Automating the Discovery of Linking Candidates

Michael Hubert Röder

07.02.2023
Version: Publication (1.0.1)

Department of Computer Science
Data Science Group (DICE)

Doctoral Dissertation

Automating the Discovery of Linking Candidates

A dissertation presented by

Michael Hubert Röder

to the
Faculty of Computer Science, Electrical Engineering and Mathematics
of
Paderborn University
in partial fulfillment of the requirements for the degree of
Dr.rer.nat.

1. Reviewer Prof. Dr. Axel-Cyrille Ngonga Ngomo
Department of Computer Science
Paderborn University

2. Reviewer Prof. Dr. Elena Demidova
Computer Science Institute
University of Bonn

Supervisor Prof. Dr. Axel-Cyrille Ngonga Ngomo

07.02.2023

Michael Hubert Röder
Automating the Discovery of Linking Candidates
Doctoral Dissertation, 07.02.2023
Reviewers: Prof. Dr. Axel-Cyrille Ngonga Ngomo and Prof. Dr. Elena Demidova
Supervisor: Prof. Dr. Axel-Cyrille Ngonga Ngomo
The thesis was defended on 03.02.2023 in Paderborn.

Paderborn University
Data Science Group (DICE)
Department of Computer Science
Faculty of Computer Science, Electrical Engineering and Mathematics
Warburger Str. 100
33098 and Paderborn

Abstract

Like the World Wide Web, the Semantic Web has a decentralized architecture. Users and organizations can make data available and connect it to other parts of the Web. However, while the creation of datasets is well supported, the support for linking new datasets to already existing datasets is poorly supported. Our work addresses key research gaps in lifting data on the Web to structured, linked data. A dataset creator needs to be able to 1) gather datasets from the Web, 2) explore existing datasets of their area of interest, and 3) determine to which dataset they should link their dataset to. For each gap, we propose an approach and evaluate it. We propose SQUIRREL—a distributed open-source crawler for the Data Web. The crawler supports a large set of formats of structured data and is built on a modularized architecture that allows its extension for processing future formats. Our evaluation shows that SQUIRREL achieves a higher recall and is able to crawl faster than the previous state-of-the-art crawler. For the second research gap, we propose LODCAT—an approach to support the exploration of the Data Web based on human-interpretable topics. It creates a topic-based view on the datasets of the Semantic Web and therewith enables dataset creators to identify interesting datasets. With our topic evaluation framework PALMETTO, we provide measures to ensure that topic-based views can be easily understood by humans. A user study shows that human volunteers agree with the topics assigned to a set of sampled datasets. We tackle the third gap using TAPIOCA—a search engine for topically similar datasets that could be candidates for creating links. Our evaluation shows that our approach achieves a higher F1-score than several baselines and scales well. A fourth research gap arose from the evaluations of the approaches aforementioned: complex, distributed systems that process Linked Data need fair benchmarks and benchmarking platforms. Hence, we propose HOBBIT—a holistic benchmarking platform that supports the benchmarking of all steps of the Linked Data life cycle. This platform allows the benchmarking of distributed systems in a controlled environment. In addition, we propose LEMMING—an approach to generate synthetic knowledge graphs of arbitrary size that mimic real-world knowledge graphs. We further propose two new benchmarks. ORCA is a benchmark for Data Web crawlers. GLISTEN is the first benchmark for dataset interlinking recommendation systems. Both ORCA and GLISTEN are used to evaluate our previously suggested approaches for the first and the third gap, respectively.

Kurzfassung

Wie das World Wide Web hat auch das Semantic Web eine dezentrale Architektur. Personen und Unternehmen können Daten zur Verfügung stellen und sie mit anderen Teilen des Web verbinden. Während es jedoch für die Erstellung von Datensätzen bereits gute Werkzeuge gibt, wird die Verknüpfung neuer Datensätze mit bereits bestehenden Datensätzen nur unzureichend unterstützt. Unsere Arbeit befasst sich mit wichtigen Forschungslücken bei der Umwandlung von Daten im Web in strukturierte, verknüpfte Daten. Ein Erzeuger eines Datensatzes muss in der Lage sein, 1) Datensätze aus dem Web zu sammeln, 2) zu erkunden und 3) zu bestimmen, mit welchen Datensätzen er seinen Datensatz verknüpfen sollte. Für jede dieser Lücken schlagen wir einen Ansatz vor und evaluieren ihn. Wir präsentiern SQUIRREL – einen verteilten Open-Source-Crawler für das Data Web. Der Crawler unterstützt eine große Anzahl von Formaten strukturierter Daten und basiert auf einer modularisierten Architektur, die eine Erweiterung für die Verarbeitung zukünftiger Formate ermöglicht. Unsere Evaluierung zeigt, dass SQUIRREL einen höheren Recall erzielt und schneller crawlen kann als der bisherige State-of-the-Art Crawler. Für die zweite Forschungslücke schlagen wir LODCAT vor – einen Ansatz zur Unterstützung der Erkundung des Data Web auf der Grundlage von menschlich interpretierbaren Themen. Unser Ansatz erzeugt eine themenbasierte Sicht auf die Datensätze des Semantic Web und ermöglicht damit dem Ersteller eines neuen Datensatzes, für ihn interessante Datensätze zu identifizieren. Mit PALMETTO stellen wir Kohärenzmaße zur Verfügung, die sicherstellen, dass die verwendeten Themen von Menschen leicht verstanden werden können. Eine Nutzerstudie zeigt, dass Freiwillige mit der Zuordnung der Themen zu einer Reihe von Datensätzen übereinstimmen. Für die dritte Lücke präsentieren wir TAPIOCA – eine Suchmaschine für thematisch ähnliche Datensätze, die Kandidaten für die Erstellung von Verknüpfungen sein können. Unsere Evaluation zeigt, dass unser Ansatz eine höhere F1-Score als mehrere Basislösungen erreicht und gut skaliert. Eine vierte Forschungslücke ergab sich aus den Evaluierungen der oben genannten Ansätze: komplexe, verteilte Systeme, die Linked Data verarbeiten, brauchen faire Benchmarks und Benchmarking-Plattformen. Daher haben wir HOBBIT entwickelt – eine ganzheitliche Benchmarking-Plattform, die das Benchmarking aller Schritte des Lebenszyklus von Linked Data unterstützt. Diese Plattform ermöglicht das Benchmarking von verteilten Systemen in einer

kontrollierten Umgebung. Darüber hinaus präsentieren wir LEMMING – einen Ansatz zur Erzeugung synthetischer Wissensgraphen beliebiger Größe, die reale Wissensgraphen imitieren. Außerdem schlagen wir zwei neue Benchmarks vor. ORCA ist ein Benchmark für Data Web Crawler. GLISTEN ist der erste Benchmark für Systeme, die Datensätze für Verknüpfungen empfehlen. Sowohl ORCA als auch GLISTEN werden verwendet, um unsere zuvor vorgeschlagenen Ansätze für die erste bzw. die dritte Lücke zu bewerten.

Publications

In the following, we list notable mentions and publications of the author of this thesis. All publications except the two book chapters went through a peer-review. Publications on which this thesis is based on are marked with a ★ symbol.

Awards and Notable Mentions

1. Best Paper: Research Track at ISWC 2014.
   Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, **Michael Röder**, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both: *AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data*
2. Best of Workshop Award at the LD4IE workshop at ISWC 2014.
   Axel-Cyrille Ngonga Ngomo, **Michael Röder**, and Ricardo Usbeck: *Cross-Document Coreference Resolution Using Latent Features*
3. Best Demo Award at ESWC 2015.
   Ricardo Usbeck, **Michael Röder**, and Axel-Cyrille Ngomo Ngonga: *Evaluating Entity Annotators Using GERBIL*.
4. 1st prize at Task 2 of the OKE challenge at ISWC 2015
   Michael Röder, Ricardo Usbeck, René Speck, and Axel-Cyrille Ngonga Ngomo: *CETUS – A Baseline Approach to Type Extraction*.
5. Runner Up for Best Paper Award at KESW 2016.
   Ricardo Usbeck, **Michael Röder**, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo: *Requirements to Modern Semantic Search Engines*
6. Best Performing System at OKE 2017 for Tasks 1 and 2.
   René Speck and **Michael Röder**: *FOX: Ensemble Learning for Named Entity Recognition*
7. Finalist of the Rich Context Competition 2019
   Rricha Jalota, Nikit Srivastava, Daniel Vollmers, René Speck, **Michael Röder**, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo: *Finding datasets in publications: The University of Paderborn approach*.

Journal Articles

1. ★ **Michael Röder**, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo: *GERBIL – Benchmarking Named Entity Recognition and Linking Consistently*, Semantic Web, 2018. [231]
2. Ricardo Usbeck, **Michael Röder**, Michael Hoffmann, Felix Conrad, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger: *Benchmarking Question Answering Systems*, Semantic Web, 2019. [292]
3. ★ **Michael Röder**, Denis Kuchelev, and Axel-Cyrille Ngonga Ngomo: *HOBBIT: A platform for benchmarking Big Linked Data*, Data Science, 2020. [237]

Conference Articles

1. **Michael Röder**, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both: *N^3 – A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format*, LREC 2014. [226]
2. Axel-Cyrille Ngonga Ngomo, **Michael Röder**, and Ricardo Usbeck: *Cross-Document Coreference Resolution Using Latent Features*, LD4IE 2014. [202]
3. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, **Michael Röder**, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both: *AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data*, ISWC 2014. [285]
4. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, **Michael Röder**, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both: *AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data*, ECAI 2014. [284]
5. Ricardo Usbeck, **Michael Röder**, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann: *GERBIL – General Entity Annotation Benchmark Framework*, WWW 2015. [287]
6. ★ **Michael Röder**, Andreas Both, and Alexander Hinneburg: *Exploring the Space of Topic Coherence Measures*, WSDM 2015. [227]
7. Ricardo Usbeck, **Michael Röder**, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo: *Requirements to Modern Semantic Search Engine*, KESW 2016. [288]
8. ★ **Michael Röder**, Axel-Cyrille Ngonga Ngomo, Ivan Ermilov, and Andreas Both: *Detecting Similar Linked Datasets Using Topic Modelling*, ESWC 2016. [230]

9. Jean Carlos Oliveira de Abreu, Renato Fileto, Axel-Cyrille Ngonga Ngomo, **Michael Röder**, Matthias Wittwer, and Horacio Saggion: *Characterizing Mention Mismatching Problems for Improving Recognition Results*, iiWAS 2017. [77]
10. Kunal Jha, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *All that glitters is not gold – rule-based curation of reference datasets for named entity recognition and entity linking*, ESWC 2017. [138]
11. Diego Moussallem, Ricardo Usbeck, Michael Röeder, and Axel-Cyrille Ngonga Ngomo: *MAG: A Multilingual, Knowledge-Base Agnostic and Deterministic Entity Linking Approach*, K-CAP 2017. [191]
12. Axel-Cyrille Ngonga Ngomo, **Michael Röder**, Diego Moussallem, Ricardo Usbeck, and René Speck: *BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking*, INLG 2018. [203]
13. Zafar Habeeb Syed, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *FactCheck: Validating RDF Triples Using Textual Evidence*, CIKM 2018. [269]
14. Zafar Habeeb Syed, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *Unsupervised discovery of corroborative paths for fact validation*, ISWC 2019. [270]
15. Abdelmoneim Amer Desouki, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *Ranking on Very Large Knowledge Graphs*, HT 2019. [79]
16. ★ **Michael Röder**, Geraldo de Souza Jr, and Axel-Cyrille Ngonga Ngomo: *Squirrel – Crawling RDF Knowledge Graphs on the Web*, ISWC 2020. [236]
17. ★ **Michael Röder**, Pham Thuy Sy Nguyen, Felix Conrads, Ana Alexandra Morim da Silva, and Axel-Cyrille Ngonga Ngomo: *LEMMING – Example-based Mimicking of Knowledge Graphs*, ICSC 2021. [240]
18. ★ **Michael Röder**, Geraldo de Souza Jr., Denis Kuchelev, Abdelmoneim Amer Desouki, and Axel-Cyrille Ngonga Ngomo: *ORCA – a Benchmark for Data Web Crawlers*, ICSC 2021. [239]
19. Abdelmoneim Amer Desouki, Felix Conrads, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *SynthG: Mimicking RDF Graphs using Tensor Factorization*, ICSC 2021. [80]
20. **Michael Röder**, Philip Frerk, Felix Conrads, and Axel-Cyrille Ngonga Ngomo: *Applying Grammar-Based Compression to RDF*, ESWC 2021. [232]
21. Ana Alexandra Morim da Silva, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *Using Compositional Embeddings for Fact Checking*, ISWC 2021. [75]

