specifically, greater the distance of the LCA from the root, more similar the taxonomies are. Intuitively, a larger distance between the root and the LCA implies a more localized structural change, incurring less diffusion of concept semantic relatedness. Hence, a similarity measure should monotonically decrease with decrease in distance of the LCA from the root.

However, the above notion of LCA property is not unique, and different structures might evaluate to the same value. Comparing such scenarios presents an extremely hard task, difficult even for humans—consider quantifying the dissimilarity between two cases where *Lizards* is either linked to *Tigers* or to *Domestic Cats*. Interestingly, a few of the above scenarios can be captured by the LCA property, as the depth of a concept node provides a measure of its *specificity*. Nodes closer to the root represent broader domains, while those closer to the leaves depict specific concepts. As such, for the same LCA, the further down in the hierarchy the structural dissimilarity occurs, the greater the semantic diffusion tends to be.

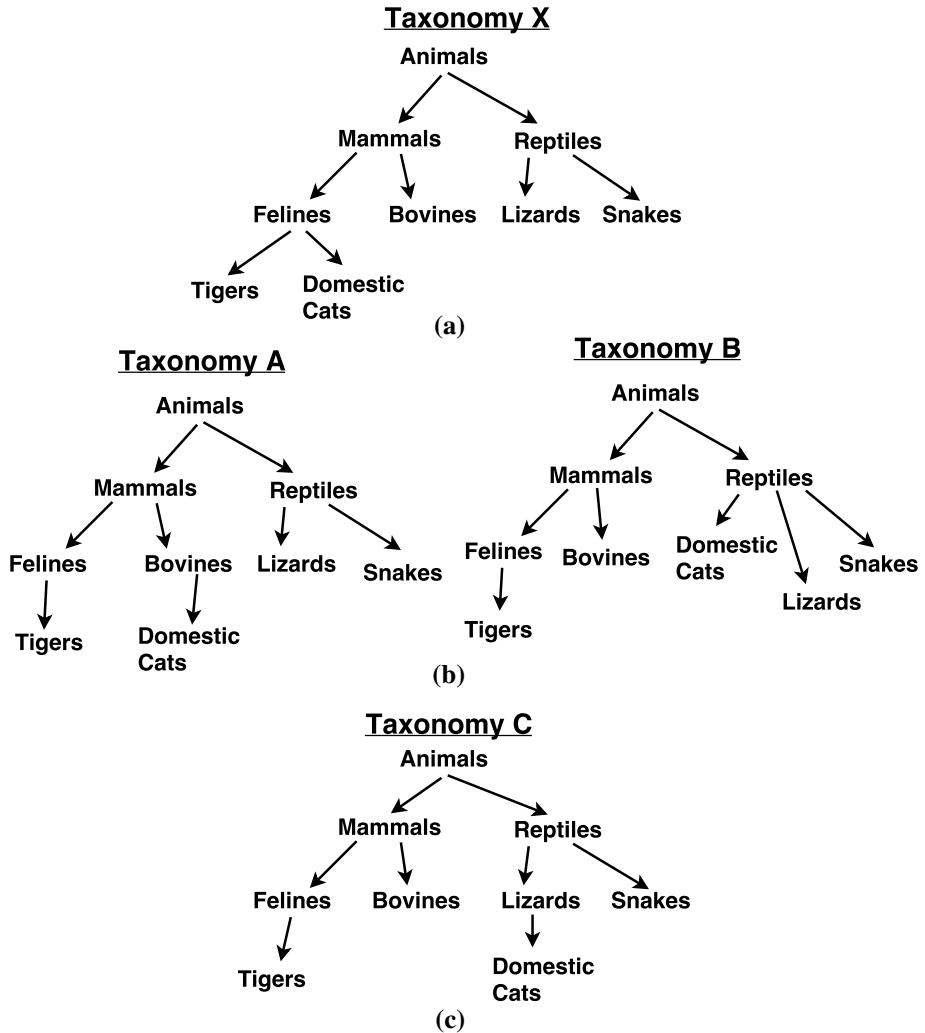


Fig. 1 Depiction for properties of similarity measure

For example, consider taxonomy *C* in Fig. 1c to link *Domestic Cats* as a sub-type of *Lizards*. The distance of the LCA (between positions of *Domestic Cats*) from root is the same for both taxonomies *B* and *C* when compared with Taxonomy *X*. However, intuitively, representing *Domestic Cats* as type of *Lizards* seems worse than *Reptiles*. Hence, the degree of semantic “damage”, in this case, can be captured by the distance of the concept node (with structural difference) from the LCA, and should monotonically increase with increase in the distance of the concept from the LCA.

Thus, the degree of similarity between structurally different hierarchies can be captured by *proximity of LCA* based on:

- (a) Distance from the root to LCA of the structurally different node.
- (b) Distance of the concept node from the LCA.

This characterization takes into account the logical subsumption of concepts, and a principled similarity measure should respect the monotonic nature of the above distances (i.e., no. of hops).

(3) *Importance of relationship* One of the subtleties of ancestor-descendant relationships is the *diminishing importance* of relationship as we move up (closer to the root) the hierarchy. A parent–child edge provides a much stronger semantic coupling than the node and ancestors. For example, the relation *Felines* → *Tigers* depicts a stronger connection than *Animals* → *Tigers*. Hence, a similarity measure should incorporate this notion of relation weights. However, the cumulative effect of a large difference in ancestral relations might degrade the overall quality of a taxonomy, and should be captured by the aggregation of edge weights by the measure.

Further, the practical utility of a measure in comparing huge hierarchies arises from a number of *computational* properties, like:

- *Scalability* Modern taxonomies comprises millions of concepts and edges. For example, DBpedia contains around 5 million concepts, while Google Knowledge Graph contains nearly 500 million concepts. The practical feasibility of a similarity measure rests on its computational efficiency to gracefully scale for huge taxonomies.
- *Tunability* A similarity measure provides a comparative measure depicting the closeness of two hierarchies. However, a few semantic inconsistencies or limited structural difference might be very “damaging” for certain applications, while it may be within the tolerable error rate for others. Hence, a *robust* similarity measure should be tunable to reflect the error sensitivity in diverse domains.
- *Interpretability* Interpreting the differences between taxonomies beyond a similarity score, and attributing the dissimilarity to nodes and edges characterizing it would enable analysis of hierarchies.

It should be noted, that intermediate concepts in one of the taxonomies (modeled with higher granularity), does not affect the meaning of the relationships in any way, and a measure should be agnostic to such scenarios. In this work, we show that our proposed similarity measure conforms to the above intuitive properties, lending tunability and scalability for huge real-world taxonomies with millions of nodes.

## 4 Proposed measures

Consider two DAGs,  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  representing two taxonomies, with identical vertex set  $V^2$  and edge sets  $E_1$  and  $E_2$  respectively. Intuitively, two graphs are said to exhibit high similarity if every pair of nodes are similar (i.e., high pair-wise node similarity). It is important to note that the same concept may be represented in different surface-forms (e.g., Neuroscience and Neurobiology) in different knowledge sources. In this work, we assume that the concepts are disambiguated and represented identically by a canonical form (possibly from a standard hierarchy like DBpedia). We further consider the taxonomies to contain the same set of vocabulary (i.e., vertex labels). For other scenarios, non-overlapping vertices (i.e., differing labels) can either be added as singletons (to model

<sup>2</sup> For non-identical sets, nodes present in one but absent in other are added as singletons.



emergence of concept) or removed and the children connected to its parents (to capture concept evolution) in the corresponding taxonomies. We adopt the latter strategy in our experimental settings.

To model node similarities, we adapt the *Katz similarity* measure (Katz 1953; Ou et al. 2016), which captures multiple short directed paths between vertices, and provides a good indicator of semantic subsumption.

**Definition 1** (*Katz similarity*) Given vertices  $u$  and  $v$ , the Katz similarity ( $KS$ ) is:

$$KS(u, v) = \sum_l \sum_{\text{Paths of length } l \text{ from } u \text{ to } v} \alpha^l \quad [\text{where, } 0 < \alpha < 1]$$

The idea is that the effectiveness of a link between two nodes is governed by a constant probability and their relatedness is accumulated over the paths, i.e., more and shorter relation paths depict higher relatedness. The Katz measure inherently exhibits the following properties, making it better than other candidates like *link precision*.

- *Asymmetry* The Katz similarity between vertices  $u$  and  $v$  is computed based on the directed paths from  $u$  to  $v$  (and not from  $v$  to  $u$ ). This captures the asymmetry present in directed concept relations, and hence is *sensitive to concept hierarchy*.
- *Attenuation with path length* The Katz similarity assigns a weight of  $\alpha^l$  to every path of length  $l$  between two nodes. Since  $\alpha \in (0, 1)$ , paths have diminishing weight as their lengths increase. This provides two-fold advantages: (1) Direct parent–child relationships are prioritized and captures *importance of relationship*, and (2) Provides tunability for requisite degree of semantic subsumption.
- *Accounting for multiple paths* A key distinguishing feature is that every path between nodes contributes additional weight to the Katz similarity. This is important since taxonomies are DAG structures, and multiple paths should be accounted.

