



Title	Automated assessment of knowledge hierarchy evolution: comparing directed acyclic graphs
Authors(s)	Nayak, Guruprasad, Dutta, Sourav, Ajwani, Deepak, et al.
Publication date	2018-12-17
Publication information	Nayak, Guruprasad, Sourav Dutta, Deepak Ajwani, and et al. "Automated Assessment of Knowledge Hierarchy Evolution: Comparing Directed Acyclic Graphs." Springer, December 17, 2018. https://doi.org/10.1007/s10791-018-9345-y .
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/9891
Publisher's statement	This is a post-peer-review, pre-copyedit version of an article published in Information Retrieval Journal. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10791-018-9345-y .
Publisher's version (DOI)	10.1007/s10791-018-9345-y

Downloaded 2026-05-02 00:30:01

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

[Click here to view linked References](#)

Noname manuscript No. (will be inserted by the editor)
--

Automated Assessment of Knowledge Hierarchy Evolution: *Comparing Directed Acyclic Graphs*

Guruprasad Nayak · Sourav Dutta · Deepak
Ajwani · Patrick Nicholson · Alessandra Sala

Received: date / Accepted: date

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

G. Nayak
University of Minnesota, USA
E-mail: nayak013@umn.edu

S. Dutta ✉, D. Ajwani, P. Nicholson, A. Sala
Nokia Bell Labs, Ireland
E-mail: {sourav.dutta, deepak.ajwani, pat.nicholson, alessandra.sala}@nokia-bell-labs.com

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; Geffet and Dagan, 2005; Harabagiu et al., 2003; Maedche and Staab, 2002). 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).

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 (de Knijff et al., 2011; Liu et al., 2012; Vedula et al., 2018). Modern linked open data repositories like the DBpedia taxonomy¹ (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 semantic subsumption just based on

¹ 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.

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

10
11 In this paper, we explore if there is a scalable, tunable and interpretable graph
12 similarity measure that can be leveraged to quantify the changes in the knowledge
13 hierarchy. To this end, we identify a set of key characteristic properties that a graph
14 similarity measure indicating semantic subsumption should possess. We then show
15 that a measure based on a group Katz similarity satisfies these requirements and is a
16 good indicator of logical subsumption change happening in a knowledge hierarchy.
17 Then, we go to the extreme and assess how well we can identify the relative growth
18 of different subject areas in the knowledge hierarchies merely based on our graph
19 structure comparison measure, independent of any syntactic or embedding feature.
20 We show that the proposed measure is able to identify key changes in the organization
21 of popular subject areas in the DBpedia taxonomy and these changes match well with
22 external events.

23
24
25 **State-of-the-art.** The literature hosts a variety of techniques for assessing the qual-
26 ity of taxonomies and evaluating their similarities with other structures, and can be
27 broadly categorized as (Brank et al., 2006):

28 • Traditionally, the generated hierarchies are presented to domain experts for
29 manual evaluation and curation. However, such techniques are practically infeasible
30 for modern taxonomies comprising millions of concepts and relations.

31 • Qualitative indicators compare taxonomies based on structural overlap, accu-
32 racy, consistency and rigidity (Volker et al., 2010). However, differences in structure
33 and scope might stem from input data, domain of interest and granularity of appli-
34 cation. Further, they fail to capture transitive closure or logical subsumption of hy-
35 ponym relations.

36 • Automated measures for evaluating taxonomies against a gold standard tax-
37 onomy pose a rigid domain dependency, and are hence unable to provide a generic
38 concept of similarity. For example, the concept ``chiaroscuro`` is categorized
39 under *picture, image, icon* (along with concept “collage”) in WordNet, while it falls
40 under *perspective and shading technique* category in Art and Architecture Thesaurus
41 (with “collage” in *image making processes and techniques*) (Velardi et al., 2012).

42 • Similarity measures for tree structures or undirected graphs (Koutra et al., 2016;
43 McVicar et al., 2016) are unable to model the structural complexity (like multiple
44 parents) within DAGs or consider the semantic coherence of parent-child links that
45 characterizes hierarchical knowledge sources.

46 Further, none of the above approaches are scalable in practice for comparing huge
47 taxonomies, and also do not provide tunability to characterize the degree of simi-
48 larity for different scenarios. Also, most existing measures require many hours or
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 days of computation on a typical workstation, even for small sub-graphs of DBpedia
2 taxonomy. In contrast, we aim for a measure that is significantly faster and scalable.
3

4 **Contributions.** This paper proposes a novel similarity measure for comparing non-
5 specialized hierarchically linked knowledge structures in a principled and scalable
6 fashion. We show that our proposed measure exhibits the essential properties to cap-
7 ture structural similarity as well as logical coherence and subsumption in directed
8 acyclic hierarchies. We also demonstrate that the measure is extremely scalable for
9 massive taxonomies and tunable across diverse applications. In a nutshell, the contri-
10 butions of this paper are:

11 • A generalized and scalable measure, based on weighted connected path count,
12 to automatically compute the similarity between taxonomies (and directed acyclic
13 graphs), capturing both the structural similarity and the logical subsumption of con-
14 cepts. Particularly, we focus on the scenario of comparing and evaluating taxonomies
15 in an automated fashion.