Book Chapters

1. **Michael Röder**, Mohamed Ahmed Sherif, Muhammad Saleem, Felix Conrads, and Axel-Cyrille Ngonga Ngomo: *Benchmarking the Lifecycle of Knowledge Graphs*. [238]
2. Rricha Jalota, Nikit Srivastava, Daniel Vollmers, René Speck, **Michael Röder**, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo: *Finding datasets in publications: The University of Paderborn approach*. [135]

Workshop Articles, Posters & Others

1. **Michael Röder**, Maximilian Speicher, and Ricardo Usbeck: *Investigating Quality Raters' Performance Using Interface Evaluation Methods*, Poster at Informatik 2013. [224]
2. **Michael Röder**, Andreas Both, and Alexander Hinneburg: *Evaluation des Konfigurationsraumes von Kohärenzmaßen fur Themenmodelle*, KDML workshop at LWA 2014. [225]
3. Frank Rosner, Alexander Hinneburg, **Michael Röder**, Martin Nettling, and Andreas Both. *Evaluating topic coherence measures*.[1] [233]
4. Ricardo Usbeck, **Michael Röder**, and Axel-Cyrille Ngomo Ngonga: *Evaluating Entity Annotators Using GERBIL*, Demo paper at ESWC 2015. [286]
5. **Michael Röder**, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo: *Developing a Sustainable Platform for Entity Annotation Benchmarks*, Developers workshop at ESWC 2015. [228]
6. **Michael Röder**, Ricardo Usbeck, René Speck, and Axel-Cyrille Ngonga Ngomo: *CETUS – A Baseline Approach to Type Extraction*, Semantic Web Evaluation Challenges 2015. [229]
7. Axel-Cyrille Ngonga Ngomo and **Michael Röder**, *HOBBIT: Holistic Benchmarking for Big Linked Data*, EU networking session at ESWC 2016. [201]
8. Ricardo Usbeck, **Michael Röder**, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo: *DIESEL – Distributed Search over Large Enterprise Data*, EU networking session at ESWC 2016. [289]
9. Muhammad Saleem, Ricardo Usbeck, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo. *SPARQL Querying Benchmarks*, Tutorial at ISWC 2016. [244]

[1] This publication has been peer-reviewed and accepted at the Topic Models: Computation, Application and Evaluation workshop at the Neural Information Processing Systems conference 2013 (see `https://sites.google.com/site/nips2013topicmodels/papers`, retrieved on 31.07.2022). The workshop had no proceedings and, hence, the paper has been published on arxiv.org.

10. René Speck, **Michael Röder**, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo: *Open Knowledge Extraction Challenge 2017*, Semantic Web Challenges at ESWC 2017. [258]
11. Kleanthi Georgala, Mirko Spasić, Milos Jovanovik, Henning Petzka, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *MOCHA2017: The Mighty Storage Challenge at ESWC 2017*, Semantic Web Challenges at ESWC 2017. [103]
12. **Michael Röder**, Tzanina Saveta, Irini Fundulaki, and Axel-Cyrille Ngonga Ngomo: *HOBBIT Link Discovery Benchmarks at Ontology Matching 2017*, OM Workshop at ISWC 2017. [235]
13. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, **Michael Röder**, and Giulio Napolitano: *7th open challenge on question answering over linked data (QALD-7)*, Semantic Web Challenges at ESWC 2017. [290]
14. Kunal Jha, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo. *Eaglet – a Named Entity Recognition and Entity Linking Gold Standard Checking Tool*, Demo paper at ESWC 2017. [137]
15. René Speck, **Michael Röder**, Felix Conrads, Hyndavi Rebba, Catherine Camilla Romiyo, Gurudevi Salakki, Rutuja Suryawanshi, Danish Ahmed, Nikit Srivastava, Mohit Mahajan, and Axel-Cyrille Ngonga Ngomo: *Open Knowledge Extraction Challenge 2018*, Semantic Web Challenges at ESWC 2018. [259]
16. Kleanthi Georgala, Mirko Spasić, Milos Jovanovik, Vassilis Papakonstantinou, Claus Stadler, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *MOCHA2018: The Mighty Storage Challenge at ESWC 2018*, Semantic Web Challenges at ESWC 2018. [104]
17. Diego Moussallem, Ricardo Usbeck, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *Entity Linking in 40 Languages Using MAG*, Demo paper at ESWC 2018. [192]
18. Ernesto Jiménez-Ruiz, Tzanina Saveta, Ondrej Zamazal, Sven Hertling, **Michael Röder**, Irini Fundulaki, Axel Ngonga Ngomo, Mohamed Sherif, Amina Annane, Zohra Bellahsene, Sadok Ben Yahia, Gayo Diallo, Daniel Faria, Marouen Kachroudi, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Maximilian Mackeprang, Majid Mohammadi, Maciej Rybinski, Booma Sowkarthiga Balasubramani, and Cássia Trojahn: *Introducing the HOBBIT platform into the ontology alignment evaluation campaign*, OM workshop at ISWC 2018. [139]
19. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Felix Conrads, **Michael Röder**, and Giulio Napolitano: *8th challenge on question answering over linked data (QALD-8)*, QALD-8 challenge at ISWC 2018. [291]

20. Zafar Habeeb Syed, **Michael Röder**, and Axel-Cyrille Ngonga Ngomo: *COPAAL – An Interface for Explaining Facts using Corroborative Paths*, Demo paper at ISWC 2019. [271]
21. Diego Moussallem, Paramjot Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, **Michael Röder**, René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo: *A General Benchmarking Framework for Text Generation*, WebNLG+ workshop at INLG 2020. [194]
22. Kleanthi Georgala, **Michael Röder**, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo: *Applying edge-counting semantic similarities to Link Discovery: Scalability and Accuracy*, OM workshop at ISWC 2020. [105]

Acknowledgments

I would like to express my deepest gratitude to Prof. Dr. Axel-Cyrille Ngonga Ngomo for his invaluable advice, continuous support, and patience during my Ph.D study.

I would also like to thank my second reviewer Prof. Dr. Elena Demidova as well as the rest of my thesis committee—Prof. Dr. Friedhelm Meyer auf der Heide, Prof. Dr. Gregor Engels, and Jun.-Prof. Dr. Sebastian Peitz—for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere thanks also goes to Prof. Dr. Andreas Both, who provided me the opportunity to start my research within his research team and his continuous support even after this time.

I am extremely grateful to all my great colleagues and coauthors with which I was allowed to work together. I would like to list all of them here, but that would go far beyond my page limit.

This endeavor would not have been possible without the support of my family and friends—especially the support of my parents and siblings.

Parts of this work were supported by 1) the European Social Fund and the Free State of Saxony (Application no. 100126307), 2) the German Federal Ministry of Transport and Digital Infrastructure (BMVI) projects LIMBO (GA no. 19F2029C) and OPAL (GA no. 19F2028A), 3) the German Federal Ministry of Education and Research (BMBF) within the EuroStars project DIESEL (Project no. E!9367, 01QE1512C) and the project DAIKIRI (GA no. 01IS19085B), 4) the German Federal Ministry for Economic Affairs and Energy (BMWi) within the project RAKI (GA no. 01MD19012D), and 5) by the European Union's H2020 research and innovation action HOBBIT (GA no. 688227) as well as the H2020 Marie Skłodowska-Curie project KnowGraphs (GA no. 860801).

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

1 Introduction

In April 2020, the Word Wide Web had 4.6 billion users [4]. This number further grew to 4.96 billion users in January 2022 [145]. The average time spent by users online has also increased [31, 145]. For example, a representative study from Germany shows that the average time a single person spent online during its free time grew from 204 minutes per day in 2020 to 227 minutes in 2021 [31].[1] In the same study, 87% of the users reported to use a Web search engine at least once per week [31]. In another study, "finding information" was the most often selected reason why people use the internet [145]. Although the majority of search queries that users type into the Google search engine comprise only 3 words on average [98], it is known that the information need of many users is much more complex [33, 210].[2] The discrepancy between users having complex information needs and the average length of queries is known to be due to multiple reasons: 1) the processing of complex natural language queries is challenging [26, 33, 210] and 2) complex information needs are more unlikely to be answered with the information from a single source [64, 148, 234]. The first reason, i.e., the challenges in processing long natural language queries, can lead to the issue that a Web search engine is not able to identify the main concept a user is looking for [26, 33, 210]. This often leads to an adaptation of user behavior, i.e., instead of writing long queries with all details of their information need, more experienced users use short keyword queries [38]. However, these users withhold detailed information that could be helpful for the search process and this behavior adds to the cognitive burden of these users [210]. The second reason, i.e., the necessity to combine information from multiple sources, is particularly relevant for the work presented herein. Hence, consider the following example of a user searching for a product online. This example fits to the aforementioned situation since the user has a complex information need [211] and 46% of all product searches on the Web begin on Google [186]. To answer the product search query of the user, a Web

[1] The time does not only cover the Word Wide Web in its form of Web pages but also services that are offered, e.g., messengers or streaming services. Kemp [145] has a global view on the subject matter and reports a higher number of daily internet usage of 6 hours and 58 minutes. It should be noted that the two studies include different activities and that their numbers are not comparable. However, both studies report an increase in the time spent online.

[2] We use the Google search engine as example since it had a market share of more than 90% in January 2022 [145].

search engine has to collect and integrate product data. Both—the collection and integration—are challenging if they rely solely on the Document Web. In particular, collecting data means having to crawl the Web pages of online shops as well as local stores. All these pages offer details about the product. In the Document Web however, the pages offer this information only as human-readable text. While a program that extracts the data from these pages can be written [236] it would have to be adapted to each of these pages since it is very likely that said pages have different Hyper Text Markup Language (HTML) trees. In addition, the maintenance of such a program would include the adaptation to layout changes of the Web pages. Similar challenges hold for the integration of the data. Different stores use different names for the same products, e.g., some have a plain product name while others include product details. This creates the need for another program for integrating the data pertaining to any given product across Web sources into a single representation.

Within the Semantic Web, all these functionalities can be implemented easier, if the shops offer the data in a machine-readable form [148]. This form of data comes with explicit semantics [35], i.e., the data does not only consist of values but also of identifiers that express the meaning of the different values. Gathering the data would hence be reduced to parsing standardized languages, e.g., a serialization of the Resource Description Framework (RDF). The integration of the data is eased as well. Web shops increasingly use a common ontology to describe products. This ontology, called `schema.org`, is the result of a project started in 2011 by Google, Microsoft, Yahoo, and Yandex. It offers a set of class hierarchies for entities from several domains and properties for the relations that these entities can have. This common ontology eases the work of application and Web developers to share their data [111]. Similarly, a system of common identifiers for products could be established. The Digital Object Identifiers (DOIs) are an example for common identifiers that are used to identify publications of several publishers within a common scheme [134]. A similar example are Global Trade Item Numbers (GTINs) [6], which most people may know in form of product bar codes that are scanned in supermarkets. However, even if shops use different ontologies and identifiers, there are several approaches for the (semi-)automatic integration of their data. For example, similar ontologies can be aligned to each other [91] and identifiers that refer to the same real-world product can be connected with links [205].[3] Based on either common ontologies and identifiers or the aforementioned approaches, the data gathered from several

[3]The creation of an alignment between two ontologies is called “Ontology Matching” [91]. The identification of entities within different datasets that represent the same real-world entity is known as “Link Discovery” (or short “Linking”) [205]. Both areas are own active fields of research that are not discussed in detail within this thesis.

shops can be fused to create a single dataset [311] that serves as the basis for the product search engine.