We next describe how the Katz similarity measure and the above properties can be extended for comparing taxonomies.

#### 4.1 Katz similarity between graphs

We initially represent a DAG,  $G = (V, E)$  by its *Katz Similarity Vector* ( $KSV$ ) capturing the similarities between vertex pairs in  $G$ .

**Definition 2** (*Katz similarity vector*) The  $p$ th element of the vector encodes the Katz similarity between the  $p$ th vertex pair in  $V \times V$ .

The similarity between two DAGs is then defined via the  $KSV$  as,

**Definition 3** (*Katz graph similarity*) Given Katz similarity vectors  $KSV_1$  and  $KSV_2$  for DAGs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  resp., the *Katz Graph Similarity*,  $KGS$ , between them is defined as,

$$KGS(G_1, G_2) = \frac{2}{1 + \exp(\gamma \cdot \|KSV_1(:, i) - KSV_2(:, i)\|_1)} \tag{1}$$

where  $\|\cdot\|_1$  is the  $L_1$ -norm of the  $i$ th vector differences and  $\gamma > 0$  is a tunable parameter that controls the sensitivity of the measure. Thus,  $KGS(G_1, G_2)$  ranges from 0 (completely different graphs) to 1 (identical edge sets).

$KGS$  captures the structural similarity between the graphs. Given vectors  $KSV_1$  and  $KSV_2$ , computing  $KGS$  takes  $O(|RV_1| + |RV_2|)$  time, where  $RV_i$  is the number of reachable vertex pairs  $(u, v)$  ( $v$  is reachable from  $u$ ) in graph  $G_i$ .

### 4.1.1 Computing the Katz similarity vectors

The Katz Similarity Vector ( $KSV$ ) for graph  $G$  is a vector of  $|V| \times |V|$  dimensions encoding the Katz similarities between vertex pairs in  $G$ . However, computing similarities between all vertex pairs independently is computationally infeasible. The acyclic nature of DAGs can thus be leveraged to speedup computations by pruning unreachable vertex pairs, using *topological ordering*. At each traversal iteration, nodes with zero in-degree are assigned the current topological level (initialized to 1). Subsequently, the graph is updated by deleting these nodes and their edges, and the current level is incremented. As there are no cycles, every vertex is assigned to a unique topological level, defining our vertex ordering in  $O(|V| + |E|)$  time.

Observe, the Katz similarity between vertex pairs with one vertex at *level*  $> k$  and the other at *level*  $k$  is *zero*, since there are no paths starting from a *level*  $> k$  node and ending at a node at *level*  $k$ . This restricts the number of vertex pairs between which the Katz similarity is computed. The following lemma states how the Katz similarity between vertices is computed based on the vertex ordering.

**Lemma 1** *The Katz similarity between nodes  $u$  and  $v$ ,  $KS(u, v)$ , is computed using the Katz similarity between  $u$  and every parent  $p$  of  $v$  as,*

$$KS(u, v) = \alpha \times \left( \sum_{p \in \text{parents}(v)} KS(u, p) \right) + \alpha \times \mathbf{I}(u \rightarrow v) \tag{2}$$

where indicator function  $\mathbf{I}(u \rightarrow v)$  is 1 if there is an edge  $(u, v)$ , 0 otherwise.

**Proof** Every path from  $u$  to  $p$  (with length  $l$ ) provides a unique path from  $u$  to  $v$  (of length  $l + 1$ ) by appending the edge from  $p$  to  $v$  (as  $p$  is a parent of  $v$ ). Since  $v$  is reachable only via its parents, this provides an exhaustive enumeration of paths to  $v$ . Observe, there are no self edges (i.e.,  $KS(u, u) = 0$ ), and  $\mathbf{I}()$  captures the case where  $u$  is a parent of  $v$ .  $\square$

Let  $KSV(:, p)$  denote a sub-vector encoding the Katz similarity between every vertex  $v$  and vertex  $p$ . Equation (2) can be rewritten as,

$$KSV(:, v) = \alpha \times \left( \sum_{p \in \text{parents}(v)} KSV(:, p) \right) + \alpha \times \mathbf{I}(:, v) \tag{3}$$

where  $\mathbf{I}(\cdot, v)$  is an indicator vector of length  $|V|$ , with 1 for all vertices that are parents of  $v$ , and 0 otherwise.

Lemma 1 and the topological ordering can be used to efficiently compute KSVs. The Katz similarity sub-vectors for nodes at each level of the ordering is computed iteratively. Since parents of nodes at level  $k$  lie at  $level < k$ , the sub-vectors of parents are already computed before reaching a node, and only the non-zero terms in sub-vectors (corresponding to their parents) need to be computed. Thus, finding reachable nodes to a node with in-degree  $d_{node}$ , incurs at most  $d_{node}$  computations. The total cost of computing KSVs is  $O(D \times |RV|)$ , where  $|RV|$  denotes the number of reachable vertex pairs (i.e.,  $(u, v)$ , for  $v$  reachable from  $u$ ) and  $D$  is the *maximum in-degree*. Note, in typical knowledge structures,  $|RV| \ll |V| \times |V|$ .

Hence, using Definition 3, the total complexity of computing the *Katz similarity between graphs*  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  is bounded by  $O(|V| + |E_1| + |E_2|) + O(D_1 \times |RV_1|) + O(D_2 \times |RV_2|) + O(|RV_1| + |RV_2|)$ , where  $D_i$  and  $|RV_i|$  denotes the maximum in-degree and the number of reachable vertex pairs respectively. The practical run-time is dominated by  $O(D_1 \times |RV_1| + D_2 \times |RV_2|)$ , as  $|V|, |E_i| < |RV_i|$ .

We next present a faster approximate variant of the above similarity measure to cater to real-time needs, albeit with error tolerability.

### 4.2 Grouped Katz similarity between graphs

For scenarios where the number of reachable pairs in the hierarchies is large, computing Katz Similarity between graphs might be expensive. We thus propose the *Grouped Katz Similarity* measure, a faster approximation to the KGS, computable in  $O(|V| + |E_1| + |E_2|)$  time (with vertex set  $V$  and edge sets  $E_1, E_2$ ).

The *Grouped Katz Similarity* measure partitions the vertex set  $V$  into  $\mathcal{G}$  groups. Subsequently, instead of computing the Katz similarity between every pair of reachable vertices, the similarity of vertices to each of the  $\mathcal{G}$  groups is computed (to obtain a similarity vector). We now define the similarity between a vertex and a group.

**Definition 4** (*Katz group-vertex similarity*) The similarity between a vertex  $v$  and a group  $g$  is defined as the sum of the Katz similarities between vertex  $v$  and every vertex  $u$  in group  $g$ . Formally, Katz Group-Vertex Similarity,  $KG(g, v) = \sum_{u \in g} KS(u, v)$ .

Considering the Katz Similarity Vector (Definition 2) as a matrix<sup>3</sup> of size  $|V| \times |V|$ , the corresponding Grouped Katz Similarity Vector (*GKSV*) is a matrix of size  $|V| \times \mathcal{G}$  formed by summing up the columns corresponding to the vertices of each group. *GKSV* can then be plugged into Eq. (1) instead of the Katz Similarity Vectors to yield the *Grouped Katz Similarity* between two graphs. Thus,

**Definition 5** (*Grouped Katz similarity for graphs*) For two directed acyclic graphs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$ , let  $GKSV_1$  and  $GKSV_2$  be the Grouped Katz Similarity Vectors (vertex set  $V$  partitioned into  $\mathcal{G}$  groups). Then, the *Grouped Katz Similarity* between  $G_1$  and  $G_2$  is,

$$GKSG(G_1, G_2) = \frac{2}{1 + \exp(\gamma \cdot \|GKSV_1 - GKSV_2\|_1)} \tag{4}$$

<sup>3</sup> The Katz similarity vector of Eq. (1) is obtained by flattening this matrix into a vector.

Since the GKSVs are of length  $|V| \times \mathcal{G}$ , computing the final similarity using the vectors take  $O(|V| \times \mathcal{G})$  time. Lemma 1 can be easily extended for computing the Katz Group-Vertex Similarity.

**Lemma 2** *The Katz Group-Vertex Similarity between vertex  $v$  and group  $g$  is computed using the Katz similarity between vertex  $v$  and the parents of vertices within group  $g$  as,*

$$\begin{aligned} KG(g, v) &= \sum_{u \in g} KS(u, v) = \sum_{u \in g} \left( \alpha \times \left( \sum_{p \in \text{parents}(v)} KS(u, p) \right) + \alpha \times \mathbf{I}(u \rightarrow v) \right) \\ &= \alpha \times \left( \sum_{p \in \text{parents}(v)} \sum_{u \in g} KS(u, p) \right) + \alpha \times \sum_{p \in \text{parents}(v)} \mathbf{I}(p \in g) \\ &= \alpha \times \sum_{p \in \text{parents}(v)} (KG(g, p) + \mathbf{I}(p \in g)) \end{aligned}$$

where indicator function  $\mathbf{I}(u \rightarrow v)$  is 1 if edge  $(u, v)$  exists, and 0 otherwise. Similarly, indicator function  $\mathbf{I}(p \in g)$  is 1 iff vertex  $p$  is in group  $g$ .