16 • Theoretical analysis to demonstrate that the performance of our similarity mea-
17 sure conforms to commonsense and intuitive properties.

18 • A linear time variant of the measure for practical applications.

19 • Detailed experimental evaluations on the desired properties of a similarity mea-
20 sure, and the novel concept of *interpretability*.

21 • Experimental validation on large real linked knowledge repositories to show-
22 case improved scalability and tunability of our measure.

23 Furthermore, the proposed technique can be generalized beyond taxonomies as a
24 measure for computing similarities between Directed Acyclic Graphs (DAGs).
25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

##

1 while *semantic similarity* measures consider the depth of the least common ancestor
2 to judge the goodness of node addition in existing hierarchies (Jurgens and Pilehvar,
3 2016; Wu and Palmer, 1994). However, this only provides “in-vitro comparison” of
4 taxonomy enrichment, and none of the above measures capture the relational sub-
5 sumption of concept linking in hierarchies. Measuring the similarities in language
6 parse trees have been studied based on edge overlap by (Cai and Knight, 2013).
7

8 Among state-of-the-art measures for taxonomy comparison (Bordea et al., 2015,
9 2016), the most commonly used is the *Fowlkes-Mallows* (FM) measure (Fowlkes and
10 Mallows, 1983; Velardi et al., 2013; Wagner and Wagner, 2007). It computes a hier-
11 archical graph partitioning (based on vertex cuts with possible overlaps) of concepts.
12 The taxonomies are then compared based on the agreement on the allocation of ev-
13 ery vertex-pair to the same partition, with associated path lengths assigned weights.
14 However, the *FM* measure suffers from 4 key drawbacks – (1) *dependence on hi-*
15 *erarchy of vertices to capture graph structure (uniquely defined only for trees) and*
16 *hence weights assignment is ad-hoc*; (2) *not scalable as it involves vertex partition*
17 *at each level of the hierarchy*; (3) *not tunable to suit the sensitivity needs of different*
18 *applications*; and (4) *lacks interpretability*.
19

20 A parallel body of related research includes *tree* and *graph similarity* measures.
21 However, it mostly involves capturing graph similarities with unlabeled nodes and
22 undirected edges (Elghawalby and Hancock, 2008; Foggia et al., 2014). Recent graph
23 embedding approaches (Goyal and Ferrara, 2017, 2018) have been proposed for com-
24 parison based on high-dimensional representation, but such techniques suffer from
25 scalability issues. Supervised methods using *guided walk* on graph (Levin et al.,
26 2016) have been studied to obtain similarities for recommendations – but requires
27 large training data.

28 In (Shervashidze et al., 2009), the authors proposed a scalable graph kernel ap-
29 proach based on the frequency of subgraph patterns called *graphlets*. Recently, Delta-
30 Con (Koutra et al., 2016), a scalable similarity measure was proposed for undirected
31 labeled graphs using pairwise node similarities, along with a faster approximate ver-
32 sion using belief propagation (Koutra et al., 2011). Kendall-Tau based distance mea-
33 sures were also proposed (Brandenburg et al., 2012) for comparing partial orderings.
34 However such approaches consider undirected graphs or unlabeled nodes, and fail
35 to model the structure of linked hierarchies (e.g., taxonomies) represented as DAGs
36 with labeled nodes.

37 Recently, a similarity measure for DAGs comparing partial rankings, based on the
38 edge set of the DAGs, assigns different weights to different kinds of edge differences
39 was proposed (Malmi et al., 2015). However, it ignores paths between nodes (uses
40 direct relations) thus failing to capture multi-hop semantic subsumption present in
41 hierarchies. Other distance metrics like *tree edit* (Bille, 2005) and *graph edit* dis-
42 tances (Gao et al., 2010) were also explored. A graph similarity approach comparing
43 taxonomies by defining edit distance on taxonomies was proposed (McVicar et al.,
44 2016). However, the input taxonomies were assumed to be trees, which is not gener-
45 ally true (Bordea et al., 2016; Kozareva and Hovy, 2010; Velardi et al., 2013). Also,
46 it suffers from quadratic time complexity and hence is not scalable.
47

48 Similarity measures like *Tversky’s parameterized ratio model* (Cross et al., 2013)
49 for comparing ontologies are sensitive to parameter setting and/or involve compu-
50