The aforementioned product search is only one example of many different applications which benefit from the usage of Semantic Web technologies. However, 20 years have passed since Berners-Lee et al. [35] described the idea of a Semantic Web and while there was a considerable amount of research results published in this area, it is arguable whether there are as many Semantic Web-based applications as some visionaries have expected [120, 156]. In 2006, Shadbolt et al. [251] reviewed the state of the art of the Semantic Web. They concluded that the Semantic Web had not fulfilled its vision in the previous years. This time had been used for defining languages and standards but the authors argue that "languages and standards are of no consequence without uptake, and uptake requires increasing the amount of data exposed in RDF" [251]. Hence, the success of the Semantic Web is bound to

1. The amount of data that is published and
2. The quality of the published data.

Still, The Semantic Web already has grown during recent years. The amount of data published as knowledge graphs has clearly increased significantly since the remarks of Shadbolt et al. [251]. The knowledge graphs available on the Web have been growing both in number and size [22, 90, 95]. This development has been accelerated by several governments publishing public sector data on the Web.[4] At the same time, the number of Web pages with embedded Semantic Web data grew as well [220]. However, while the Semantic Web can easily grow by simply adding more data, achieving the second requirement to gain uptake is not as easy. Berners-Lee [34] points out the importance of having links between the data. A main value of the Semantic Web is the creation of links that connect data points in a semantically meaningful way. These links can be used by humans or machines to gather more information. For example, a user of the aforementioned product search might be looking for a specific camera. Imagine the search engine retrieved the information that although the camera is sold in several Web shops, it is not available in any shop at the moment. The user may want to continue his search for an alternative camera by asking the search engine for similar products. In this situation, our product search needs to be able to connect the camera of one manufacturer with other products of other manufacturers to be able to compare them. Searching for these connections in the moment in which the user asks for them is time-consuming [205]. The links have to be created beforehand. This could be done by the consumer of the data (e.g.,

[4] Examples include the European Union (`https://ec.europa.eu/digital-single-market/en/open-data`; last accessed on 31.07.2022) and the German Federal Ministry of Transport and Digital Infrastructure (`https://mobilithek.info/`; last accessed on 31.07.2022)

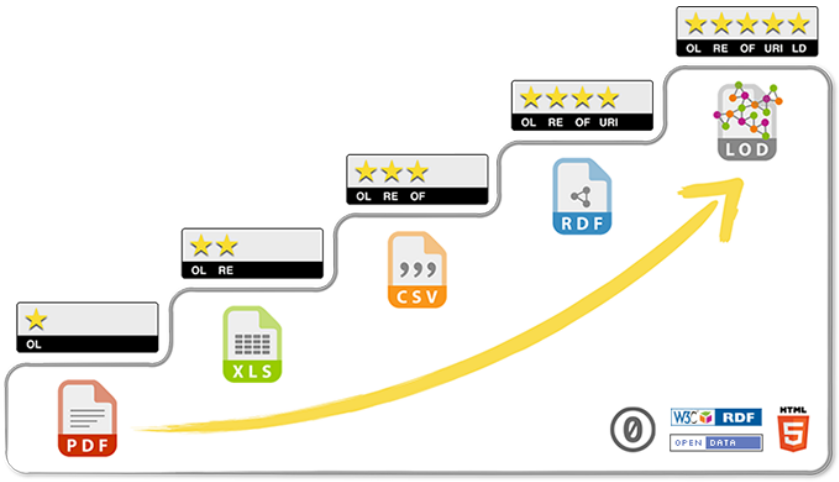


Figure 1.1.: The five star deployment scheme for Linked Open Data [147].

Table 1.1.: Single levels of the five star deployment scheme for Linked Open Data [132].

Stars	Description
⋆	Data published on the Web in any format (e.g., PDF, JPEG) accompanied by an explicit Open License (expression of rights).
⋆ ⋆	The published data is structured data in a machine-readable format (e.g., Extensible Markup Language (XML)).
⋆ ⋆ ⋆	The published data has a documented, non-proprietary data format (e.g., Comma Separated Value (CSV), Keyhole Markup Language (KML)).
⋆ ⋆ ⋆ ⋆	The data is published as RDF (e.g., Turtle, Resource Description Framework in Attributes (RDFa), JavaScript Object Notation for Linked Data (JSON-LD), SPARQL Protocol And RDF Query Language (SPARQL)).
⋆ ⋆ ⋆ ⋆ ⋆	Identifiers in the published RDF data are links (Uniform Resource Locators (URLs)) to useful data sources.

our product search) or by the data provider. The latter solution has the advantage that the provided data already comes with existing links and, hence, has a higher quality. Berners-Lee defines two terms to describe this lifted type of data on the Semantic Web. *Linked Data* represents data that is linked to other data. *Linked Open Data (LOD)* is an extension of this term that additionally includes an open license. He proposes to measure the quality of published data on the scale of the incremental five star deployment scheme for Linked Open Data [34] shown in Figure 1.1. The definitions of the single levels are listed in Table 1.1.

The categories show a clear separation between three-, four- and five-star data. The three-star data is hard to use for autonomous machines since each CSV file may have different columns with different meanings. Even if two columns have the same

meaning, they may come with different column headings. Imagine the different ways the owner of a Web page could name a column that contains the product names. This could range from generic terms like “product name” or “label” to more specific terms like “book title”. In contrast, a four-star RDF dataset comes with explicit identifiers for properties [250]. This enables machines to derive connections between them, e.g., that two properties are equivalent.

Although RDF datasets are valuable, datasets that achieve all five stars are most valuable since they are connected to other datasets. However, not all four-star datasets are lifted to five-star quality. The LODStats project indexed 9960 RDF datasets from the Web in 2016 [90].[5] The authors found that although the number of links between datasets increased between 2012 and 2016, only 10% of the entities in datasets were linked to entities from other datasets. Schmachtenberg et al. [246] analyze 1014 datasets and find that most datasets are only sparsely linked. 44% of the analyzed datasets have no outgoing links to other datasets and thus have only incoming links, or are completely isolated. A similar observation can be made when comparing the number of RDF datasets available on the web. Fernández et al. [95] gathered more than 650 thousand RDF datasets from data portals. However, the manually curated Linked Open Data cloud project lists only 1255 datasets [177].[6] One of the requirements the latter project raises for listing a dataset is that it has to be linked to one of the already listed datasets. These examples of the large difference between the number of four- and five-star datasets evince an important research gap. While the Semantic Web community defined the target that a person should reach for publishing a good dataset, it is not further clarified how a data publisher can easily lift a dataset to five-star quality. The tasks necessary to lift a dataset from three to five stars cover two main steps: the exploration of existing datasets and the identification of datasets that can be used for creating links. At the moment, both steps are mainly covered by manual tasks and not well supported by tools. Hence, these steps are very costly and even unsolvable for non-experts.

The exploration of existing datasets enables a data publisher to be aware of datasets and ontologies that are already available. This can give several benefits to the data publisher. First, they can identify existing ontologies and how they are used in practice. Second, a data provider can identify existing RDF datasets that they can reuse. Imagine the owner of a small online shop that would like to increase the visibility of his shop by making his offers available to the product search described above. The main entities he would like to provide to the search engine are the products he sells in the shop. It would remove a lot of effort if he could find an

[5] `http://stats.lod2.eu/`; last accessed on 27.07.2022.
[6] Status from May 2020 [177], `https://www.lod-cloud.net/`; last accessed on 24.08.2021.

open dataset that describes some of the products in his shop and that he can reuse. This can free a data provider from defining entities, classes, and properties on their own and, hence, save costs. Bontas et al. [48] describe two case studies from 2005 and report that for creating an ontology to publish data in the domain of e-recruitment, 15% of the time was needed to identify related ontologies. Additional 40% of the time was needed to transform the identified ontologies into the Web Ontology Language (OWL) representation that is used in the Semantic Web. The conclusion of the authors is that the various steps that are necessary to create an ontology are not supported by tools. Shadbolt et al. [251] state that reusing existing datasets and ontologies is one of the major uptakes the Semantic Web offers. As an additional benefit, reusing entities from an already existing dataset within a new dataset creates a link between these two datasets and can already lift the new dataset to the targeted five star quality. However, identifying related datasets and ontologies is an important preliminary step for reusing them. An online search that supports the identification of ontologies and datasets that already exist in the Semantic Web is a necessary tool that can support this. Vandenbussche et al. [295] propose a Web search for RDF vocabularies, which covers a part of the aforementioned need.[7] However, it is designed as a classic keyword-based search on the metadata and single elements of the vocabulary combined with manually curated tags. Hence, the user has to match exactly the right term to be able to identify a potentially interesting vocabulary. As Chapman et al. [65] point out, such a search is not optimal to find interesting datasets.

A similar problem arises for the second step, i.e., the identification of datasets that can be used for creating links. When a newly created RDF dataset should be linked to existing RDF datasets using (semi-)automatic link discovery algorithms, the dataset provider has to know to which dataset the new dataset can be linked to. With the growth of the number of datasets available as well as the growth of their size comes the problem of effectively detecting not only the links between the datasets (as studied in previous works [200]) but also of determining the datasets with which a novel dataset should be linked. A näive approach to link these datasets would choose two datasets and check whether they can be linked with each other. Such an approach would need a quadratic-growing number of pairwise comparisons of datasets to find possible candidates for linking, which is clearly impracticable. This is hardened by the fact that the Semantic Web (as the Web itself) is decentralized and datasets are distributed [35]. Addressing the problem of finding relevant datasets for linking is however of crucial importance to facilitate the integration of novel

[7] `https://lov.linkeddata.es/dataset/lov`, last accessed on 31.07.2022.

datasets into the Linked Data Cloud as well as the discovery of relevant data sources in enterprise Linked Data [200].

We can summarize that the Semantic Web does not only need to grow but that it also needs high-quality data to have an impact. However, the growth of the Web increases the complexity of publishing more high-quality data. *Our work addresses four key research gaps in lifting data from three to five stars.* These research gaps are as follows:

RG1 Datasets are distributed across the Web. Existing Data Web crawlers are limited with respect to the data formats they support and their scalability. Hence, we need a way to gather datasets from the Web in an effective and efficient way. We propose SQUIRREL—a distributed open-source crawler for the Data Web that supports a large set of formats of structured data and is built on a modularized architecture that allows the extension for future formats. Our evaluation shows that SQUIRREL achieves a higher recall and is able to crawl faster than the previous state-of-the-art crawler. SQUIRREL is used to gather datasets from the Semantic Web.

RG2 Dataset creators need to be aware of already existing datasets. Hence, we need to create a service that allows dataset creators to explore existing datasets of their area of interest. Searching for datasets on the Web is different to a classic Web search since keyword-based search engines only cover a single view on datasets [65] and is in many cases limited to the dataset's meta data [53]. Other approaches rely on user-defined tags to offer a faceted search [217]. However, this does not only involve manual work but may also be bound to the subjective tags a data provider defines in the meta data of their datasets. Within this thesis, we propose LODCAT—an approach to support the exploration of the Semantic Web based on human-interpretable topics. It creates a topic-based view on the datasets of the Semantic Web to enable dataset creators to identify interesting datasets . With our topic evaluation framework PALMETTO, we provide measures to ensure that this view can be easily understood by humans so that it supports the identification of datasets that are connected to a certain topic a user is interested in. A user study shows that human volunteers agree with the topics assigned to a set of sampled datasets.

RG3 Dataset publishers need to know to which dataset they could link their dataset to. We tackle this research gap using TAPIOCA—a search engine for topically similar datasets that could be candidates for link discovery. With this approach, we index existing datasets based on extracted metadata. Given a newly created dataset, TAPIOCA identifies candidate datasets that can be used to create links

between both datasets using existing linking approaches. Our evaluation shows that our approach is better than several baselines and scales well on a large number of datasets.