**Computation** Computing the Grouped Katz Similarity between two hierarchies involves topologically ordering the vertices and computing the similarity of each group with the vertices (as in Lemma 2) to obtain the Grouped Katz Similarity Vectors. Finally, the GKSG is obtained from Eq. (4) using the grouped similarity vectors.

For each parent of a vertex,  $\mathcal{G}$  computations corresponding to each group are performed to generate the Grouped Katz Similarity Vectors, incurring a time complexity of  $O(\mathcal{G} \times |E|)$ . The total time for computing GKSG between graphs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  is, therefore, bounded by  $O(|E_1| + |E_2| + \mathcal{G} \times O(|V| + |E_1| + |E_2|))$ . Observe for  $\mathcal{G} \ll |V|, |E_1|, |E_2|$ , the complexity becomes  $O(|V| + |E_1| + |E_2|)$ , providing a linear time algorithm for comparing linked hierarchies.

We next discuss a special case of Grouped Katz Similarity, where the number of groups is 1, i.e., all vertices flattened into one group.

### 4.2.1 Katz Index similarity between graphs

The *Katz Index* is a centrality measure that measures the influence of a vertex in a graph (Katz 1953). Intuitively, it measures the number of paths that are incident onto a vertex from other vertices. Similar to the attenuation effect captured in Katz similarity, these paths are also weighted by their length (using an exponentially decreasing function). Katz index has been used to measure centrality of nodes in directed graphs (world wide web and citation networks) (Newman 2010; Ou et al. 2016).

We now show that the Katz Index is special case of the Grouped Katz Similarity (with  $\mathcal{G} = 1$ ) for measuring acyclic graph similarity. The *Katz Index* for vertex  $v$  in graph  $G = (V, E)$  is defined as,

$$KI(v) = \sum_{u \in V} \sum_l \sum_{\text{Paths of length } l \text{ from } u \text{ to } v} \alpha^l$$

For a general graph with adjacency matrix  $A$ , the vector of Katz indices,  $C_{Katz}$ , is computed by solving the following linear system.

$$C_{Katz} = ((I - \alpha A^T)^{-1} - I)\vec{1} \quad [\text{refer (Katz., 1953)}] \tag{5}$$

where  $I$  is the identity matrix and  $\vec{1}$  the identity vector. Solving this linear system involves computing the inverse and takes  $O(|V|^3)$ .

However, as before, for a DAG, the computation of Katz index can be scalably performed in terms of the parent vertex (similar to Lemma 1) as,

**Lemma 3** *The Katz index of a vertex  $v$  in a DAG,  $G = (V, E)$  is defined in terms of the Katz indices of its parent vertices as,*

$$KI(v) = \alpha \times \sum_{p \in \text{parents}(v)} (KI(p) + 1) \tag{6}$$

**Proof** Consider the definitions and equations in Sect. 4, we have,

$$\begin{aligned} KI(v) &= \sum_{u \in V} \sum_l \sum_{\substack{\text{Paths of length } l \\ \text{from } u \text{ to } v}} \alpha^l = \sum_{u \in V} KS(u, v) \\ &= \sum_{u \in V} \left( \alpha \times \sum_{p \in \text{parents}(v)} KS(u, p) + \alpha \times \mathbf{I}(u \rightarrow v) \right) \\ &= \alpha \times \sum_{p \in \text{parents}(v)} \sum_{u \in V} KS(u, p) + \alpha \times \sum_{u \in V} \mathbf{I}(u \rightarrow v) \\ &= \alpha \times \sum_{p \in \text{parents}(v)} (KI(p) + 1) \quad [\text{since, } \forall p, \mathbf{I}(u \rightarrow v) = 1] \end{aligned}$$

□

We next define the *Katz Index similarity for graphs (KIG)* as,

**Definition 6** (*Katz Index similarity for graphs*) Given DAGs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$ , let  $KIV_1$  and  $KIV_2$  be the respective Katz Index Vectors. The Katz index similarity between  $G_1$  and  $G_2$  is,

$$KIG(G_1, G_2) = \frac{2}{1 + \exp(\gamma \cdot \|KIV_1 - KIV_2\|_1)} \tag{7}$$

Intuitively, this measure captures the fact that if the centrality of vertices across the two graphs is similar then the graphs are similar.

Computing the Katz Index Vectors ( $KIV$ ) is similar to the previous approaches based on the parents of the vertices in the graph, with a complexity of  $O(|E| + |V|)$ . As,  $O(|V|)$  is required to compute the final similarity score between the vectors, the total time complexity for Katz Index Similarity between  $G_1$  and  $G_2$  is  $O(|E_1| + |E_2| + |V|)$ .

Finally, we provide a relationship between the different similarity scores obtained from the above measures for comparing DAGs.

**Theorem 1** Given directed acyclic graphs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$ , the Group Katz Similarity with  $G$  groups is lower bounded by Katz Graph Similarity and upper bounded by Katz Index Similarity, i.e.,  $KIG(G_1, G_2) \geq GKSG(G_1, G_2) \geq KGS(G_1, G_2)$

**Proof** Let us define the following based on the Katz’s vectors for the two graphs:

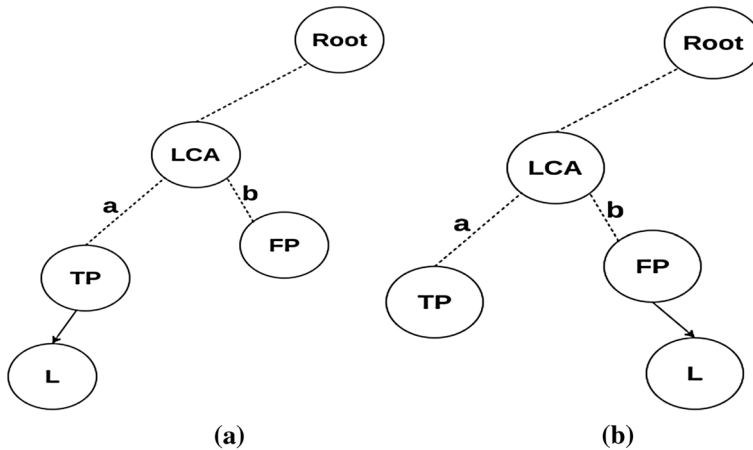
$$\begin{aligned}
 d_{KI} &= \sum_i |KI_1(i) - KI_2(i)| \\
 d_{GKSV} &= \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_1, \text{ and} \\
 d_{KSV} &= \sum_i \|KSV_1(:, i) - KSV_2(:, i)\|_1
 \end{aligned}$$

Observe,  $KGS$ ,  $GKSG$  and  $KIG$  are monotonically decreasing functions of  $d_{KSV}$ ,  $d_{GKSV}$  and  $d_{KI}$  respectively, for an input graph. Hence,

$$\begin{aligned}
 d_{KSV} &= \sum_i \|KSV_1(:, i) - KSV_2(:, i)\|_1 = \sum_i \left( \sum_j |KSV_1(j, i) - KSV_2(j, i)| \right) \\
 &= \sum_i \left( \sum_g \sum_{j \in g} |KSV_1(j, i) - KSV_2(j, i)| \right) \\
 &\geq \sum_i \left( \sum_g | \sum_{j \in g} (KSV_1(j, i) - KSV_2(j, i)) | \right) \\
 &= \sum_i \left( \sum_g | \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) | \right) \\
 &= \sum_i \left( \sum_g |GKSV_1(g, i) - GKSV_2(g, i)| \right) \\
 &= \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_1 = d_{GKSV}
 \end{aligned}$$

Thus,  $d_{KSV} \geq d_{GKSV}$ . Similarly, we show,

$$\begin{aligned}
 d_{GKSV} &= \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_1 = \sum_i \left( \sum_g | \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) | \right) \\
 &\geq \sum_i \left( | \sum_g \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) | \right) \\
 &= \sum_i \left( | \sum_j (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) | \right) \\
 &= \sum_i |KI_1(i) - KI_2(i)| = d_{KI}
 \end{aligned}$$



**Fig. 2** Example of *atomic change* of moving a leaf node  $L$  in a taxonomy  $G_1$  to  $b$  another parent in taxonomy  $G_2$ . **a**  $G_1$ : original taxonomy, **b**  $G_2$ : modified taxonomy

Thus,  $d_{GKSV} \geq d_{KI}$ . Hence, we have  $d_{KSV} \geq d_{GKSV} \geq d_{KI}$ . Since  $KGS$ ,  $GKSG$  and  $KIG$  are inversely proportional to  $d_{KSV}$ ,  $d_{GKSV}$  and  $d_{KI}$  (by definition), we obtain  $KIG(G_1, G_2) \geq GKSG(G_1, G_2) \geq KGS(G_1, G_2)$ .  $\square$

## 5 Analyzing similarity measures

We study the behaviour of our proposed similarity measures for comparing hierarchies, and explore the contributing factors. Specifically, we quantify the top  $k$  vertices and edges attributing to the similarity (or dissimilarity) between DAGs, thus providing interpretability.