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

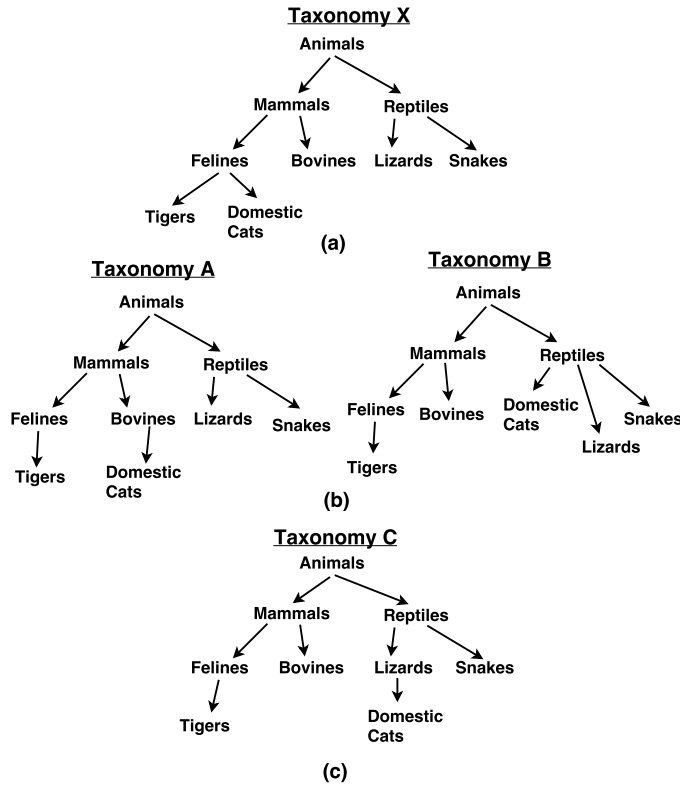


Fig. 1 Depiction for Properties of Similarity Measure

tation of Information Content (IC) of concepts, making it practically infeasible for large hierarchies. Other approaches in this area (d'Aquin, 2009; David et al., 2010) 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

1 parent-child edge depicts two related concepts, with the parent representing a broader
2 category. For example, in Figure 1(a), the edge $Mammals \rightarrow Felines$ models the
3 relation that *Felines* are a sub-type of *Mammals*.

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

10
11 **(2) Proximity of Least Common Ancestor:** Given two structurally different tax-
12 onomies, it may not be possible to completely characterize the effects of the differ-
13 ences on their similarity. However, analytically we can argue about the behaviour that
14 a similarity measure should demonstrate for varying degrees of dissimilarity.

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

22 The above notion of semantic similarity (with structural differences) is captured
23 by the *Least Common Ancestor* (LCA) of the “differing” concept nodes (Resnik,
24 1995). Specifically, greater the distance of the LCA from the root, more similar the
25 taxonomies are. Intuitively, a larger distance between the root and the LCA implies a
26 more localized structural change, incurring less diffusion of concept semantic relat-
27 edness. Hence, a similarity measure should monotonically decrease with decrease in
28 distance of the LCA from the root.

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

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

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

47 (a) Distance from the root to LCA of the structurally different node.
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

(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* \rightarrow *Tigers* depicts a stronger connection than *Animals* \rightarrow *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 la-

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

bels) can either be added as singletons (to model 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 α^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,

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

where $\|\cdot\|_p$ is the L_p -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).

1 KGS captures the structural similarity between the graphs, and p is set to 1 in our
 2 case. Given vectors KSV_1 and KSV_2 , computing KGS takes $O(|RV_1| + |RV_2|)$
 3 time, where RV_i is the number of reachable vertex pairs (u, v) (v is reachable from
 4 u) in graph G_i .
 5
 6

7 4.1.1 Computing the Katz Similarity Vectors

8
 9 The Katz Similarity Vector (KSV) for graph G is a vector of $|V| \times |V|$ dimen-
 10 sions encoding the Katz similarities between vertex pairs in G . However, computing
 11 similarities between all vertex pairs independently is computationally infeasible. The
 12 acyclic nature of DAGs can thus be leveraged to speedup computations by prun-
 13 ing unreachable vertex pairs, using *topological ordering*. At each traversal iteration,
 14 nodes with zero in-degree are assigned the current topological level (initialized to
 15 1). Subsequently, the graph is updated by deleting these nodes and their edges, and
 16 the current level is incremented. As there are no cycles, every vertex is assigned to a
 17 unique topological level, defining our vertex ordering in $O(|V| + |E|)$ time.
 18

19 Observe, the Katz similarity between vertex pairs with one vertex at *level* $> k$
 20 and the other at *level* k is *zero*, since there are no paths starting from a *level* $> k$
 21 node and ending at a node at *level* k . This restricts the number of vertex pairs between
 22 which the Katz similarity is computed. The following lemma states how the Katz
 23 similarity between vertices is computed based on the vertex ordering.
 24

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

$$27 \quad KS(u, v) = \alpha \times \left(\sum_{p \in \text{parents}(v)} KS(u, p) \right) + \alpha \times \mathbf{I}(u \rightarrow v) \quad (2)$$

28
 29 where indicator function $\mathbf{I}(u \rightarrow v)$ is 1 if there is an edge (u, v) , 0 otherwise.
 30
 31

32
 33 *Proof . Every path from u to p (with length l) provides a unique path from u to v*
 34 *(of length $l + 1$) by appending the edge from p to v (as p is a parent of v). Since v*
 35 *is reachable only via its parents, this provides an exhaustive enumeration of paths to*
 36 *v . Observe, there are no self edges (i.e., $KS(u, u) = 0$), and $\mathbf{I}()$ captures the case*
 37 *where u is a parent of v .*
 38