RG4 Complex, distributed systems that process Linked Data like the approaches proposed in this thesis need fair benchmarks and benchmarking platforms. Hence, we propose HOBBIT—a holistic benchmarking platform that supports the benchmarking of all steps of the Linked Data life cycle [23]. This platform allows the benchmarking of distributed systems in a controlled environment. In addition, we propose LEMMING—an approach to generate synthetic knowledge graphs of arbitrary size that mimic real-world knowledge graphs. We use the generated datasets to evaluate the scalability of Linked-Data-based systems. We further propose the two benchmarks ORCA and GLISTEN. ORCA is a benchmark for Data Web crawlers. GLISTEN is the first benchmark for dataset interlinking recommendation systems. Both are used to evaluate our previously suggested approaches for the first and the third gap, respectively.

The remaining of this thesis is structured as follows. First, preliminaries are defined in Section 2. After that, we present the HOBBIT benchmarking platform and the LEMMING algorithm to generate synthetic graphs in Chapter 3. Chapter 4 presents our distributed open source crawler for the Data Web named SQUIRREL, and the ORCA benchmark that is used to evaluate it. In Chapter 5, we present our topic-based RDF dataset search LODCAT and our coherence-based topic evaluation tool PALMETTO. Chapter 6 presents our linking candidate recommendation approach TAPIOCA and the fact-checking-based benchmark GLISTEN that is used for the evaluation. We conclude the thesis in Chapter 7.

Note that parts of this thesis have been published as peer-reviewed articles at research conferences and in scientific journals. Hence, at the beginning of the chapters 3–6, a footnote marked with the ¶ symbol lists published articles that overlap with the chapter's content and the role of the thesis' author within the creation of these articles.

2 Preliminaries

This chapter addresses preliminaries that are necessary for the other chapters of this thesis. It comprises three parts—1) Semantic Web, 2) Latent Dirichlet Allocation, and 3) statistical measures.

2.1 Semantic Web

The Semantic Web has been proposed by Berners-Lee et al. [35] as an extension of the World Wide Web. It offers information in a structured way so that it can be processed by machines. Its name refers to its goal that "information is given well-defined meaning" [35]. This section introduces basic concepts of the Semantic Web—Linked Data and the Resource Description Framework. The interested reader is referred to Ngonga Ngomo et al. [200] for an extended introduction and to Hogan et al. [131] for a recent survey.

2.1.1 Linked Data

The Semantic Web is built on the concept of representing content in a machine-readable format to give programs access to it [35]. However, Berners-Lee [34] points out that publishing large amounts of data does not create a Web. The pieces of data that are available should be linked to each other. Hence, he suggests four Linked Data principles [34]:

1. Internationalized Resource Identifiers (IRIs) [86] should be used as identifiers for things.[1]
2. The IRIs should use the Hypertext Transfer Protocol (HTTP) to enable users and machines to look up these identifiers.
3. When such an IRI is looked up, useful information should be provided.[2]

[1] In the original article, Berners-Lee suggests the usage of Uniform Resource Identifiers (URIs). Nowadays the usage of their extension—IRIs [86]—is preferred [200].

[2] Berners-Lee suggests the usage of standards like RDF-star [18] or SPARQL [34].

4. The data should include links to other IRIs so that users can discover more things.

As in the World Wide Web, where documents are linked with each other via hyperlinks, the single pieces of data within the Semantic Web should be linked to each other as well. To this end, the Semantic Web relies on established standards like IRIs. They are used to identify items in the domain of interest [200]. These items are also called entities or resources and are the things that are further described in the data [200].

As stated in the Linked Data principles, HTTP IRIs are preferred in comparison to identifier schemes like Uniform Resource Name (URN) [241] or Digital Object Identifier (DOI) [134, 200]. Following RFC 3987 [86], the structure of an IRI is as follows:

```
scheme ":" hier-part [ "?" query ] [ "#" fragment ]
```

The `scheme` is the identifier of the IRI scheme. The hierarchical part (`hier-part`) of an IRI typically comprises an authority and an optional path.[3] It is followed by the optional query and fragment parts. An IRI can also be thought of as a URI that is not restricted to the usage of ASCII symbols.

An HTTP IRI is an IRI that uses the HTTP scheme. An example for such an identifier that could be used for Paderborn is

```
http://en.wikipedia.org/wiki/Paderborn .
```

These identifiers have two major advantages [200]. First, they allow an easy sheme to use global identifiers for resources that has already shown its worth on the Web. Second, they can be used to get more information about the entity, i.e., in this case about the city of Paderborn, by using HTTP requests. The latter feature is called *dereferencability*.

Definition 2.1 (Dereferencing)**.** *The process of using an identifier of a resource to request more information about it is called dereferencing.*

HTTP IRIs tend to be lengthy. Hence, we will use a prefixed writing for IRIs within this thesis. Similar to XML namespaces, we shorten IRIs by defining a prefix which is a local replacement for the namespace IRI [74]. The example IRI from above could

[3] For simplicity, we skip the usage of relative paths and similar features of IRIs.

be shortened to `wiki:Paderborn_University` where `wiki` is a local prefix that would be replaced with `http://en.wikipedia.org/wiki/` to retrieve the original IRI. A list of prefixes used throughout this thesis can be found on page 262.

Bernsers-Lee [34] defines a subset of Linked Data named Linked Open Data. This is Linked Data that is published on the Web and has an open license, e.g., a Creative Commons license.[4] Within this thesis, we define the terms Web of Data [57, 296] and Data Web [106] as synonyms for Linked Data.

2.1.2 Resource Description Framework

Means for expressing knowledge in a manner which abides by the Linked Data principles are offered by the Resource Description Framework (RDF). RDF is a World Wide Web Consortium (W3C) recommendation to express information about resources. "Resources can be anything, including documents, people, physical objects, and abstract concepts" [250]. The main building block of RDF are triples that express facts about resources [250]. Triples can be thought of like simple statements that comprise three parts—a subject, a predicate, and an object. The subject is the resource for which the fact holds. The predicate expresses the property that is further defined for this resource. The object represents the value of the property.

In RDF, there are three basic sets that can be used for a triple—IRI resources, literals, and blank nodes [74].

Definition 2.2 (IRI resource)**.** *An IRI resource is a resource which is identified by at least one global identifier, i.e., an IRI [74].*

Resources that are related to each other can be organized by using the same XML namespace for their IRIs. These sets of resources are named RDF vocabulary. The XML namespace is used as IRI of the RDF vocabulary [74].

Definition 2.3 (Literal)**.** *A literal is a basic value that is not represented as IRI [74].*

A literal has a datatype [74]. The datatype defines the structure of the literal and enables the parsing of the literal. String literals may have an additional language tag [74].

[4] `https://creativecommons.org/`

Definition 2.4 (Blank node). *A blank node is a resource that is not further defined [74]. In contrast to an IRI resource, it does not have a global identifier. In contrast to literals, it does not express a basic value.*

Definition 2.5 (Property). *A property is an IRI resource that expresses a relation between a subject and an object [74].*

Definition 2.6 (RDF Triple). *Let $\mathfrak{I}$, $\mathfrak{B}$, and $\mathfrak{L}$ be pairwise disjoint infinite sets representing IRI resources, blank nodes, and literals, respectively. Let $\mathfrak{P} \subseteq \mathfrak{I}$ be the set of all properties. A triple $(s, p, o) \in (\mathfrak{I} \cup \mathfrak{B}) \times \mathfrak{P} \times (\mathfrak{I} \cup \mathfrak{B} \cup \mathfrak{L})$ is called an RDF triple. In this tuple, s is the subject, the property p is the predicate, and o is the object [74, 119].*[5]

Let T be a set of RDF triples. It should be noted that this set comes with a set of resources $R \subset \mathfrak{I}$, properties $P \subset \mathfrak{P}$, blank nodes $B \subset \mathfrak{B}$, and literals $L \subset \mathfrak{L}$. These sets can be derived from the set of triples as follows:

$$R = \{s | (s, p, o) \in T \wedge s \in \mathfrak{I}\} \cup \{o | (s, p, o) \in T \wedge o \in \mathfrak{I}\}, \tag{2.1}$$

$$P = \{p | (s, p, o) \in T\}, \tag{2.2}$$

$$B = \{s | (s, p, o) \in T \wedge s \in \mathfrak{B}\} \cup \{o | (s, p, o) \in T \wedge o \in \mathfrak{B}\}, \tag{2.3}$$

$$L = \{o | (s, p, o) \in T \wedge o \in \mathfrak{L}\}. \tag{2.4}$$

Definition 2.7 (Knowledge Graph). *A knowledge graph $\mathcal{G}$ is a representation of a set of triples as a directed edge-labelled multigraph [131].*[6] *Each triple is represented as directed, single-labeled edge between its subject and object. The triple's predicate is the label of the edge. Let V be the set of nodes within the knowledge graph with $V = R \cup B \cup L$ [74]. Then, the knowledge graph can be defined as a pair of the set of nodes and the set of edges, i.e., triples, $\mathcal{G} = (V, T)$.*

It should be noted that within this work, we do not differentiate between a triple and a directed, labeled edge since their representation would be the same. We will name sets that belong to a certain knowledge graph by using the knowledge graph as subscript where appropriate, e.g., $V_{\mathcal{G}}$ is the set of nodes of $\mathcal{G}$.

[5] Most resources define the space of RDF triples as $(\mathfrak{I} \cup \mathfrak{B}) \times \mathfrak{I} \times (\mathfrak{I} \cup \mathfrak{B} \cup \mathfrak{L})$ [74]. We use $\mathfrak{P}$ for the predicate position as it directly follows from the W3C recommendation "RDF 1.1 Semantics" [119] which defines that each IRI that is used in the predicate position has the type `rdf:Property` and the usage of a set of properties fits better to the research discussed within this thesis.

[6] It should be noted that there exist various definitions of knowledge graphs [47, 131]. Hogan et al. [131] refer to the usage of hypernodes and hypergraphs with complex edges. Bonifati et al. [47] discuss various extensions to property graphs to represent knowledge. However, for the goals of this thesis, the given definition is sufficient.

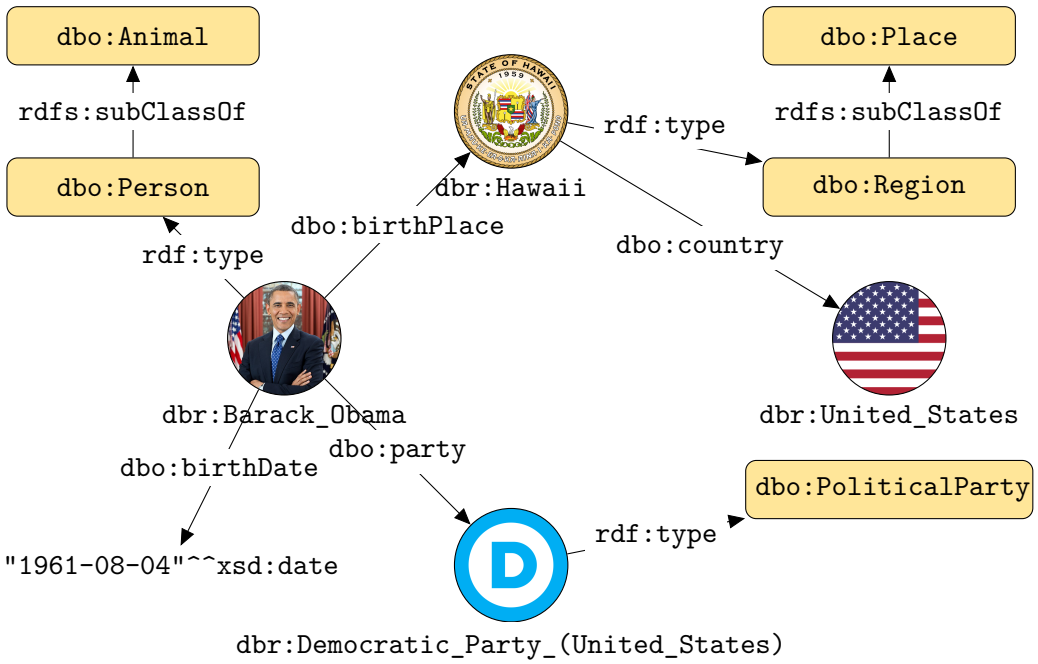


Figure 2.1.: Example of an RDF knowledge graph based on an excerpt of the DBpedia [21, 162]. IRIs have been shortened by using prefixes. We will refer to this graph as $\mathcal{G}_{\text{ex}}$.