### 5.1 Proximity of LCA

The proposed *Katz Graph Similarity Measure* (in Definition 3) inherently captures the desired properties of *Concept Hierarchy* and *Relationship Importance* via the Katz measure as discussed in Sect. 4. We now provide a mathematical analysis to show that our proposed measure also satisfies the *Proximity of Least Common Ancestor* property. For tractability of analysis, we consider:

1. The taxonomy under consideration is a directed tree instead of a DAG, so there is a unique parent–child relation and there is at most one path from one vertex to another.
2. In this scenario, we consider an atomic structural change (between two hierarchies) wherein a leaf node  $L$  is moved from its correct parent,  $TP$ , (considered as ground truth) to another (possibly false) parent,  $FP$  as depicted in Fig. 2.

Let  $LCA$  denote the *least common ancestor* of  $TP$  and  $FP$ , the parent nodes of  $L$  in the two taxonomies. Without loss of generality, considering  $TP$  to be constant, the different choices of  $FP$  as a new parent can be encoded in terms of 2 parameters:

- (1) The distance of  $LCA$  to  $TP$ , denoted as  $a$ , and
- (2) The distance from  $LCA$  to  $FP$ , denoted as  $b$ .

Observe that the above distances capture the notion of  $LCA$  proximity as described previously in Sect. 3. Hence, we have,

**Theorem 2** *The Katz similarity measure demonstrates monotonic behavior with respect to both the distances,  $a$  and  $b$ , for structural changes between two taxonomies modeled as tree structures.*

**Proof** Considering the example in Fig. 2, the only vertex between  $G_1$  and  $G_2$ , to undergo a change in its *Katz Similarity sub-vector* is the modified leaf vertex  $L$ , i.e., only  $KSV(\cdot, L)$  is affected. This is because, all other paths, except those ending at vertex  $L$ , remain intact between the graphs.

Note that since there is at most one path between the two vertices, if  $L$  is  $k$  hops away from a vertex  $u$ , the Katz similarity  $KS(u, L) = \alpha^k$ . The terms that contribute to the factor  $\|KSV_1 - KSV_2\|_1$  in Eq. (1) are those capturing the similarity between all other nodes and vertex  $L$  in  $G_1$  and  $G_2$ .

Only the vertices (in subtree of  $LCA$ ) that lie on the path from  $LCA$  to  $TP$  have a path to  $L$  in  $G_1$ , and are affected by the change in  $G_2$ . Hence, the difference in Katz similarity, induced by structural change, is,

$$\Delta_1 = \sum_{i=1}^a \alpha^i = \alpha(1 + \alpha + \dots + \alpha^{a-1}) = \frac{\alpha(1 - \alpha^a)}{1 - \alpha}$$

Similarly, vertices lying on path from  $LCA$  to  $FP$  have a path to  $L$  in  $G_2$ , and the associated difference in Katz similarity (to  $G_1$ ) is,

$$\Delta_2 = \sum_{i=1}^b \alpha^i = \alpha(1 + \alpha + \dots + \alpha^{b-1}) = \frac{\alpha(1 - \alpha^b)}{1 - \alpha}$$

Further, all vertices that lie between the *Root* and the  $LCA$ , also have a path to vertex  $L$ . Hence, the norm of the Katz vector difference for these vertices will depend on the greater of  $a$  and  $b$  values. Without loss of generality, assume  $a < b$ , i.e.,  $L$  is closer to  $LCA$  in  $G_1$  compared to  $G_2$ . Considering  $TP$  to be  $TPD$  hops away from *Root*,  $LCA$  becomes  $(TPD - a)$  hops away from the *Root*. Hence, the difference in Katz similarity for these vertices is,

$$\begin{aligned} \Delta_3 &= \sum_{i=0}^{TPD-a} (\alpha^{a+i+1} - \alpha^{b+i+1}) = (\alpha^{a+1} - \alpha^{b+1}) \sum_{i=0}^{TPD-a} \alpha^i \\ &= \frac{(\alpha^{a+1} - \alpha^{b+1})(1 - \alpha^{TPD-a+1})}{1 - \alpha} \end{aligned}$$



Thus, combining the above equations, the total difference in term  $||KSV_1 - KSV_2||_1$  for Katz graph similarity (Eq. 1) when  $a < b$  is,

$$\begin{aligned} \Delta &= \Delta_1 + \Delta_2 + \Delta_3 = \frac{\alpha(1 - \alpha^a + 1 - \alpha^b + (\alpha^a - \alpha^b)(1 - \alpha^{TPD-a+1}))}{1 - \alpha} \\ &= \frac{\alpha(2 - \alpha^{TPD+1} - \alpha^b[2 - \alpha^{TPD+1-a}])}{1 - \alpha} \end{aligned} \tag{8}$$

Similarly, for the case where  $a > b$ ,  $\Delta$  evaluates to,

$$\Delta = \frac{\alpha(2 + \alpha^{TPD+1} - \alpha^a[2 - \alpha^{TPD+1+b-2a}])}{1 - \alpha} \tag{9}$$

Assuming  $TPD \gg a, b$ , i.e., a displacement of a vertex occurs across similar depths within the taxonomy, Eq. (8) becomes,

$$\Delta = k_1 - \alpha^b k_2 \tag{10}$$

where  $k_1 = \frac{\alpha(2+\alpha^{TPD+1})}{1-\alpha}$  and  $k_2 = \frac{\alpha(2-\alpha^{TPD+1})}{1-\alpha}$  are constants as  $TPD$  is constant. Similarly, when  $a > b$ , Eq. (9) becomes,  $\Delta = k_3 - \alpha^a k_4$ , where  $k_3$  and  $k_4$  are constants. The above two cases can be combined into,

$$\Delta = K_1 - \alpha^{\max\{a,b\}} K_2 \quad [\text{for } TPD \gg a, b] \tag{11}$$

where  $K_1$  and  $K_2$  are constants that depend on the depth of the parent  $TP$  in the first taxonomy. From Eq. 11, we can observe that  $\Delta = ||KSV_1 - KSV_2||_1$  is monotonically non-decreasing with respect to both  $a$  and  $b$ , while one is kept constant. Hence, the *Katz Graph Similarity* between  $G_1$  and  $G_2$  given by Eq. (1) is also individually monotonic with respect to both  $a$  and  $b$ , and exhibits the *Proximity of LCA* property.  $\square$

### 5.2 Vertex attribution

To provide a notion of interpretability, as to why two hierarchies have been considered to be dissimilar by the measure, in this section, we quantify the role of differing vertex plays in the Katz similarity vector dissimilarity. Since, the amount of change influences the final similarity value, vertices with greater quantifiable change are more responsible for the dissimilarity between taxonomies. Thus,

**Definition 7** (*Vertex importance*) The importance of a vertex  $v$  with respect to its influence on the change in similarity between graphs  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  is defined by,

$$Imp(v) = ||KSV_1(:, v) - KSV_2(:, v)||_1 \tag{12}$$

where  $KSV_1$  and  $KSV_2$  are the Katz Similarity sub-vectors of vertex  $v$  in the two graphs respectively, and  $||\cdot||_1$  denotes the  $L_1$ -norm.

Note, the Katz Similarity sub-vectors can be replaced by the corresponding Katz grouped (or index) vectors, for the different variants. Also, note that these vectors and the difference i.e.,  $Imp(v)$  can be computed as a by-product of the similarity computation. Hence, the top- $k$  vertices with the highest *importance* values contributing the most for the dissimilarities between the hierarchies can be obtained in  $O(k \log V)$ .

### 5.3 Edge attribution

The contribution of an edge to the dissimilarities between linked structures is captured by its *centrality* in the graphs. Hence, the edge *importance* is the total Katz similarity that it contributes.

**Definition 8** (*Edge importance*) The importance of an edge  $(u, v)$  in a graph  $G = (V, E)$  is defined as,

$$Imp(e(u, v)) = \sum_l \sum_{\substack{\text{paths of length } l \\ \text{that contain edge } (u,v)}} \alpha^l \tag{13}$$

The above definition, provides the following interesting mathematical relation for efficient computation of the *edge importance*.