39 Let $KSV(:, p)$ denote a sub-vector encoding the Katz similarity between every vertex
 40 v and vertex p . Eq. (2) can be rewritten as,
 41

$$42 \quad KSV(:, v) = \alpha \times \left(\sum_{p \in \text{parents}(v)} KSV(:, p) \right) + \alpha \times \mathbf{I}(:, v) \quad (3)$$

43
 44 where $\mathbf{I}(:, v)$ is an indicator vector of length $|V|$, with 1 for all vertices that are parents
 45 of v , and 0 otherwise.
 46

47
 48 Lemma 1 and the topological ordering can be used to efficiently compute KSVs.
 49 The Katz similarity sub-vectors for nodes at each level of the ordering is computed
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65

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 Def. 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 (Def. 2) as a matrix³ 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,

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

³ 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 Equation (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,

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

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

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

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, 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,*

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

Proof . Consider the definitions and equations in Section 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,

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

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:

$$d_{KI} = \sum_i |KI_1(i) - KI_2(i)|$$

$$d_{GKSV} = \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_p, \text{ and}$$

$$d_{KSV} = \sum_i \|KSV_1(:, i) - KSV_2(:, i)\|_p$$

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

$$\begin{aligned} d_{KSV} &= \sum_i \|KSV_1(:, i) - KSV_2(:, i)\|_p = \sum_i \left(\sum_j |KSV_1(j, i) - KSV_2(j, i)|^p \right)^{1/p} \\ &= \sum_i \left(\sum_g \sum_{j \in g} |KSV_1(j, i) - KSV_2(j, i)|^p \right)^{1/p} \\ &\geq \sum_i \left(\sum_g \left| \sum_{j \in g} (KSV_1(j, i) - KSV_2(j, i)) \right|^p \right)^{1/p} \\ &= \sum_i \left(\sum_g \left| \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) \right|^p \right)^{1/p} = \sum_i \left(\sum_g |GKSV_1(g, i) - GKSV_2(g, i)|^p \right)^{1/p} \\ &= \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_p = d_{GKSV} \end{aligned}$$

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

$$\begin{aligned} d_{GKSV} &= \sum_i \|GKSV_1(:, i) - GKSV_2(:, i)\|_p = \sum_i \left(\sum_g \left| \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) \right|^p \right)^{1/p} \\ &\geq \sum_i \left(\left| \sum_g \sum_{j \in g} (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) \right|^p \right)^{1/p} \\ &= \sum_i \left(\left| \sum_j (KSV_1(j, i)) - \sum_{j \in g} (KSV_2(j, i)) \right|^p \right)^{1/p} \\ &= \sum_i |KI_1(i) - KI_2(i)| = d_{KI} \end{aligned}$$

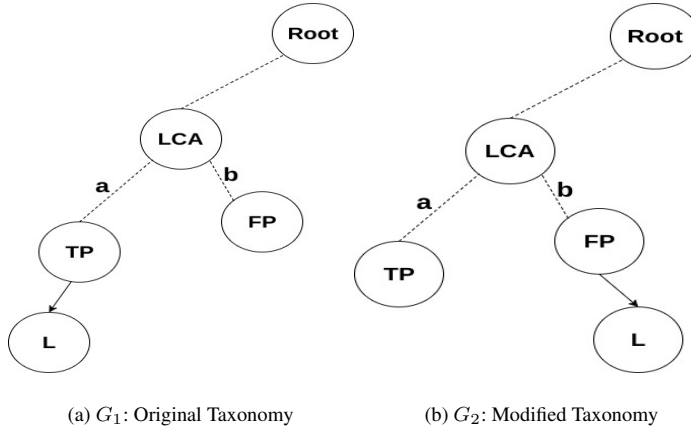


Fig. 2 Example of *atomic change* of moving (a) leaf node L in a taxonomy G_1 to (b) another parent in taxonomy G_2 .

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 $KIG(G_1, G_2) \geq GKSG(G_1, G_2) \geq KGS(G_1, G_2)$.

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 Section 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 Figure 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 Section 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,

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

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

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

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

$$\Delta = k_1 - \alpha^b k_2 \quad (10)$$

, 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,

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

where K_1 and K_2 are constants that depend on the depth of the parent TP in the first taxonomy. From Equation 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 Equation (1) is also individually monotonic with respect to both a and b , and exhibits the Proximity of LCA property.

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,

$$Imp(v) = \|KSV_1(:, v) - KSV_2(:, v)\|_p \quad (12)$$

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

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

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^l obtained from G by reversing the edges.*

$$Imp(e(u, v)) = KI(u) \times \alpha \times RevKI(v) \quad (14)$$

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.

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 ⁴. We also

⁴ 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.