Figure 2.1 depicts a small example knowledge graph. It comprises several IRIs, e.g., `dbr:Barack_Obama`.[7] A literal can be found in the lower left corner. It has the type `xsd:date` and represents Barack Obama's birth date August 4th 1961. Triples are depicted as arrows that connect two nodes. The predicate of the triple is printed on the arrow.

Definition 2.8 (Classes and instances)**.** *Nodes in a knowledge graph can be organized in groups called classes [52]. The Entities within such a group are called instances of this class. The relationship between an instance and the class is expressed by triples with the* `rdf:type` *property [52]. Each class is an instance of* `rdfs:Class` *and can be an instance of further classes [52]. Let $\mathfrak{C}$ be the global set of classes with $\mathfrak{C} \subset \mathfrak{I} \cup \mathfrak{B}$.*

Within this thesis, we will use C to denote a set of classes ($C \subset \mathfrak{C}$) and $C_{\mathcal{G}}$ if we want to emphasize that the classes belong to a certain knowledge graph $\mathcal{G}$. Let $\mathfrak{c} : V \to 2^{\mathfrak{C}}$ be a mapping function that derives for a given node $v \in V$ the set of all classes. In practice, $\mathfrak{C}$ is not known and $\mathfrak{c}$ is limited to the knowledge that is available, e.g., within a given knowledge graph. Hence, we use $\mathfrak{c}_{\mathcal{G}} : V_{\mathcal{G}} \to 2^{C_{\mathcal{G}}}$ to express this. For

[7] We use the prefixes `dbo`, `dbr`, `rdf`, `rdfs`, and `xsd` for the IRIs `http://dbpedia.org/ontology/`, `http://dbpedia.org/resource/`, `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, and `http://www.w3.org/2001/XMLSchema#`, respectively.

our example graph $\mathcal{G}_{\text{ex}}$, $\mathfrak{c}_{\mathcal{G}_{\text{ex}}}(\texttt{dbr:Democratic\_Party\_(United\_States)})$ gives the set $\{\texttt{dbo:PoliticalParty}\}$.

Let $\mathfrak{i} : \mathfrak{C} \rightarrow 2^V$ be a mapping function that derives all instances of a given class. As for $\mathfrak{c}$, the mapping is limited to a given knowledge graph when used in practice. We will express this by using $\mathfrak{i}_{\mathcal{G}} : C_{\mathcal{G}} \rightarrow 2^{V_{\mathcal{G}}}$ throughout this thesis. In the example, $\mathfrak{i}_{\mathcal{G}_{\text{ex}}}(\texttt{dbo:PoliticalParty})$ gives the set $\{\texttt{dbr:Democratic\_Party\_(United\_States)}\}$ since only one instance of this class is present in the example graph.

Definition 2.9 (Subclass)**.** *A class c_1 can be defined as* subclass *of another class c_2. In this case, all instances of c_1 are also instances of c_2 [52]. This can be formalized as follows:*

$$\forall v \in \mathfrak{i}_{\mathcal{G}}(c_1) \Rightarrow c_2 \in \mathfrak{c}_{\mathcal{G}}(v) \tag{2.5}$$

In the example graph, `dbo:Person` and `dbo:Region` are subclasses of `dbo:Animal` and `dbo:Place`, respectively. Since `dbr:Hawaii` is a `dbo:Region`, it can be inferred that $\mathfrak{c}_{\mathcal{G}_{\text{ex}}}(\texttt{dbr:Hawaii}) = \{\texttt{dbo:Region}, \texttt{dbo:Place}\}$. In the same way, we can infer that $\mathfrak{c}_{\mathcal{G}_{\text{ex}}}(\texttt{dbr:Barack\_Obama}) = \{\texttt{dbo:Person}, \texttt{dbo:Animal}\}$.

A property may define further restrictions with respect to the types that a subject and an object have by defining a domain and a range.

Definition 2.10 (Domain)**.** *The domain of a property p is a set of classes defined using the `rdfs:domain` property. If s is the subject of a triple with p as predicate, it can be inferred that s is an instance of all classes in the domain of p [52].*

Definition 2.11 (Range)**.** *The range of a property p is a set of classes defined using the `rdfs:range` property. If o is the object of a triple with p as predicate, it can be inferred that o is an instance of all classes in the domain of p [52].*

When dereferencing the properties used in our example graph, we can derive that `dbo:birthDate` and `dbo:birthPlace` have the domain `dbo:Animal`. However, `dbo:birthDate` has the range `xsd:date` while `dbo:birthPlace` has the range `dbo:Place`.[8] Based on the different ranges, we define define two classes of properties.[9] Let $\mathfrak{d}(p)$ and $\mathfrak{r}(p)$ be functions that retrieve the domain and range of the given property, respectively.

[8] The deferencing was done on August 1st, 2022. Later calls of the same IRIs may lead to different domain and range information.

[9] The class of annotation properties [124] is not mentioned since it will not be used within this thesis.

Definition 2.12 (Object property)**.** *An object property connects two entities [124]. We define the set of object properties as follows:*

$$P_R = \{p_i \,|\mathfrak{r}(p_i) \subset \mathfrak{I} \cup \mathfrak{B}\} \tag{2.6}$$

Definition 2.13 (Datatype property)**.** *A datatype property assigns a data value to an entity [124]. We define the set of datatype properties as follows:*

$$P_L = \{p_i \,|\mathfrak{r}(p_i) \subset \mathfrak{L}\} \tag{2.7}$$

In the example graph, `dbo:birthDate` is the only datatype property. All other properties are object properties.

RDF can be processed in various ways. However, the W3C offers the SPARQL Protocol And RDF Query Language (SPARQL) [17]. This language is designed to query and manipulate RDF data [17]. We define a SPARQL endpoint as a service that answers SPARQL queries. A software which stores RDF data and enables its querying is called triple store throughout this thesis. Further details of SPARQL are no prerequisites for this thesis.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a probabilistic topic model [43]. Topic models are used to structure large text corpora by identifying topics and assigning these topics to the documents in an unsupervised way. An example application is the exclusion (or inclusion) of documents from a large collection in an investigation based on their topics [51]. The topics can be seen as latent dimensions that structure the large corpora. Formally, they are distributions over words. However, for humans, they are typically represented as a set of words that are most important to that specific topic.

Before going into the details of probabilistic topic modeling, we define some terms. Let D be a corpus, i.e., a collection of documents. Let a document d be an ordered bag of words $d = \{w_1, w_2, \ldots\}$. Following Jurafsky et al. [141], we distinguish between word tokens and word types within a corpus. Let the set of words that occur within D be the corpus' vocabulary $\mathbb{V}_D$. The elements of $\mathbb{V}_D = \{\mathrm{w}_1, \mathrm{w}_2, \ldots\}$ are called word types. In contrast, word tokens are the occurrences of the word types within the documents of the corpus. Hence, it is possible that two word tokens

w_i and w_j are part of the same document d and share the same word type but are still two distinct tokens. For example, the following example document has 14 word tokens but only 11 word types since the types “invention”, “a”, and “door” occur twice:[10]

The cat flap; invention, pure creative invention. It is a door within a door.

In this section, LDA—one of the best researched topic models—is described.[11] In the following, the generative model of LDA is presented, before the inference algorithms are described. In Section 2.2.3, the problem to identify the number of latent topics is briefly discussed.

2.2.1 Generative Model

LDA is a generative model for the creation of natural language documents [43]. This process is based on probabilistic sampling rules [266] and the following assumptions [43]:

- Every topic is defined as a distribution over word types (ϕ) with higher probabilities for words that are essential for the topic.
- A document is a mixture of topics. Thus, it has a distribution over topics θ.

The generation of a corpus D based on a given vocabulary $\mathbb{V}_D$ as well as the hyper parameters α and β is defined as follows [41]:

1. Create the set of topics by sampling a discrete distribution over word types ϕ for every topic using a Dirichlet distribution (Dir) and a single prior β.[12] For the i-th topic, this is defined as

$$\phi_i \sim \text{Dir}(\beta). \tag{2.8}$$

2. Create every single document d of the corpus using the following steps.

[10] We ignore punctuation characters within the example. The example document is an adapted quote from Douglas Adam’s book “Dirk Gently’s Holistic Detective Agency”.

[11] The interested reader is referred to Blei [41] for a more detailed introduction and Boyd-Graber et al. [51] for a more recent overview of the topic modeling research field.

[12] A single number as prior for the Dirichlet distribution means that all needed priors have the same given value. These priors are sometimes also called symmetric prior since they create a symmetric Dirichlet distribution [51]. Although it is possible to use an asymmetric prior (i.e., β would be a vector), Wallach et al. [300] show that a symmetric prior leads to better results.

a) Sample a discrete distribution over topics θ using a Dirichlet distribution and a prior $\alpha = \{\alpha_1, \ldots\}$ with one prior value per topic.[13] For the j-th document, this is defined as

$$\theta_j \sim \text{Dir}(\alpha). \tag{2.9}$$

b) For every word token w in the document, choose a topic that creates it by sampling a topic index z from θ_j. After that, sample a word type for this token from the ϕ_z distribution of the z-th topic as follows:

$$z \sim \theta_j \,, \tag{2.10}$$

$$\mathrm{w} \sim \phi_z \,. \tag{2.11}$$

Let $|D|$ be the number of documents and $|d_i|$ the length of document d_i, i.e., the number of word tokens in that document. Let $w_{i,j}$ be the j-th word token in the i-th document, let $\mathrm{w}_{i,j}$ be its word type, and let $z_{i,j}$ denote the id of the topic from which the word type of this token has been sampled. Let ϱ be the number of topics and let $Z = \left\{ z_{1,1}, \ldots, z_{|D|,|d_{|D|}|} \right\}$ be the set of the topic indices of all word tokens in the corpus D. Let $\Phi = \{\phi_1, \ldots, \phi_\varrho\}$ be the set of word distributions and $\Theta = \{\theta_1, \ldots, \theta_{|D|}\}$ be the set of topic distributions. The generative process defines a joint distribution that is defined as follows [41]:

$$\mathbb{P}(\Phi, \Theta, Z, D) = \left(\prod_{i=1}^{\varrho} \mathbb{P}(\phi_i) \right) \left(\prod_{i=1}^{|D|} \mathbb{P}(\theta_i) \left(\prod_{j=1}^{|d_i|} \mathbb{P}(z_{i,j} | \theta_i) \, \mathbb{P}\left(\mathrm{w}_{i,j} \left| \phi_{z_{i,j}} \right. \right) \right) \right) . \tag{2.12}$$

Figure 2.2 shows the model using the plate notation. The figure as well as Equation 2.12 show several assumptions of LDA [41]:

- The words have no influence on their neighbors within a document, i.e., LDA relies on the assumption that the order of words within the documents has no influence. This assumption is also known as the "bag of words" assumption.
- The documents have no influence on each other. This can hold in some situations. However, when processing documents from a longer period this assumption may not hold.
- The number of topics ϱ is known. It is a parameter to the model and not derived from the data. This will be further discussed in Section 2.2.3.

[13] While it is also possible to use a symmetric α prior, Wallach et al. citeWallach2009 showed that using an asymmetric α and a symmetric β leads to better results.

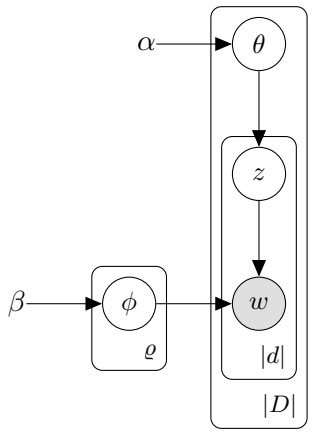


Figure 2.2.: LDA in plate notation [266]. White circles are hidden variables while the gray circle is the only observed variable (i.e., the words in a document). The plates denote repetitions, e.g., the $|d|$ plate denotes the collection of words in document d [41].