**Theorem 3** *The importance of edge  $(u, v)$ ,  $Imp(e(u, v))$ , in  $G = (V, E)$  is proportional to the product of the Katz Index of  $u$ ,  $KI(u)$ , and the reverse Katz Index of  $v$ ,  $RevKI(v)$ , i.e., the Katz index of  $v$  in graph  $G'$  obtained from  $G$  by reversing the edges.*

$$Imp(e(u, v)) = KI(u) \times \alpha \times RevKI(v) \tag{14}$$

**Proof** Every path including edge  $(u, v)$  can be viewed as a combination of a path ending at  $u$ , the edge  $(u, v)$ , and the remaining path from  $v$ . Subsequently, every path ending at  $u$  can be combined with every path beginning at  $v$  to obtain a unique path going through the edge  $(u, v)$ . Using Definition 8, we have,

$$\begin{aligned} Imp(e(u, v)) &= \sum_l \sum_{\substack{\text{paths of length } l \\ \text{that contain edge } (u,v)}} \alpha^l \\ &= \left( \sum_{l_1} \sum_{\substack{\text{paths of length } l_1 \\ \text{that end at } u}} \alpha^{l_1} \right) \times \alpha \times \left( \sum_{l_2} \sum_{\substack{\text{paths of length } l_2 \\ \text{that start at } v}} \alpha^{l_2} \right) = KI(u) \times \alpha \times RevKI(v) \end{aligned}$$

□

Observe, Katz and reverse Katz index for vertices in  $V$  of  $G$  can be computed in  $O(|E|)$  time. We can thus compare the centrality contribution for edges present in one of the DAGs, attributing the difference in similarity scores in  $O(k \log E)$  time.

**Table 1** Running times (in s) for different methods on taxonomies of varying sizes

Taxonomy	V	E	FM	node2vec	Katz Simil.	Grouped Katz@250	Katz Index
Bio Tx	338	346	0.03	10.80	0.01	0.03	0.01
Mobile Tech.	11,267	19,036	62.59	292.74	0.15	0.54	0.05
Climate Change	22,251	29,629	334.90	583.46	0.29	0.75	0.08
Domest. Animals	49,814	93,475	1923.84	1213.28	0.83	1.74	0.20
Aeronautics	174,896	323,596	45,518	4256	6.41	7.24	0.73
Ethnobiology	289,479	472,145	NA*	7332.26	7.76	10.14	1.09
Particle Physics	832,980	1,618,005	NA*	22,102	183.06	33.42	3.85
Physical Chem.	1,945,984	4,032,670	NA*	52,900	465.11	85.57	9.69
Computer Science	2,616,557	6,774,884	NA*	70,005	1,435	140.56	15.82
Death	3,643,987	10,283,884	NA*	NA*	2,979	253.57	26.85
DBpedia	6,535,441	27,065,747	NA*	NA*	OM <sup>+</sup>	1222	171.47

\*Killed after 24 h

<sup>+</sup>Out of memory

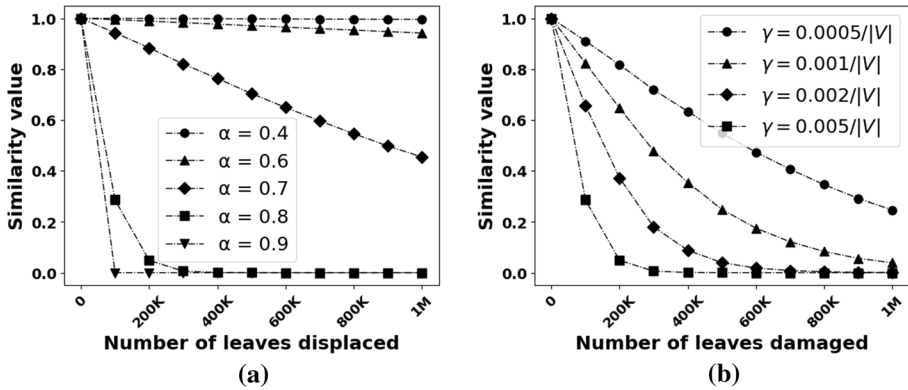
## 6 Experimental evaluation

In this section, we empirically validate the scalability and tunability of our proposed measures, and demonstrate that it conforms to the intuitive properties and theoretical analysis presented previously. We perform experiments on different sub-hierarchies of the DBpedia taxonomy with varying the number of concept nodes.<sup>4</sup> We also use the real-life biological plant kingdom hierarchy for our experimental setup. The characteristics of the datasets can be observed in Table 1. Since cycles are inconsistent with the logical usage of taxonomies (modeling broader-specific relation among nodes), we remove cycles as a pre-processing step using DFS-based technique in line with Suominen and Hyvönen (2012), to obtain a DAG. We note that alternative cycle removal techniques (e.g., Sun et al. 2017) can also be used.

*Tunability and parameter setting* One key aspect of our proposed measures is the tunability for capturing different notions of similarity for diverse applications. The parameters  $\alpha$  and  $\gamma$  in the Katz Similarity measure model the degree of structural and semantic differences between two hierarchies that might be tolerable. We use the *Death* taxonomy and Group Katz similarity measure with 250 groups.

The parameter  $\alpha$  controls the decay of similarity between two vertices as their path length increases, modeling importance of relationships. A higher value of  $\alpha$  increases the influence of distant ancestors to structural changes, while setting  $\alpha = 1$  treats every path between two vertices as same, irrespective of its length, and is similar to measuring the transitive closure (multiple paths added with equal weights). Figure 3a shows the behaviour of different  $\alpha$  values for increasing number of displaced leaves with ( $a = 2, b = 1$ ) and  $\gamma = 0.005/|V|$ .

<sup>4</sup> DBpedia, derived from Wikipedia pages, contains over 6 million vertices and 25 million edges for a hierarchical ordering of Wikipedia concepts using the *Simple Knowledge Organization System* (SKOS) broader (i.e., skos:broader) relationships.



**Fig. 3** Variation of parameters for similarity measures. **a** Varying  $\alpha$  with  $\gamma = 0.005/|V|$ , **b** varying  $\gamma$  with  $\alpha = 0.8$

The parameter  $\gamma$ , on the other hand, controls the sensitivity of the measures to structural differences between taxonomies, and is a normalizing factor characterizing the similarity score to differences in the Katz similarity vectors. Figure 3b exhibits the similarity scores for different values of  $\gamma$ , with  $\alpha = 0.8$ . Observe that, setting  $\gamma = 0.005/|V|$  or to  $0.002/|V|$ , tunes the sensitivity of the measure to report a similarity score of 0 when the structural difference is more than 300K or 700K nodes respectively, depending on application.

Table 1 lists the top-level sub-concepts of the taxonomies and their characteristics used for our evaluations. The sub-taxonomies were derived by choosing a sub-concept node and considering the sub-graph induced by its descendants. For the remaining sections, the parameters  $\alpha$  and  $\gamma$  were set to their default value of 0.8 and  $0.001/|V|$  respectively, while for the *Grouped Katz Similarity* measure, the vertices were randomly split into  $G = 250$  groups. All algorithms were implemented in C and run-times reported for Intel(R) Xeon(R) E5-2470 processor with 150 GB memory.

We next benchmark the performance and the different features of proposed measure against state-of-the-art *node2vec* embedding technique and the *Fowlkes-Mallows* (FM) measure. For the *node2vec* approach, the vector representation of each node (based on random walk) in both the taxonomic structures were constructed. The cosine similarity between the vectors were then added across all the nodes to obtain the similarity score between the hierarchies. An open-source implementation for *node2vec* was obtained from <http://github.com/palash1992/GEM> (Goyal and Ferrara 2018).

### 6.1 Qualitative analysis

In this section, we study the qualitative performance of the competing approaches based on the intuitive features that a similarity measure should demonstrate. We use the “Mobile Tech.” sub-concept from the DBpedia hierarchy. Further, we also use a real-life Biological taxonomy on plant kingdom classification (named *dwca – census\_plants\_pahou – v1.2*) obtained from [www.gbif.org/](http://www.gbif.org/).

#### 6.1.1 Concept hierarchy awareness

To understand the degree of sensitivity of the competing measures to the input concept hierarchy structure, we perturb the original hierarchy by flipping (i.e., reversing) the

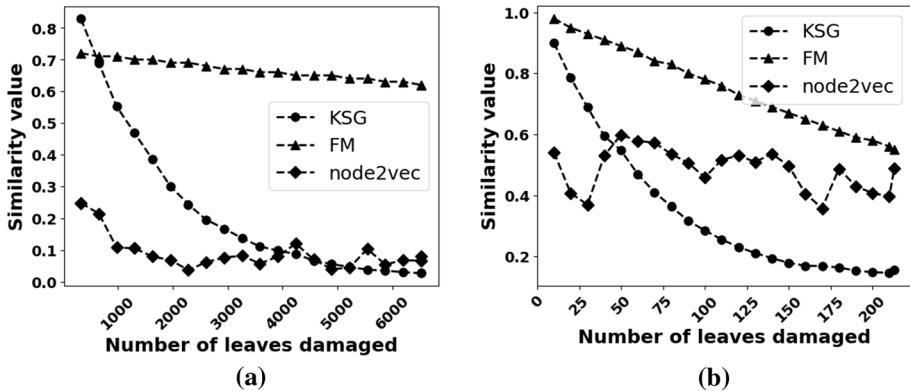


Fig. 4 Awareness of concept hierarchy on **a** Mobile Tech. in DBpedia and **b** biological plant kingdom

directions of increasing number of parent–child edges (at the level of the leaves). With this increasing change in the logical subsumption of concept, from Fig. 4a, we observe that both the FM and the Grouped Katz index similarity measure value decreases with increase in the damage of the Mobile Tech. hierarchy from DBpedia. However, the FM measure shows a slow decrease in its similarity value, while our measure demonstrates a steeper decrease. We argue that this better captures the logical difference between the hierarchies, as with increase in reversal of edge directions, the quality of the taxonomy decreases dramatically (and not a slow degradation as captured by FM). On the other hand, we find that the “node2vec” approach exhibits nearly constant similarity with some random spikes due to the effect of the inherent random walk procedure. Similar results were also obtained on the Bio-taxonomy dataset, as shown in Fig. 4b.