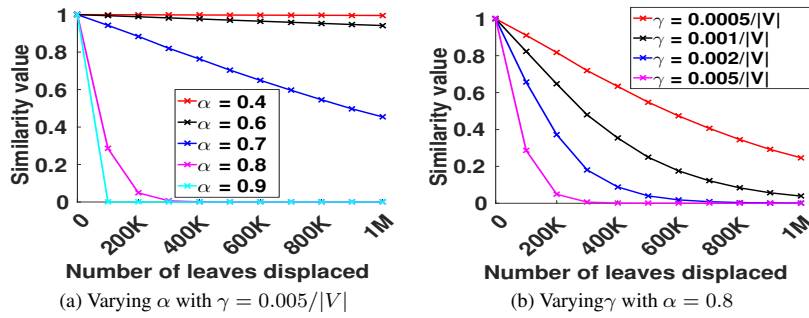


Fig. 3 Variation of Parameters for Similarity Measures.

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 α and γ 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 α controls the decay of similarity between two vertices as their path length increases, modeling importance of relationships. A higher value of α 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 3(a) shows the behaviour of different α values for increasing number of displaced leaves with ($a = 2, b = 1$) and $\gamma = 0.005/|V|$.

The parameter γ , 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 3(b) exhibits the similarity scores for different values of γ , 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 α and γ 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 $\mathcal{G} = 250$ groups. All algorithms were implemented in C and run-times reported for Intel(R) Xeon(R) E5-2470 processor with 150 GB memory.

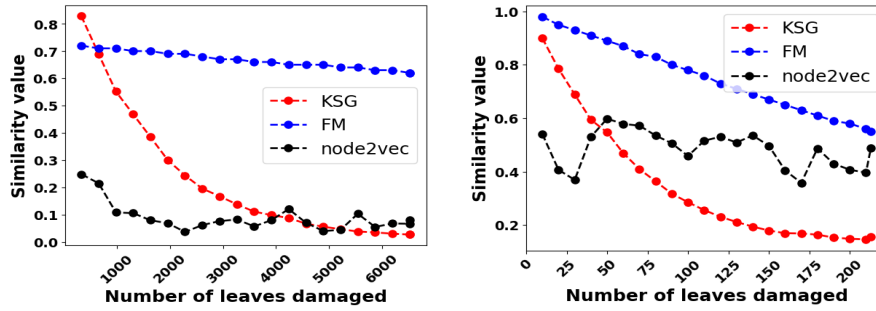


Fig. 4 Awareness of Concept Hierarchy on (a) Mobile Tech. in DBpedia & (b) Biological plant kingdom.

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 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/.

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 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 Figure 4(a), 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 Figure 4(b).

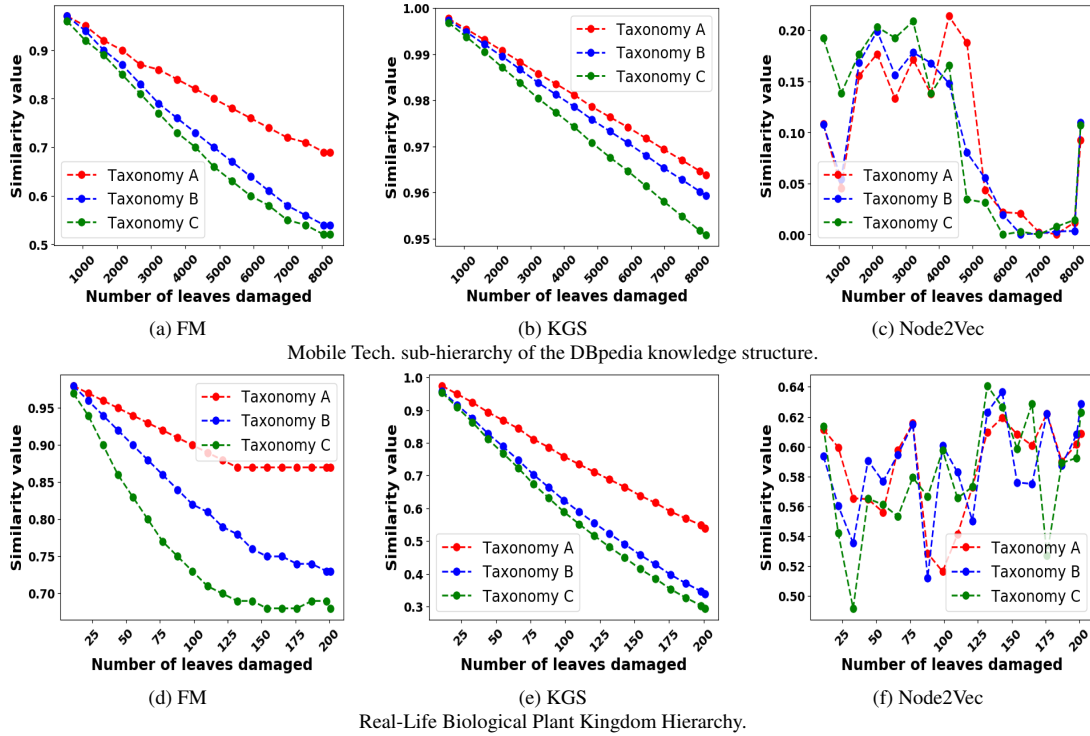


Fig. 5 Logical Subsumption property on different taxonomic perturbations for the two datasets.