Equation 2.12 also shows an advantage of LDA. The usage of the Dirichlet distribution for defining ϕ and θ leads to the concentration of the distributions on single elements with high probabilities if the hyper parameters α and β have been chosen accordingly [51]. This sparsity means that a topic is more likely to have a small amount of words that have a high probability within that topic. In the same way, documents are likely to have a small amount of topics with a high probability [51]. Since both are combined by the last product in Equation 2.12, they act like contradicting goals since the distributions have to be chosen in a way that they concentrate on single elements but at the same time all documents of the given corpus can be sampled from them [42].

2.2.2 Inference

Figure 2.2 shows that only the word tokens w are observable. All other elements are hidden and have to be derived from the observed word tokens. This can be formulated as finding the model that maximizes the following posterior [41]:[14]

$$\mathbb{P}\left(\Phi, \Theta, Z \middle| D\right) = \frac{\mathbb{P}\left(\Phi, \Theta, Z, D\right)}{\mathbb{P}\left(D\right)}. \tag{2.13}$$

While the numerator is the joint distribution of all variables in a concrete model the denominator is the marginal probability of the given corpus under any topic model and, hence, intractable to compute [41]. Several inference algorithms have

[14] We skipped the hyper parameters α and β for simplicity.

been developed, that try to estimate the hidden distributions [43, 110, 126]. Within this work, we make use of two inference algorithms. The first is based on Gibbs sampling [110, 176] while the second uses Variational Bayesian inference [126].

Gibbs Sampling

This approach concentrates on the topic assignments Z and derives Φ as well as Θ from them by counting the assignments. Let $\zeta_{i,k}$ be the count of tokens in document d_i that have been assigned to the k-th topic, i.e.,:

$$\zeta_{i,k} = |\{z_{i,j} | z_{i,j} = k \wedge 0 < j \leq |d_i|\}| \ . \tag{2.14}$$

The topic distribution θ_i of the i-th document can be derived based on these counts by calculating the probability for each topic as follows [51]:

$$\theta_{i,k} = \mathbb{P}\left(z_{i,*} = k\right) \approx \left(\frac{\zeta_{i,k} + \alpha_k}{\sum_{j=1}^{\varrho} \zeta_{i,j} + \alpha_j}\right) . \tag{2.15}$$

where α_k is the k-th prior in α and $*$ denotes a wildcard, i.e., the position of the token in the document has no influence. Let $\eta_{k,\mathrm{w}}$ be the count of all occurrences of the word type w in the corpus at which it has been assigned to the k-th topic. We define this as follows:

$$\eta_{k,\mathrm{w}} = |\{z_{i,j} | z_{i,j} = k \wedge w_{i,j} = \mathrm{w} \wedge 0 < i \leq |D| \wedge 0 < j \leq |d_i|\}| \ . \tag{2.16}$$

Then, the word distribution of the k-th topic can be derived from these counts by calculating the probability as follows [51]:

$$\phi_{k,\mathrm{w}} = \mathbb{P}\left(w_{i,j} = \mathrm{w} \,|z_{i,j} = k\right) \approx \left(\frac{\eta_{k,\mathrm{w}} + \beta}{\sum_{\mathrm{w}' \in \mathbb{V}_D} \eta_{k,\mathrm{w}'} + \beta}\right) . \tag{2.17}$$

The inference algorithm starts by randomly initializing Z, i.e., all word tokens in the corpus are randomly assigned to a topic. After that, the algorithm iterates over all single assignments and updates them based on all other counts. Let $\tilde{Z}_{i,j} = Z \setminus \{z_{i,j}\}$ denote all topic assignments except the assignment $z_{i,j}$ that is currently updated. For this update, the following probability is calculated for each topic [51, 308]:

$$\mathbb{P}\left(z_{i,j} = k \,\middle|\, \tilde{Z}_{i,j}, w_{i,j} = \mathrm{w}\right) = \left(\frac{\zeta_{i,k} + \alpha_k}{\sum_{j=1}^{\varrho} \zeta_{i,j} + \alpha_j}\right) \left(\frac{\eta_{k,\mathrm{w}} + \beta}{\sum_{\mathrm{w}' \in \mathbb{V}_D} \eta_{k,\mathrm{w}'} + \beta}\right) . \tag{2.18}$$

We use $\tilde{Z}_{i,j}$ in the equation to emphasize that $z_{i,j}$ is not taken into account for all ζ and η counts. Based on the probabilities calculated for each topic, a new topic is sampled and assigned to $z_{i,j}$. This is carried out for all topic assignments Z in the corpus to finish a single iteration. After that, it is repeated until a maximum number of iterations is reached. Additionally, we use hyper parameter optimisation to periodically update α and β during inference as suggested by Wallach et al. [300].

Variational Bayesian Inference

The Variational Bayesian approach approximates the intractable posterior using a simpler distribution $q(\Phi, \Theta, Z)$ [43, 126]. This distribution is restricted to a factorized form, i.e., the parameters are assumed to be independent from each other [20]. To this end, the variational parameters $\Gamma = \{\gamma_{1,1}, \ldots, \gamma_{|D|,\varrho}\}$, $X = \{\chi_{1,1}, \ldots, \chi_{\varrho,|\mathbb{V}_D|}\}$ and $\Xi = \left\{\xi_{1,1,1}, \ldots, \xi_{|D|,|\mathbb{V}_D|,\varrho}\right\}$ are introduced [43, 126]:

$$q(\Phi, \Theta, Z) = \left(\prod_{k=1}^{\varrho} q(\phi_k)\right)\left(\prod_{i=1}^{|D|} q(\theta_i)\right)\left(\prod_{i=1}^{|D|}\prod_{j=1}^{|d_i|}\prod_{k=1}^{\varrho} q(z_{i,j}=k)\right), \tag{2.19}$$

$$q(\theta_i) = \mathrm{Dir}(\theta_i|\gamma_i)\,, \tag{2.20}$$

$$q(\phi_k) = \mathrm{Dir}(\phi_k|\chi_k)\,, \tag{2.21}$$

$$q(z_{i,j}=k) = \xi_{i,\mathrm{w}_{i,j},k}\,. \tag{2.22}$$

Let $\mathbb{E}_q$ denote the expected value according to the distribution q. To approximate the posterior, the following Evidence Lower Bound (ELBO) has to be maximized [126]:

$$\log(\mathbb{P}\,(D|\,\alpha,\beta)) \geq \mathcal{L}\,(D,\Xi,\Gamma,X) \triangleq \mathbb{E}_q\,(\mathbb{P}\,(\Phi,\Theta,Z,D|\,\alpha,\beta)) - \mathbb{E}_q\,(q(\Phi,\Theta,Z))\,. \tag{2.23}$$

Hoffman et al. [126] point out that maximizing the ELBO is equivalent to minimizing the KL divergence between the intractable posterior $\mathbb{P}\,(\Phi,\Theta,Z|\,D)$ and the simpler distribution $q(\Phi,\Theta,Z)$. The ELBO can be factorized into the following form [126]:

$$\begin{aligned}\mathcal{L}\,(D,\Xi,\Gamma,X) = \sum_{i=0}^{|D|}\Big(&\mathbb{E}_q\,(\mathbb{P}\,(d_i|\,\theta_i,Z_i,\Phi)) + \mathbb{E}_q\,(\mathbb{P}\,(Z_i|\,\theta_i))\\ &-\mathbb{E}_q\,(q(Z_i)) + \mathbb{E}_q\,(\mathbb{P}\,(\theta_i|\,\alpha)) - \mathbb{E}_q\,(q(\theta_i))\\ &+\frac{1}{|D|}\Big(\mathbb{E}_q\,(\mathbb{P}\,(\Phi|\,\beta)) - \mathbb{E}_q\,(q(\Phi))\Big)\Big)\,,\end{aligned} \tag{2.24}$$

where Z_i denotes the topic indices of the word tokens in document d_i. Hoffman et al. [126] further transform this equation by expanding the expected values

to be functions of the variational parameters. This leads to the insight, that the likelihood can be optimized by updating the variational parameters. Let $\psi_{i,\mathrm{w}}$ be the number of times word type w occurs in document d_i and let Ψ denote the digamma function. Hoffman et al. [126] define the following three update functions for the variational parameters:

$$\xi_{i,\mathrm{w},k} \propto \exp\left(\Psi(\gamma_{i,k}) - \Psi\left(\sum_{j=0}^{\varrho}\gamma_{i,j}\right) + \Psi(\chi_{k,\mathrm{w}}) - \Psi\left(\sum_{\mathrm{w}'\in\mathbb{V}_D}\chi_{k,\mathrm{w}'}\right)\right), \tag{2.25}$$

$$\gamma_{i,k} = \alpha_k + \sum_{\mathrm{w}'\in\mathbb{V}_D}\psi_{i,\mathrm{w}'}\xi_{i,\mathrm{w}',k}\,, \tag{2.26}$$

$$\chi_{k,\mathrm{w}} = \beta + \sum_{i=0}^{|D|}\psi_{i,\mathrm{w}}\xi_{i,\mathrm{w},k}\,. \tag{2.27}$$

Similar to the Expectation Maximization approach, the update functions can be separated into two steps [43, 126]. During the "E" step, Γ and Ξ are optimized while X is treated as a constant. In the "M" step, X is optimized based on the values of Γ and Ξ. Hoffman et al. [126] further enhance this algorithm by exploiting their formulation of the ELBO as a sum over the documents in Equation 2.24. This allows to formulate the update functions in a way that enables the usage of mini batches, i.e., subsets of D. In addition to that, it should be noted that the update functions rely on $\psi_{i,\mathrm{w}}$ but not on Z. This means that only word counts are needed for the inference instead of the single word tokens [126]. Both advantages—mini batches and relying on word counts—make the Variational Bayesian approach interesting when using large corpora.

2.2.3 Number of Topics

As described above, a central assumption of LDA is that the number of topics ϱ is given as a parameter. If this number is too low, the topic model is not able to describe the complexity of the training data. If it is too high, one of the model's main assumptions, i.e., the orthogonality of the topics, will not hold anymore. Thus, picking a good number of topics has a high influence on the model's performance. However, there is no general applicable method to determine a good number of topics for a given corpus [19]. In this section, we present two different methods that we will apply throughout this thesis. Both methods suggest to generate topic models with different numbers of topics. After the generation, the single models are evaluated regarding their quality using different approaches [19, 110].

The first approach is the calculation of $\mathbb{P}(D|\Phi)$ proposed by Griffiths et al. [110]. This probability shows how likely it is that the model could generate the corpus on that it has been trained. The model that maximizes this probability is the best performing model and, hence, has the best number of topics. However, this probability is intractable since the probabilities over all possible combinations of topic assignments Z would have to be summed up. To this end, Griffiths et al. present an approximation by calculating the harmonic mean of a set of $\mathbb{P}(D|Z,\Phi)$ where Z are topic assignments that are sampled from the posterior $\mathbb{P}(Z|D,\Phi)$.

The second approach has been proposed by Arun et al. [19] and is based on the observation that LDA can be regarded as a non-negative matrix factorization. This factorization transforms the corpus Matrix M of order $|D| \times |\mathbb{V}_D|$ into two matrices M_1 of order $|D| \times \varrho$ and M_2 of order $\varrho \times |\mathbb{V}_D|$ where D is the set of documents, $\mathbb{V}_D$ is the vocabulary and ϱ is the number of topics. The proposed measure—which we will call $\mathcal{A}$ throughout this thesis—is based on the idea that the sum of assignments to the single topics have to be the same in both matrices. Since the rows of both matrices represent probability distributions and are thus normalized, these sums cannot be used directly. Let KL be the Kullback-Leibler divergence, $\mathfrak{h}_1$ the distribution of singular values of M_1, $l = \{|d_i| \,|\, 0 < i \leq |D|\}$ a vector containing the lengths of the single documents and $\mathfrak{h}_2 = l \times M_2$. Then, $\mathcal{A}$ is defined as

$$\mathcal{A}(M_1, M_2) = \mathrm{KL}(\mathfrak{h}_1||\mathfrak{h}_2) + \mathrm{KL}(\mathfrak{h}_2||\mathfrak{h}_1) \ . \tag{2.28}$$

Arun et al. [19] predict that with an increasing number of topics the values of $\mathcal{A}$ will decrease until a minimum is reached before the measure's value to increases again. They argue that the lowest point inside this dip is created by the model with the best number of topics [19].