### 6.1.2 Capture of logical subsumption

We now consider how the competing approaches behave under the presence of various types of structural difference in the taxonomies. We broadly consider the three different types of deviation in the logical subsumption within the hierarchies as shown by the caricature example of Fig. 1 in Sect. 3. Specifically, a particular concept vertex is dislodged from its current position and is attached to: (1) sibling of parent, (2) sibling of grandparent, and (3) to another concept at the same level as that of its parent, but in a different sub-hierarchy with respect to its grandparent. These perturbation (referred to as Taxonomy A, B, and C respectively) have been intuitive argued (in Sect. 3) to demonstrate increasing degrees of degradation in the logical subsumption and semantic integrity of the hierarchy. Hence, the similarity values for the three perturbed hierarchy when compared to the original taxonomy should monotonically decrease.

We induce the above three structural perturbations to the Mobile Tech. and Bio-Plant taxonomy, vary the number of vertices taking part in such perturbations, and compute the similarity measures with respect to the original unchanged taxonomy. From Fig. 5, we observe that the behavior for both the FM measure and our proposed Katz Graph Similarity measure is persistent with the properties of Sect. 3—that is, the similarity decreases as we move from perturbation type A to that of C (Fig. 1). This also captures and models the property of *Proximity to LCA*. However, we again observe that the performance

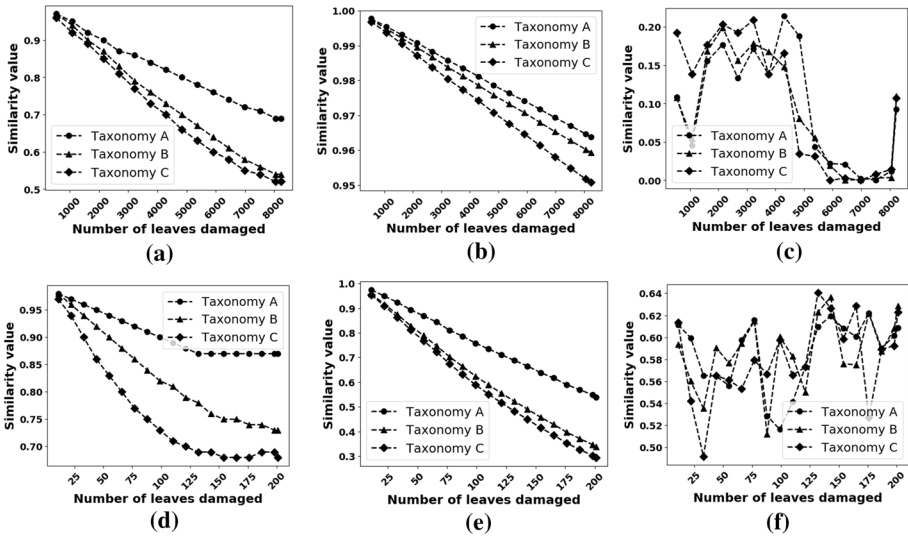


Fig. 5 Logical subsumption property on different taxonomic perturbations for the two datasets

of the embedding technique rapidly oscillates and behaves randomly with respect to the taxonomic “damage type”.

### 6.1.3 Closeness to LCA: empirical validation

Next, we empirically show the *monotonicity property* of our proposed measures, with respect to distances  $a$  and  $b$  capturing the *Proximity of LCA* (of Sect. 5.1). We demonstrate the effect for a general DAG structure instead of a directed tree, as assumed in our analysis. However, unlike in a tree, the least common ancestor (*LCA*) of two nodes in a DAG is not uniquely defined. Hence, we define the *LCA* of a vertex present in two DAGs as the ancestor having the shortest path to the vertex in one of the DAGs (i.e., to *TP* for example in Fig. 2).

For evaluation, we take the *Death* taxonomy derived from DBpedia (Table 1) and generate the comparing taxonomy by inducing structural “damages” by—taking a leaf node, detaching it from its original parent and reattach it as a child of another node at a distance of  $(a, b)$ .

Figure 6a, b depicts the similarity between the modified taxonomy and the original taxonomy with increasing number of induced structural differences. As presented in Sect. 5.1, we observe monotonic decrease in the similarity score as the distance values of  $a$  and  $b$  are increased. Further, the score decreases as the number of structural differences are increased. This monotonic behaviour is depicted by all our measures, conforming to Theorem 2.

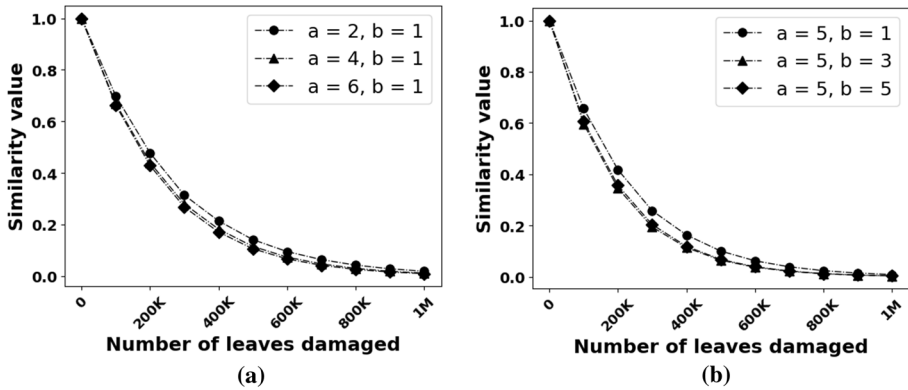


Fig. 6 Variation in Katz similarity measures with increasing distance of induced structural differences. **a** Varying distance  $a$  ( $b$  fixed), **b** varying distance  $b$  ( $a$  fixed)

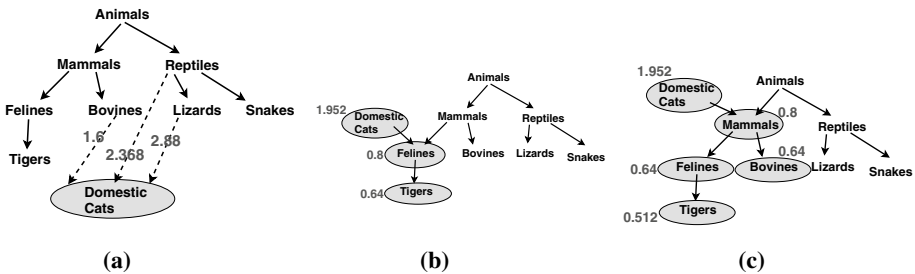


Fig. 7 Vertex Importance of concept vertices for **a** Caricature example perturbations, **b, c** other perturbations

### 6.1.4 Interpretability: vertex importance

A key feature of our proposed measure in the concept of *interpretability*, wherein structural or logical difference between hierarchies, that play a key part in their dissimilarities, can be identified. In fact, such differences in modeling parent–child relationships and/or logical and semantic binding of concepts may indicate key insights or as well be presented to domain-experts for evaluation.

*DBpedia subhierarchy experiment* We compute the vertex importance (as shown in Definition 7) for the vertices of the input taxonomies, with an induced perturbation where the edges between leaves and their parents were reversed. It was interesting to observe that exactly the vertices affected by the perturbation (i.e., the parent and the corresponding leaves) received a positive  $Imp()$  score, while the score remained 0 for all other vertices.

*Caricature taxonomy experiment* Consider the perturbations of the caricature example in Sect. 3, we observe an increase in the *Vertex Importance* score of the concept node with increase in the logical difference between the compared hierarchies—as shown in Fig. 7a.

For other diverse changes in the hierarchy structures, as shown in Fig. 7b, c wherein the structural change does not occur at the leaves, we observe the concept nodes with shorter connected paths (to the “damaged” vertex) demonstrating a higher vertex importance score, which percolates down the children (albeit with a lower score) to capture the effect of change in logical subsumption of the hierarchy due to the structural change.