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 Figure 1 in Section 3. Specifically, a particular concept vertex is dislodged from its current position and is attached to: (i) sibling of parent, (ii) sibling of grandparent, and (iii) 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 Section 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 Figure 5, we observe that the behavior for both the FM measure and our proposed Katz Graph Similarity measure is persistent with the properties of Section 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,

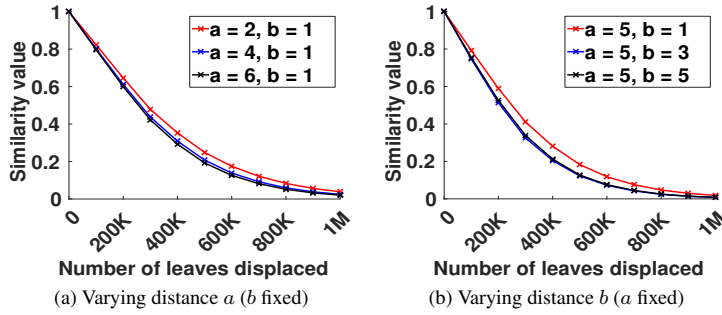


Fig. 6 Variation in Katz similarity measures with increasing distance of induced structural differences.

we again observe that the performance of the embedding technique rapidly oscillates and behaves randomly with respect to the taxonomic “damage type”.

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 Sec. 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 Figure 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) .

Figures 6(a) and 6(b) depict the similarity between the modified taxonomy and the original taxonomy with increasing number of induced structural differences. As presented in Section 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.

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.

We compute the vertex importance (as shown in Def. 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.

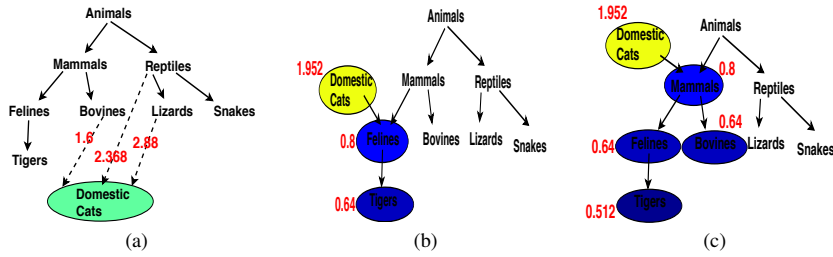


Fig. 7 Vertex Importance of concept vertices for (a) Caricature example perturbations, (b) & (c) other perturbations.

Consider the perturbations of the caricature example in Section 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 Figure 7(a).

For other diverse changes in the hierarchy structures, as shown in Figure 7(b) and (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.

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 hours to complete, while the Katz similarity measure incurs less than 4 seconds, providing an improvement of around $10000\times$. 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.

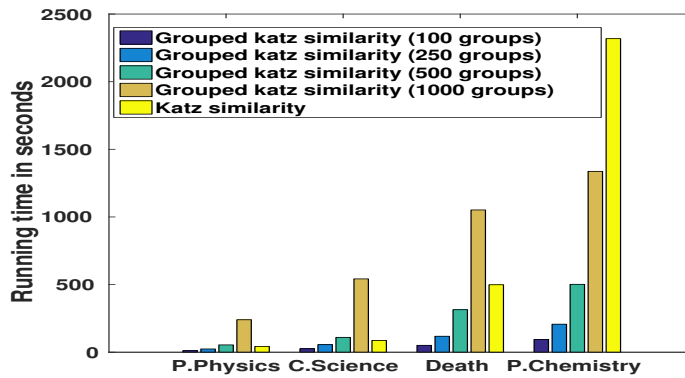
The Katz similarity measure relies on the computation of path distances between every pair of reachable vertices, and hence for huge taxonomies might be practically inefficient. Hence, the more compute efficient Grouped Katz similarity measure and

Table 1 Running times (in secs) 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	45518	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*	52900	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 ⁺	1222	171.47

*killed after 24 hrs

†out of memory

**Fig. 8** Variation of the run-time for Grouped Katz Similarity.

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 *Physical Chemistry* sub-hierarchy (with 4 million concepts), the Katz similarity and Grouped Katz similarity takes around 40 and 4 minutes respectively, while the FM measure was terminated after 24 hours. Even for the whole of DBpedia, the Grouped Katz similarity took only around 11 minutes, 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 Figure 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 500 groups was more compute intensive than the Katz similarity on *Computer Science* taxonomy.