2.3 Measures

Several measures are used throughout this thesis. This section introduces common measures that are used in more than one chapter.

2.3.1 Pointwise Mutual Information

Let $\mathcal{X}$ and $\mathcal{Y}$ be two random variables. Their values have the marginal probabilities $\mathbb{P}(x)$ and $\mathbb{P}(y)$, and the joint probability $\mathbb{P}(x,y)$. The Pointwise Mutual Information

(PMI) is defined as the logarithm of the ratio between the measured joint probability of two random values and the joint probability they should have under the assumption that they are independent [49], i.e.,:

$$\text{PMI}(x, y) = \log\left(\frac{\mathbb{P}(x, y)}{\mathbb{P}(x)\mathbb{P}(y)}\right). \tag{2.29}$$

The codomain of the PMI measure is $[-\infty, \infty]$. If the independence assumption between $\mathcal{X}$ and $\mathcal{Y}$ holds, the PMI measure has the result 0. If the measure returns a positive result, the two values cooccur more often than by chance. A negative value indicates that the values occur less often together than by chance.

If the probabilities are based on counts that are retrieved from some reference data, a small value ϵ can be added to the nominator to avoid the calculation of the logarithm of 0 [265]. Within this thesis, we will mark this variant as PMI_ϵ, which is defined as:

$$\text{PMI}_\epsilon(x, y) = \log\left(\frac{\mathbb{P}(x, y) + \epsilon}{\mathbb{P}(x)\mathbb{P}(y)}\right). \tag{2.30}$$

Bouma [49] suggests a normalized variant of this measure—the Normalized Pointwise Mutual Information (NPMI). It is defined as follows:

$$\text{NPMI}(x, y) = \frac{\text{PMI}(x, y)}{-\log(\mathbb{P}(x, y))} = \frac{\log\left(\frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)}\right)}{-\log(\mathbb{P}(x, y))}. \tag{2.31}$$

As the PMI, the NPMI measure returns a 0 in case the values x and y are independent of each other. However, its maximum value is 1 while its minimum value is -1, which allows an easier interpretability of the result [49]. Similar to PMI_ϵ, the variant NPMI_ϵ uses a small constant ϵ to avoid the logarithm of 0 and is defined as:

$$\text{NPMI}_\epsilon(x, y) = \frac{\text{PMI}_\epsilon(x, y)}{-\log(\mathbb{P}(x, y) + \epsilon)} = \frac{\log\left(\frac{\mathbb{P}(x,y)+\epsilon}{\mathbb{P}(x)\mathbb{P}(y)}\right)}{-\log(\mathbb{P}(x, y) + \epsilon)}. \tag{2.32}$$

2.3.2 Precision, Recall, and F1-measure

We will use precision, recall, and F1-measure in cases in which the result of an algorithm can be compared to a given ground truth and the result of the comparison is a binary value, i.e., whether the result calculated by the evaluated algorithm fits to the ground truth. Figure 2.3 shows the schema of a confusion matrix, which can

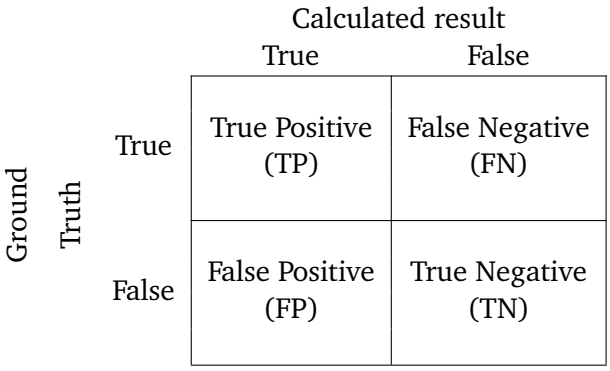


Figure 2.3.: Schema of a confusion matrix [93].

be used to represent the result of such comparisons. It comprises 4 counts that are defined as follows [93]:

- True positive (TP): the number of elements that have been identified as true by the algorithm and are marked as true in the ground truth.
- True negative (TN): the number of elements that have been identified as false by the algorithm and are marked as false in the ground truth.
- False positive (FP): the number of elements that have been falsly identified as true by the algorithm but are marked as false in the ground truth.
- False negative (FN): the number of elements that have been falsly identified as false by the algorithm but are marked as true in the ground truth.

Based on these counts, several measures can be used to describe the performance of the evaluated algorithm. Precision (Pr) is defined as the amount of correct results in the set of results returned by the evaluated algorithm [93]:

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} . \tag{2.33}$$

Recall (Re) is defined as the number of correct results returned by the evaluated algorithm in comparison to the overall number of positive elements [93]:

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} . \tag{2.34}$$

The F1-measure (F1) can be used to summarize the precision and the recall of an algorithm. It is defined as the harmonic mean of precision and recall [93, 100]:

$$\mathrm{F1} = \frac{2\mathrm{PrRe}}{\mathrm{Pr} + \mathrm{Re}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} . \tag{2.35}$$

2.3.3 Micro and Macro Averages

When evaluating an algorithm using several datasets multiple confusion matrices are generated—one for each dataset. Within this thesis, we distinguish two main ways to calculate a summarizing value for precision, recall, and F1-measure in this scenario. The micro measures are based on a confusion matrix that sums up the single TP, TN, FP and FN counts over all datasets. Let n be the number of datasets and let TP_i, TN_i, FP_i and FN_i be the TP, TN, FP, and FN of the i-th dataset, respectively. We define micro precision ($\mathrm{Pr}_{\mathrm{mic}}$), micro recall ($\mathrm{Re}_{\mathrm{mic}}$), and micro F1-measure ($\mathrm{F1}_{\mathrm{mic}}$) as follows [100, 141]:[15]

$$\mathrm{Pr}_{\mathrm{mic}} = \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FP}_i}\,, \tag{2.36}$$

$$\mathrm{Re}_{\mathrm{mic}} = \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FN}_i}\,, \tag{2.37}$$

$$\mathrm{F1}_{\mathrm{mic}} = \frac{2 \times \mathrm{Pr}_{\mathrm{mic}} \times \mathrm{Re}_{\mathrm{mic}}}{\mathrm{Pr}_{\mathrm{mic}} + \mathrm{Re}_{\mathrm{mic}}} = \frac{2 \times \sum_{i=1}^{n} \mathrm{TP}_i}{2 \times \sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FP}_i + \sum_{i=1}^{n} \mathrm{FN}_i}\,. \tag{2.38}$$

Let Pr_i, Re_i, and $\mathrm{F1}_i$ be the precision, recall, and F1-measure calculated based on the results for the i-th dataset, respectively. A macro average is the arithmetic average of the measures calculated per dataset. Hence, we define macro precision ($\mathrm{Pr}_{\mathrm{mac}}$), macro recall ($\mathrm{Re}_{\mathrm{mac}}$), and macro F1-measure ($\mathrm{F1}_{\mathrm{mac}}$) as follows [100, 141]:[16]

$$\mathrm{Pr}_{\mathrm{mac}} = \sum_{i=1}^{n} \mathrm{Pr}_i = \sum_{i=1}^{n} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}\,, \tag{2.39}$$

$$\mathrm{Re}_{\mathrm{mac}} = \sum_{i=1}^{n} \mathrm{Re}_i = \sum_{i=1}^{n} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}\,, \tag{2.40}$$

$$\mathrm{F1}_{\mathrm{mac}} = \sum_{i=1}^{n} \mathrm{F1}_i = \sum_{i=1}^{n} \frac{2 \times \mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}\,. \tag{2.41}$$

[15] Forman et al. [100] use the name $F_{\mathrm{tp,fp}}$ for the micro F1-measure.
[16] Forman et al. [100] use the name F_{avg} for the macro F1-measure.

3 Benchmarking Linked Data Systems

While the adoption of Linked Data is increasing steadily, the selection of the right frameworks for a given application driven by this paradigm remains elusive. This is partly due to the lack of 1) large-scale benchmarks for most steps of the Linked Data life cycle [23] and 2) scalable benchmarking platforms able to generate uniform comparable evaluation results for the technologies which deal with this type of data [231]. The usefulness of benchmarks for characterising the performance of families of solutions has been clearly demonstrated by the varied benchmarks made available over recent decades [109, 231, 276, 281]. For example, the TPC family of benchmarks is widely regarded as having provided the foundation for the development of efficient relational databases [109]. Modern examples of benchmarks that have achieved similar effects include the QALD [281] and BioASQ [276] benchmarks, which have successfully contributed to enhancing the performance of question answering systems over Linked Data and in the bio-medical domain, respectively. Modern benchmarking platforms have also contributed to the comparability of measurements used to evaluate the performance of systems. For example, benchmarking platforms such as BAT [68], GERBIL [228, 231, 286, 287, 292], and IGUANA [67] provide implementations and corresponding theoretical frameworks to benchmark different aspects of the Linked Data life cycle in a consistent manner. Still, *none of these benchmarking platforms can scale up to the requirements of benchmarking modern Big Linked Data applications*.

A major challenge of benchmarking Linked Data solutions is the ability of knowledge graphs to grow within a short period of time. For example, the Google Knowledge Graph grew from 3.5 billion facts to 18 billion facts in 7 months. Noy et al. [206] point out that comparable growth rates can be observed in knowledge graphs of other large companies. The same phenomenon is also present in open data sets. For example, DBpedia [21, 162] crossed the mark of 23 billion triples in 2017[1] while it begun with 0.1 billion triples in 2007 [21]. The ranking of corresponding storage

¶ Parts of this chapter have been published as journal and conference articles [231, 237, 240]. For all three publications, the author developed the main ideas, designed and implemented major parts of the solution, and wrote the majority of the publication.

[1] `https://wiki.dbpedia.org/develop/datasets/dbpedia-version-2016-10`

solutions w.r.t. their runtime performance has been observed to change with the size of the knowledge graphs [67, 209]. For example, Papakonstantinou et al. [209] report the performance of four triple stores on different versions of an RDF knowledge graph ranging from 10^5 to 10^6 triples. While BlazeGraph Free version 8.5 achieves the second-best performance on the smallest version of the knowledge graph, it achieves the worst performance on the subsequent version, which is merely five times larger. Similar insights can be derived from the work of Conrads et al. [67], where Jena TDB version 2.3.0 is ranked first across three triple stores and achieves the best performance on a 10% fragment of DBpedia version 2016-10 but is outperformed by Virtuoso version 7.0.0 on the full version of the same dataset and even achieves the worst performance in some high-load settings with 16 concurrent queries. Given that the performance of Linked Data systems like triple stores changes across dataset versions, there is a need to predict the future performance of Linked Data solutions given existing versions of a dataset. Such a prediction can facilitate the deployment of reliable knowledge graph infrastructures, the timely acquisition and alteration of software components and the maintenance of quality-of-service requirements. Benchmarking Linked Data systems hence faces the challenge of *predicting the performance and ranking of Linked Data systems on a (future) version of a knowledge graph given previous versions of the same dataset*. This task differs from that addressed by current RDF generators, which assume a particular dataset or ontology (e.g., universities) and generate data based thereupon [14, 16, 89, 114, 248, 272].

Another challenge for benchmarking Linked Data systems arises from the properties of Linked Data. Linked Data may be separated into several distinct knowledge graphs that are linked with each other. This leads to the situation that some systems may rely on one knowledge graph while other systems rely on another. At the same time, a benchmarking dataset may have been manually created with a third knowledge graph. A classic solution to this is to stick to a single knowledge graph. For example, the BAT framework [68] translates all IRIs and other identifiers to Wikipedia article IDs and runs its comparison only with these IDs. However, this limits the applicability of the BAT framework to systems and datasets that either rely on these IDs or which work with identifiers that can be translated into them. Instead, *the strengths of Linked Data should be used to provide a generic solution to this problem.*

This chapter has three main contributions. The first contribution is the presentation that Linked Data features can be used within benchmarking platforms to solve several challenges. To this end, we present an extension of GERBIL [228, 286, 287]—a platform presented by Usbeck et al. [287] to benchmark knowledge extraction systems. This extension enables the benchmarking of systems that have been

developed for a certain knowledge graph with a dataset that has been created based on a different knowledge graph by using links between these graphs. In addition, it reduces the impact of changes that may arise from the aforementioned growth of knowledge graphs on the validity of benchmarking datasets.