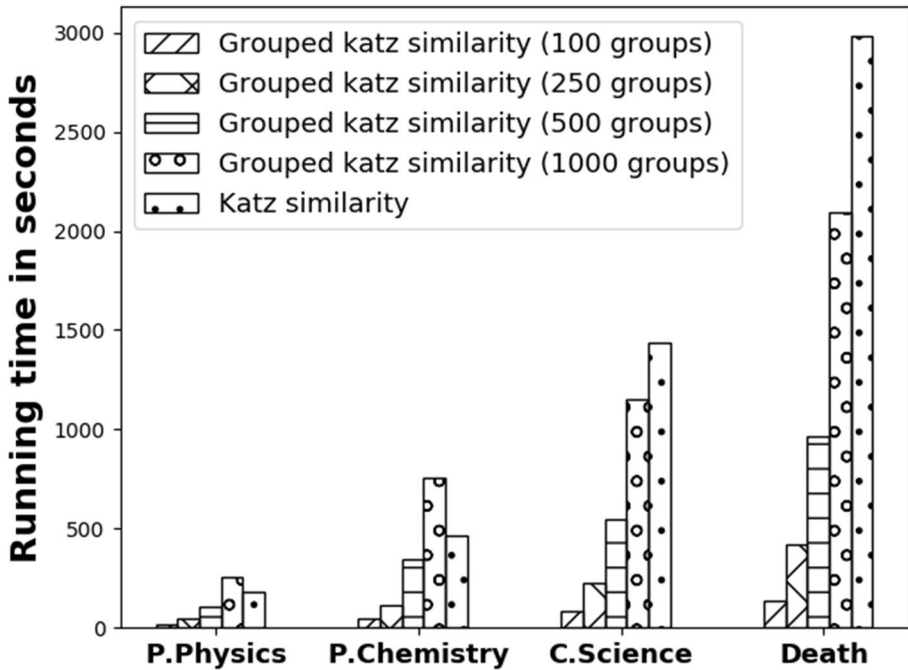


Fig. 8 Variation of the run-time for grouped Katz similarity

This is indeed a key novelty provided by our measure, and enables it to identify the probable key differences between the hierarchies, providing insights into the logical difference between them.

## 6.2 Scalability studies

We demonstrate the scalability of our proposed measures against state-of-the-art *Fowlkes-Mallows* (FM) measure and *node2vec* based similarity, by comparing the time taken by the competing approaches to compute the similarity scores between the input taxonomy and itself.

Table 1 tabulates the time taken by the *FM*, *node2vec*, *Katz Similarity*, *Grouped Katz Similarity*, and *Katz Index Similarity* measures. We observe that the compute time of FM measure is manageable for smaller taxonomies (with  $< 100,000$  vertices), but quickly escalates into multiple hours even on the medium sized *Aeronautics taxonomy* with 167,297 vertices. The FM measure takes more than 11 h to complete, while the Katz similarity measure incurs less than 4 s, providing an improvement of around  $10,000\times$ . The *node2vec* approach is tractable in the sense that it runs to completion for large taxonomies (where FM fails). However, the Katz similarity demonstrates nearly  $70\times$  run-time improvements for larger hierarchies like *Computer Science*, and shows nearly  $7000\times$  speedup for the proposed Katz index technique.



The Katz similarity measure relies on the computation of path distances between every pair of reachable vertices, and hence for huge taxonomies might be practically inefficient. Hence, the more compute efficient Grouped Katz similarity measure and the Katz Index measure provide approximations to the Katz similarity measure in such scenarios. Figure 8 shows the running time for Grouped Katz similarity with varying group sizes on four large taxonomies from Table 1. For example, on the *Death* sub-hierarchy (with 3.6 million concepts), the Katz similarity and Grouped Katz similarity takes around 50 and 4 min respectively, while the FM measure was terminated after 24 h. Even for the whole of DBpedia, the Grouped Katz similarity took only around 20 min, showcasing extreme scalability in gracefully handling Web-scale taxonomies.

Intuitively, the run-time of Grouped Katz measure increases with increase in the number of groups (shown in Fig. 8), as the group-vertex similarity computation increases. Interestingly, we observed that the run-time for Katz Similarity measure might not always be more than the Group Katz measure for various group sizes, as Grouped Katz computes a dense matrix of size  $|V| \times \mathcal{G}$ , and  $\mathcal{G}$  can be greater than the number of reachable vertex pairs in some cases. For example, Grouped Katz with 1000 groups was more compute intensive than the Katz similarity on *Physical Chemistry* taxonomy.

### 6.2.1 Discussion

Hence, we observe that our proposed similarity measures are indeed adept in capturing the structural similarities along with incorporating the property of logical subsumption of concepts (transitivity-aware) in DAGs or other hierarchical structures. We show that our measure respects all the intuitive properties of a similarity measure in this problem domain. Further, our measure demonstrates the vital features of interpretability, tunability, and scalability making it far superior as compared to other existing approaches.

## 6.3 Further analysis

### 6.3.1 Loss of structural information from Katz similarity to grouped variant

As we have observed in Fig. 6, Katz similarity captures the logical subsumption among concepts. However, as we go from Katz similarity to the Grouped version, not all structural changes might be captured by the Grouped variant (and the special case of Katz Index). For example, in Taxonomy X (Fig. 1a) consider parents of concepts *Lizards* and *Bovines* to be interchanged (i.e., *Mammals*  $\rightarrow$  *Lizards* and *Reptiles*  $\rightarrow$  *Bovines*). Katz Index only considers the path length incident on a vertex, and not the origin or path vertices. Since our modified taxonomy still preserves the overall incident structure (i.e., number of paths ending) of the vertices, the Katz Index remains unchanged. Contrarily, the Katz similarity measure accounts for this difference by considering the entire path (i.e., in Taxonomy X, path of length 1 incident on *Bovines* originated from *Mammals*, while in the new taxonomy it originated from *Reptiles*).

This loss of structural information occurs as it is oblivious to paths coming from different vertices only if they originate from vertices within the same group. Thus, the more the number of groups, the closer Grouped Katz Similarity is to the Katz Similarity measure

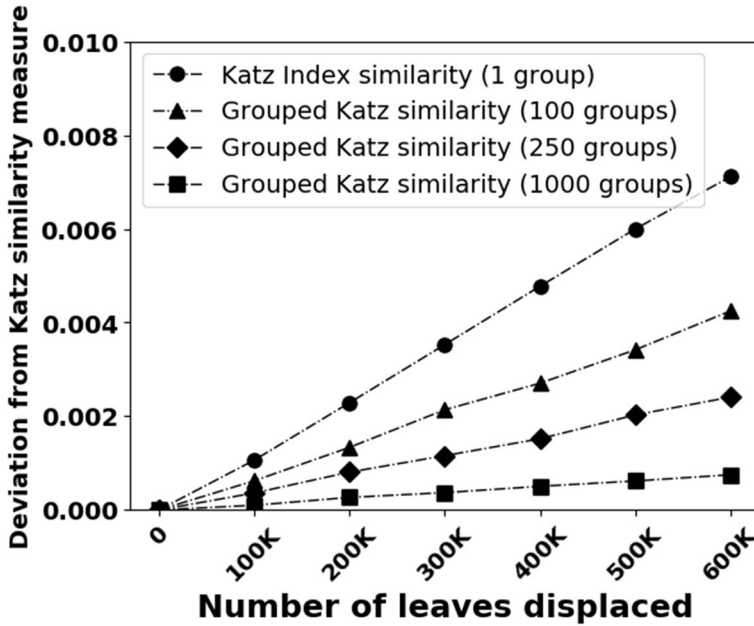
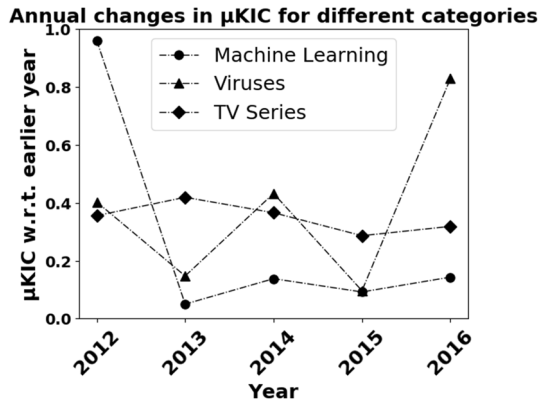


Fig. 9 Deviation in group Katz similarity compared to the Katz similarity measure with varying group sizes

Fig. 10  $\mu KIC$  for different categories



(Fig. 9). This corresponds to the bounds on the Grouped Katz similarity, as shown in Theorem 1.

### 6.3.2 Assessing temporal evolution of categories

Next, we consider the extreme case where we assess how well we can identify the relative growth of different subject areas in the knowledge hierarchies merely based on our graph structure comparison measure, independent of any syntactic or embedding feature. Specifically, we capture:

- (1) *emergence* of new concepts within a category, and
- (2) *disruptive evolution* of categories.

We consider snapshots of categories in DBpedia for years 2011–2016. The change in category,  $C$ , from year  $y_1$  to  $y_2$  is measured by the micro-averaged Katz index change,  $\mu KIC$ , defined as:

$$\mu KIC = \sum_{v_i \in C} |KI_{i,y_1} - KI_{i,y_2}| / |C| * KI_{i,y_1}$$

with nodes  $i$  computed by BFS from category node to leaves.

Figure 10 depicts that the field of “Machine Learning” underwent a major disruption in 2012, which corresponds to the development of deep learning techniques in this area. Similarly, the organization of “Viruses” category underwent major changes in 2012, 2014 and 2016, with the outbreak of H1N1, Ebola and Zika virus strains. In contrast, the field of “Television Series” doesn’t witness sudden radical changes, but rather undergoes smooth regular changes.