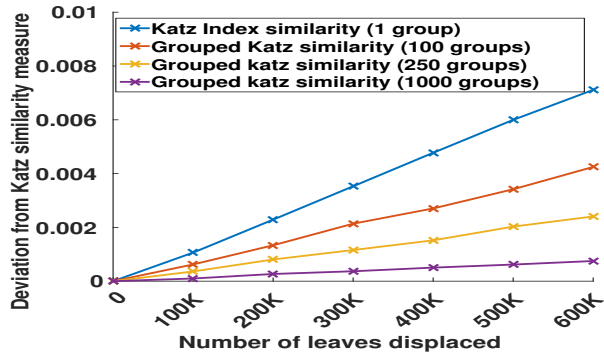


Fig. 9 Deviation in Group Katz Similarity compared to the Katz Similarity Measure with varying group sizes.

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

Loss of structural information from Katz similarity to Grouped variant. As we have observed in Figure 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 (Figure 1(a)) 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 (Figure 9). This corresponds to the bounds on the Grouped Katz similarity, as shown in Theorem 1.

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

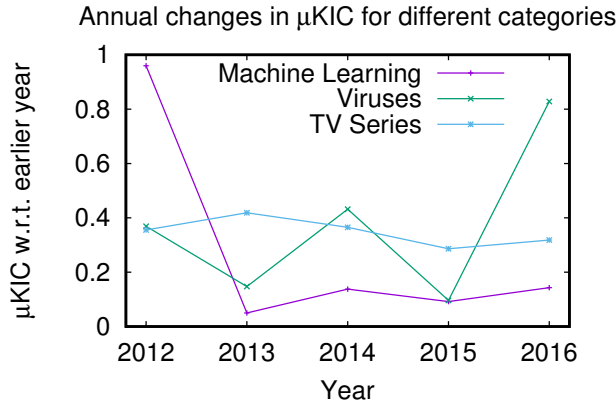


Fig. 10 μKIC for different categories.

areas in the knowledge hierarchies merely based on our graph structure comparison measure, independent of any syntactic or embedding feature. Specifically, we capture:

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

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

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

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

1 relations in these hierarchies. We also presented a linear time variant, to cater to var-
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

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× 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.

References

- E. Adams III. Consensus techniques and the comparison of taxonomic trees. *Systematic Zoology*, 21(4):390–397, 1972.
- P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1):217–239, 2005.
- K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 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, pages 1247–1250, 2008. ISBN 978-1-60558-102-6.
- G. Bordea, P. Buitelaar, S. Feralli, and R. Navigli. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *SemEval-2015*, 2015.
- G. Bordea, E. Lefever, and P. Buitelaar. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *SemEval-2016*, pages 1081–1091, 2016.
- F. Brandenburg, A. Gleiner, and A. Hofmeier. Comparing and aggregating partial orders with kendall tau distances. In *WALCOM: Algorithms and Computation*, pages 88–99, 2012.
- J. Brank, M. Grobelnik, and D. Mladenić. A survey of ontology evaluation techniques. In *SiKDD*, 2005.
- J. Brank, D. Mladenic, and M. Grobelnik. Gold standard based ontology evaluation using instance assignment. In *Workshop on Evaluation of Ontologies for the Web, EON*, 2006.
- S. Cai and K. Knight. Smatch: an evaluation metric for semantic feature structures. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 748–752. Association for Computational Linguistics, 2013.
- V. Cross, X. Yu, and X. Hu. Unifying ontological similarity measures: A theoretical and empirical investigation. *International Journal of Approximate Reasoning*, 54(7):861–875, 2013.
- M. d’Aquin. Formally measuring agreement and disagreement in ontologies. In *K-CAP*, pages 145–152, 2009.
- J. David, J. Euzenat, and O. Šváb-Zamazal. Ontology similarity in the alignment space. In *ISWC*, pages 129–144, 2010.
- J. de Knijff, K. Meijer, F. Frasincar, and F. Hogenboom. Word sense disambiguation for automatic taxonomy construction from text-based web corpora. In *WISE*, 2011.