Hence, the temporal evolution of categories as captured by our measure seems to correspond well with the known disruptive emergence and evolution of concepts in those areas.

## 7 Conclusion

This paper proposed principled and scalable similarity measures, adapting the Katz similarity, for comparing DAGs. We identified key properties that a similarity measure for knowledge hierarchies should capture, and provided a theoretical analysis depicting that our measures capture the structure and logical subsumption of concept relations in these hierarchies. We also presented a linear time variant, to cater to various real-world applications, and empirically showed that our measures are scalable, efficient (with upto  $10,000 \times$  run-time improvements) and tunable. We also demonstrated that our measure depicts interpretability, and identifies the precise region that contributes to the semantic and logical differences between hierarchies. Furthermore, we showed that the temporal evolution of different subgraphs, as captured by our measure, corresponds well with known disruptions in the related subject areas. Future work involves deriving approximation bounds for Grouped Katz similarity and comparing different knowledge hierarchies with varying granularity within the proposed framework. The similarity measures proposed in this paper have been implemented in C and the code for the same is available at <https://github.com/guruprasadnk7/DAGSimilarityKatz>.

## References

- Adams, E. I. I. (1972). Consensus techniques and the comparison of taxonomic trees. *Systematic Zoology*, 21(4), 390–397.
- Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1), 217–239.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on management of data, SIGMOD '08* (pp. 1247–1250). ISBN 978-1-60558-102-6.
- Bordea, G., Buitelaar, P., Feralli, S., & Navigli, R. (2015). Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *SemEval-2015*.

- Bordea, G., Lefever, E., & Buitelaar, P. (2016). Semeval-2016 task 13: Taxonomy extraction evaluation (text-eval-2). In *SemEval-2016* (pp. 1081–1091).
- Brandenburg, F., Gleiner, A., & Hofmeier, A. (2012). Comparing and aggregating partial orders with Kendall tau distances. In *WALCOM: Algorithms and computation* (pp. 88–99).
- Brank, J., Grobelnik, M., & Mladenić, D. (2005). A survey of ontology evaluation techniques. In *SiKDD*.
- Brank, J., Mladenić, D., & Grobelnik, M. (2006). Gold standard based ontology evaluation using instance assignment. In *Workshop on evaluation of ontologies for the web, EON*.
- Cai, S., & Knight, K. (2013). Smatch: An evaluation metric for semantic feature structures. In *Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 748–752). Association for Computational Linguistics.
- Cross, V., Yu, X., & Hu, X. (2013). Unifying ontological similarity measures: A theoretical and empirical investigation. *International Journal of Approximate Reasoning*, 54(7), 861–875.
- d'Aquin, M. (2009). Formally measuring agreement and disagreement in ontologies. In *K-CAP* (pp. 145–152).
- David, J., Euzenat, J., & Šváb-Zamazal, O. (2010). Ontology similarity in the alignment space. In *ISWC* (pp. 129–144).
- de Knijff, J., Meijer, K., Frasincaer, F., & Hogenboom, F. (2011). Word sense disambiguation for automatic taxonomy construction from text-based web corpora. In *WISE*.
- Dellschaft, K., Staab, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In *International semantic web conference* (pp. 228–241).
- Elghawalby, H., & Hancock, E. R. (2008). Measuring graph similarity using spectral geometry. In *ICIAR* (pp. 517–526).
- Fensel, D. (2005). *Spinning the semantic web: Bringing the World Wide Web to its full potential*. Cambridge: MIT Press.
- Foggia, P., Percannella, G., & Vento, M. (2014). Graph matching and learning in pattern recognition in the last 10 years. *Pattern Recognition and AI*, 28(01), 1450001.
- Fowlkes, E., & Mallows, C. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78, 553–569.
- Gao, X., Xiao, B., Tao, D., & Li, X. (2010). A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1), 113–129.
- Geffert, M., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *ACL* (pp. 107–114).
- Goyal, P., & Ferrara, E. (2017). *Graph embedding techniques, applications, and performance: A survey*. [arXiv:1705.02801v4](https://arxiv.org/abs/1705.02801v4) [cs.SI].
- Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151, 78–94.
- Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with ontoclean. *Communications of ACM*, 45, 61–65.
- Harabagiu, S. M., Maiorano, S. J., & Paşca, M. A. (2003). Open-domain textual question answering techniques. *Natural Language Engineering*, 9(3), 231–267.
- Hulpuş, I., Prangnawarat, N., & Hayes, C. (2015). Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *ISWC* (pp. 442–457).
- Jurgens, D., & Pilehvar, M. T. (2016). Semeval-2016: Semantic taxonomy enrichment. In *SemEval* (pp. 1092–1102).
- Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1), 39–43.
- Koutra, D., Ke, T.-Y., Kang, U., Chau, D., Pao, H.-K., & Faloutsos, C. (2011). Unifying guilt-by-association approaches: Theorems and fast algorithms. In *ECML-PKDD* (pp. 245–260).
- Koutra, D., Shah, N., Vogelstein, J. T., Gallagher, B., & Faloutsos, C. (2016). Deltacon: A principled massive-graph similarity function with attribution. *TKDD*, 10(3), 28.
- Kozareva, Z., & Hovy, E. (2010). A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1110–1118). Association for Computational Linguistics.
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., et al. (2015). Dbpedia: A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2), 167–195.
- Levin, R., Abassi, H., & Cohen, U. (2016). Guided walk: A scalable recommendation algorithm for complex heterogeneous social networks. In *Recommender systems* (pp. 293–300).
- Liu, X., Song, Y., Liu, S., & Wang, H. (2012). Automatic taxonomy construction from keywords. In *KDD* (pp. 1433–1441).
- Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. In *EKAW* (pp. 251–263).

- Malmi, E., Tatti, N., & Gionis, A. (2015). Beyond rankings: Comparing directed acyclic graphs. *Data Mining and Knowledge Discovery*, 29(5), 1233–1257.
- McVicar, M., Sach, B., Mesnage, C., Lijffijt, J., Spyropoulou, E., & De Bie, T. (2016). Sumoted: An intuitive edit distance between rooted unordered uniquely-labelled trees. *Pattern Recognition Letters*, 79, 52–59.
- Newman, M. (2010). *Networks: An introduction*. Oxford: Oxford University Press.
- Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. In *KDD* (pp. 1105–1114).
- Pesquita, C., Faria, D., Falco, A. O., Lord, P., & Couto, F. M. (2009). Semantic similarity in biomedical ontologies. *PLOS Computational Biology*, 5(7), 1–12.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *JASA*, 66(336), 846–850.
- Resnik, P. (1995). Using information content to evaluate semantic similarity in taxonomy. In *IJCAI* (pp. 448–453).
- Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., & Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. In *AI and Statistics* (pp. 488–495).
- Suchanek, F. M., Kasneci, G., & Weikum, G. (2008). YAGO: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3), 203–217.
- Sun, J., Ajwani, D., Nicholson, P. K., Sala, A., & Parthasarathy, S. (2017). Breaking cycles in noisy hierarchies. In *WebSci* (pp. 151–160).
- Suominen, O., & Hyvönen, E. (2012). Improving the quality of SKOS vocabularies with skosify. In *EKAW* (pp. 383–397).
- Vedula, N., Nicholson, P. K., Ajwani, D., Dutta, S., Sala, A., & Parthasarathy, S. (2018). Enriching taxonomies with functional domain knowledge. In *SIGIR*.
- Velardi, P., Navigli, R., Faralli, S., & Maria Ruiz-Martínez, J. (2012). A new method for evaluating automatically learned terminological taxonomies. In *LREC* (pp. 1498–1504).
- Velardi, P., Faralli, S., & Navigli, R. (2013). Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3), 665–707.
- Volker, J., Vrandečić, J., Sure, J., & Hotho, A. (2010). AEON: An approach to the automatic evaluation of ontologies. *Journal of Applied Ontology*, 3, 41–62.
- Wagner, S., & Wagner, D. (2007). Comparing clusterings: An overview. *Analysis*, 4769, 1–19.
- Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *ACL* (pp. 133–138).
- Zavitsanos, E., Paliouras, G., & Vouros, G. A. (2011). Gold standard evaluation of ontology learning methods through ontology transformation and alignment. *TKDE*, 23, 1635–1648.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Guruprasad Nayak<sup>1</sup> · Sourav Dutta<sup>2</sup>  · Deepak Ajwani<sup>2</sup> · Patrick Nicholson<sup>2</sup> · Alessandra Sala<sup>2</sup>

Guruprasad Nayak  
nayak013@umn.edu

Deepak Ajwani  
deepak.ajwani@nokia-bell-labs.com

Patrick Nicholson  
pat.nicholson@nokia-bell-labs.com

Alessandra Sala  
alessandra.sala@nokia-bell-labs.com

<sup>1</sup> University of Minnesota, Minneapolis, USA

<sup>2</sup> Nokia Bell Labs, Blanchardstown, Ireland