- 1 K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation
2 of ontology learning. In *International Semantic Web Conference*, pages 228–241,
3 2006.
- 4 H. Elghawalby and E. R. Hancock. Measuring graph similarity using spectral geom-
5 etry. In *ICIAR*, pages 517–526, 2008.
- 6 D. Fensel. *Spinning the Semantic Web: Bringing the World Wide Web to its full*
7 *potential*. Mit Press, 2005.
- 8 P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern
9 recognition in the last 10 years. *Pattern Recognition and AI*, 28(01), 2014.
- 10 E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings.
11 *Journal of the American Statistical Association*, 78:553–569, 1983.
- 12 X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis*
13 *and Applications*, 13(1):113–129, 2010.
- 14 M. Geffet and I. Dagan. The distributional inclusion hypotheses and lexical entail-
15 ment. In *ACL*, pages 107–114, 2005.
- 16 P. Goyal and E. Ferrara. Graph Embedding Techniques, Applications, and Perform-
17 ance: A Survey, 2017. arXiv:1705.02801v4 [cs.SI].
- 18 P. Goyal and E. Ferrara. Graph embedding techniques, applications, and perfor-
19 mance: A survey. *Knowledge-Based Systems*, 2018.
- 20 N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. *Commu-*
21 *nications of ACM*, 45:61–65, 2002.
- 22 S. M. Harabagiu, S. J. Maiorano, and M. A. Paşca. Open-domain textual question
23 answering techniques. *Natural Language Engineering*, 9(3):231–267, 2003.
- 24 I. Hulpuş, N. Prangnawarat, and C. Hayes. Path-based semantic relatedness on linked
25 data and its use to word and entity disambiguation. In *ISWC*, pages 442–457, 2015.
- 26 D. Jurgens and M. T. Pilehvar. Semeval-2016: Semantic taxonomy enrichment. In
27 *SemEval*, pages 1092–1102, 2016.
- 28 L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18
29 (1):39–43, 1953.
- 30 D. Koutra, T.-Y. Ke, U. Kang, D. Chau, H.-K. Pao, and C. Faloutsos. Unifying Guilt-
31 by-Association Approaches: Theorems and Fast Algorithms. In *ECML-PKDD*,
32 pages 245–260, 2011.
- 33 D. Koutra, N. Shah, J. T. Vogelstein, B. Gallagher, and C. Faloutsos. Deltacon: A
34 principled massive-graph similarity function with attribution. *TKDD*, 10(3):28,
35 2016.
- 36 Z. Kozareva and E. Hovy. A semi-supervised method to learn and construct tax-
37 onomies using the web. In *EMNLP*, pages 1110–1118. Association for Computa-
38 tional Linguistics, 2010.
- 39 J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hell-
40 mann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia - A large-scale,
41 multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–
42 195, 2015.
- 43 R. Levin, H. Abassi, and U. Cohen. Guided walk: A scalable recommendation al-
44 gorithm for complex heterogeneous social networks. In *Recommender Systems*,
45 pages 293–300, 2016.
- 46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1 X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from key-
2 words. In *KDD*, pages 1433–1441, 2012.
- 3 A. Maedche and S. Staab. Measuring similarity between ontologies. In *EKAW*, pages
4 251–263, 2002.
- 5 E. Malmi, N. Tatti, and A. Gionis. Beyond rankings: comparing directed acyclic
6 graphs. *Data Mining and Knowledge Discovery*, 29(5):1233–1257, 2015.
- 7 M. McVicar, B. Sach, C. Mesnage, J. Lijffijt, E. Spyropoulou, and T. De Bie.
8 Sumoted: An intuitive edit distance between rooted unordered uniquely-labelled
9 trees. *Pattern Recognition Letters*, 79:52–59, 2016.
- 10 M. Newman. *Networks: an introduction*. Oxford university press, 2010.
- 11 M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu. Asymmetric transitivity preserving
12 graph embedding. In *KDD*, pages 1105–1114, 2016.
- 13 C. Pesquita, D. Faria, A. O. Falco, P. Lord, and F. M. Couto. Semantic similarity in
14 biomedical ontologies. *PLOS Computational Biology*, 5(7):1–12, 2009.
- 15 W. M. Rand. Objective criteria for the evaluation of clustering methods. *JASA*, 66
16 (336):846–850, 1971.
- 17 P. Resnik. Using information content to evaluate semantic similarity in taxonomy. In
18 *IJCAI*, pages 448–453, 1995.
- 19 N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient
20 graphlet kernels for large graph comparison. In *AI and Statistics*, pages 488–495,
21 2009.
- 22 F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A large ontology from
23 wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217, 2008.
- 24 J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, and S. Parthasarathy. Breaking cycles in
25 noisy hierarchies. In *WebSci*, pages 151–160, 2017.
- 26 O. Suominen and E. Hyvönen. Improving the quality of SKOS vocabularies with
27 skosify. In *EKAW*, pages 383–397, 2012.
- 28 N. Vedula, P. K. Nicholson, D. Ajwani, S. Dutta, A. Sala, and S. Parthasarathy. En-
29 riching taxonomies with functional domain knowledge. In *SIGIR*, 2018.
- 30 P. Velardi, R. Navigli, S. Faralli, and J. María Ruiz-Martínez. A new method for eval-
31 uating automatically learned terminological taxonomies. In *LREC*, pages 1498–
32 1504, 2012.
- 33 P. Velardi, S. Faralli, and R. Navigli. Ontolearn reloaded: A graph-based algorithm
34 for taxonomy induction. *Computational Linguistics*, 39(3):665–707, 2013.
- 35 J. Volker, J. Vrandečić, J. Sure, and A. Hotho. AEON – an approach to the automatic
36 evaluation of ontologies. *Journal of Applied Ontology*, 3:41–62, 2010.
- 37 S. Wagner and D. Wagner. Comparing clusterings - an overview. *Analysis*, 4769:
38 1–19, 2007.
- 39 Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, pages 133–138,
40 1994.
- 41 E. Zavitsanos, G. Paliouras, and G. A. Vouros. Gold standard evaluation of ontol-
42 ogy learning methods through ontology transformation and alignment. *TKDE*, 23:
43 1635–1648, 2011.
- 44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65