The second contribution is the HOBBIT (Holistic Benchmarking of Big Linked Data) platform [237]. HOBBIT was designed to accommodate the benchmarking of Big Linked Data applications, i.e., applications driven by Linked Data that exhibit Big Data requirements as to the volume, velocity, and variety of data they process [61]. The platform was designed with extensibility in mind. Thus, its architecture is modular and allows the benchmarking of any step of the Linked Data life cycle.[2] The comparability of results was the second main design pillar. Consequently, HOBBIT abides by the FAIR data principles [306]. The practical usability of the platform was ensured by its use in 13 challenges between 2016 and 2020 (e.g., [12, 13, 103, 104, 112, 113, 139, 197, 218, 235, 258, 259]). HOBBIT is open-source and can be deployed locally, on a local cluster, and on computing services such as Amazon Web Services. Additionally, we offer an online instance of the platform deployed on a cluster and available for experimentation.[3]

The third contribution is LEMMING—an approach to generate synthetic knowledge graphs of arbitrary size that mimic real-world knowledge graphs. It takes several versions of a knowledge graph as input to determine the characteristics of the knowledge graph. Based on these characteristics and gathered statistical information, LEMMING generates a synthetic knowledge graph with a given size and similar characteristics as the original graph. Hence, LEMMING can serve as a general purpose data generator that allows the benchmarking of Linked Data systems with respect to their scalability using synthetic data that has similar characteristics as real-world knowledge graphs.

The rest of this chapter is structured as follows: We begin by giving an overview of the state of the art in benchmarking Linked Data in Section 3.1. In Section 3.2, we present our extension to the GERBIL benchmarking platform. In Section 3.3, we present requirements for a benchmarking platform for Big Linked Data applications that were gathered from experts. We use these requirements to derive the architecture for the HOBBIT platform and present it in Section 3.4. Section 3.5 presents our approach to generate knowledge graphs that mimic real-world graphs. We demonstrate the use of the benchmarking platform and the knowledge graph generation in Section 3.6. We show how the benchmarking platform can be applied

[2] Code and dataset generators are available at `http://github.com/hobbit-project`. The project homepage can be found at `https://project-hobbit.eu/` (last accessed on 03.08.2022).

[3] `http://master.project-hobbit.eu`; last accessed on 03.08.2022.

to benchmark a knowledge extraction framework along the axes of accuracy and scalability—a dimension that was not considered in previous benchmarking efforts. We also present evaluation results that underline that our generated knowledge graphs are similar to held-out real-world knowledge graphs. We present the different applications of the benchmarking platform in Section 3.7. Finally, we discuss limitations and derive future work for our approaches in Section 3.8 before concluding this chapter with Section 3.9.

3.1 Related Work

3.1.1 Benchmarking

The work presented herein is mostly related to benchmarking platforms for RDF and Linked-Data-based systems. Several benchmarks have been developed in the area of linking RDF datasets [198]. A detailed comparison of instance matching benchmarks has been published by Daskalaki et al. [76]. They show that there are several benchmarks using either real or synthetically generated datasets. SEALS is the best-known platform for benchmarking link discovery frameworks.[4] It offers the flexible addition of datasets and measures for benchmarking link discovery. However, the platform was not designed to scale and can thus not deal with datasets which demand distributed processing.

For a large proportion of existing benchmarks and benchmark generators the focus has commonly been on creating frameworks able to generate data and query loads able to stress triple stores [39, 67, 114, 187, 243]. For example, the Lehigh University Benchmark [114] is a synthetic benchmark aiming to test triple stores and reasoners for their reasoning capabilities. SP^{2}Bench [247] is a synthetic benchmark for testing the query processing capabilities of triple stores. The Berlin SPARQL Benchmark [39] is a synthetic triple store benchmark based on an e-commerce use case in which a set of products is provided by a set of vendors and consumers post reviews regarding those products. SRBench [309] is an RDF benchmark designed for benchmarking streaming RDF/SPARQL engines. Przyjaciel-Zablocki et al. [221] propose a synthetic query benchmark centered around social network data. Other, similar benchmarks include the works of Aluç et al. [16], Morsey et al. [187, 188], Saleem et al. [243], Schmidth et al. [249], and Tarasova et al. [273]. IGUANA [67] is the first benchmarking framework for the unified execution of these data and

[4] `http://oaei.ontologymatching.org/2015/seals-eval.html`; last accessed on 03.08.2022.

query loads. However, like the platforms aforementioned, IGUANA does not scale up to distributed processing and can thus not be used to benchmark distributed solutions at scale.

Knowledge Extraction—especially Named Entity Recognition and Linking—has also seen the rise of a large number of benchmarks [231]. Several conferences and workshops aiming at the comparison of information extraction systems (including the Message Understanding Conference [268] and the Conference on Computational Natural Language Learning [275]) have created benchmarks for this task. In 2014, Carmel et al. [59] introduced one of the first Web-based evaluation systems for Named Entity Recognition and Linking. The BAT benchmarking framework [68] was also designed to facilitate benchmarking based on these datasets by combining seven Wikipedia-based systems and five datasets. The GERBIL framework [228, 231, 286, 287] extended this idea by being knowledge-base-agnostic and addressing the NIL error problem in the formal model behind the BAT framework. We will present these features of GERBIL in more detail in Section 3.2. However, while these systems all allow for benchmarking knowledge extraction solutions, they do not scale up to the requirements of distributed systems.

In the area of Question Answering using Linked Data, challenges such as BioASQ [276], and QALD [66, 279, 280, 281, 282, 283, 290, 291] aimed to provide benchmarks for retrieving answers to human-generated questions. The GERBIL-QA platform [292] is the first open benchmarking platform for question answering which abides by the FAIR data principles. It builds upon the aforementioned GERBIL platform. GERBIL is also used as a platform for several Semantic Web Challenges to evaluate the performance of fact validation systems [204, 216, 260]. In a similar way, the BENG platform [193] is an extension of GERBIL that is used in the area of natural language generation based on RDF data. However, like its knowledge extraction companion, both of them are not designed to scale up to large data and task loads.

Frameworks aiming at benchmarking in a generic fashion are very rare. The Peel framework[5] supports the automation of experiments on Big Data infrastructure. However, the framework only supports systems that can be executed on one of the supported Big Data solutions like Flink or Spark which excludes a lot of existing Linked Data benchmarks and systems.[6] Moreover, it does not support a large portion of the specific requirements for benchmarking Big Linked Data described in Section 3.3. A major drawback is that the results generated by the platform are not transparent as the execution of systems and benchmarks is hidden from the

[5] `http://peel-framework.org`; last accessed on 03.08.2022.

[6] The complete list can be found at `https://github.com/peelframework/peel#supported-systems`; last accessed on 03.08.2022.

users. This makes a comparison of the resources used by benchmarked systems impossible.

Also relevant according to the literature are novel Big Data benchmarks for benchmarking relational databases (e.g., BigBench [108] and OLTP [82]). However, although they come with scalable data and task generators, these benchmarks are solely focused on the benchmarking of relational databases and are not benchmarking frameworks.

A similar data generation-based approach is used by Plug and Play Bench [62]. However, in contrast to the other benchmarks, Plug and Play Bench aims at benchmarking different hardware settings on which the benchmark is executed instead of comparing different software solutions.

Table 3.1.: Comparison of Linked Data benchmarking frameworks, their applicability for all eight steps of the Linked Data life cycle and their support of features necessary for benchmarking Big Linked Data solutions.

	Year	Extraction	Storage	Manual Revision	Linking	Enrichment	Quality Analysis	Evolution	Exploration	Scalable Data	Scalable Tasks	FAIR Benchmarking
BAT [68]	2013	✓		—								
GERBIL [228, 231, 286, 287]	2014	✓		—								✓
GERBIL-QA [292]	2018			—					✓			✓
BENG [193]	2020	✓		—					✓			✓
IGUANA [67]	2017		✓	—							✓	
HOBBIT [237]	2017	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓

Table 3.1 compares the existing benchmarking frameworks used to benchmark Linked Data systems regarding their applicability for all eight steps of the Linked Data life cycle as well as their support of features necessary for benchmarking Big Linked Data solutions. The step "Manual Revision" is mentioned only for the completeness of the life cycle steps. The table shows that the HOBBIT platform is the first benchmarking framework which supports all steps of the Linked Data life cycle that can be benchmarked automatically. In addition, it is the first benchmarking platform for Linked Data which scales up to the requirements of Big Data platforms through horizontal scaling. The comparability of HOBBIT's benchmarking results are ensured by the cluster underlying the open instantiation of the platform.

3.1.2 RDF Graph Generation

The generation of synthetic graphs that can mimic real-world graphs is an important field of research. One of the first works in this field is the Erdös-Renyi model [88]. The model creates a random graph with a given number of nodes and a given number of edges by randomly assigning edges to the nodes. The model assumes that each edge is equally likely and that the edges can be sampled independently from each other. The degrees of the nodes follow a binomial distribution. In contrast to that, the Watts-Strogatz model [301] is able to create random graphs with small-world properties. It starts with connecting each node to a number of neighbouring nodes creating a lattice. After this first construction step, each edge can be removed and replaced by a new edge connecting a node with a completely different node based on a given probability. The Barabasi-Albert model [28] is able to create scale-free graphs similar to the link graph of the World Wide Web. The model adds one node after the other to the graph connecting them to the existing nodes based on the nodes degree, i.e., highly connected nodes have a higher probability to be chosen than others. There are several extensions of these models [153, 165]. Leskovec et al. [165] report that networks densify and shrink with respect to their diameter over time while Krioukov et al. [153] propose the usage of a hyperbolic geometry as basis to simulate networks. However, all these models and their extensions aim at general graphs and do not take special features of RDF graphs into account.

The Attribute Synthetic Generator [14] mimics social networks and takes different types of edges and features of nodes into account. It uses the preferential attachment model to assure the richer-get-richer phenomenon of nodes and the label homophily measure to control the creation of edges based on node labels. In addition, the generator applies a stochastic optimisation to fine-tune the feature distributions with respect to statistics of the original network like the node degree distribution. In a similar way, the Property Graph Model [245] takes node features and link types into account. In addition, it is able to scale the generated graphs to larger sizes. However, both approaches are not applicable for RDF as they create undirected graphs and take only a limited set of graph features into account.

There are several generators for RDF datasets. Theoharis et al. [274] propose an approach to generate synthetic schemas of RDF datasets. This is different to our work since we focus on the instance data and not the schema. The Lehigh University Benchmark [114] generates RDF graphs with a given number of triples describing synthetic universities, their lectures etc. The LDBC [89] generator creates RDF data describing a social network. In a similar way, SP2Bench [248] relies on the publication domain. The Waterloo SPARQL Diversity Test Suite [16] offers a data

generator for scalable RDF datasets relying on the WatDiv schema. PoDiGG [272] is an RDF generator for an artificial transport network based on a given population density. While all these generators create RDF graphs, they are bound to a certain domain or ontology.

Some approaches support the generation of RDF datasets independently of the dataset's domain. Grr [44] is a generator that relies on commands written in a domain specific language describing the single steps that are necessary to generate the dataset. In contrast, gMark [25] offers a more comfortable generator for an RDF dataset and a set of queries that can be used to benchmark the dataset. However, gMark needs a large amount of statistical information about the dataset including in and out degree distributions. Similarly, LinkGen [140] relies on a given ontology and a set of parameters including distribution parameters. Apart from that, LinkGen has never been evaluated with respect to the quality of the generated graphs. In comparison, the generator proposed in this chapter relies solely on the given RDF graphs without additional ontological data and gathers all statistical values that are needed for the generation process by itself. In addition, LEMMING is the first graph generation algorithm able to mimic real-world RDF datasets by determining necessary statistics and characteristic expressions that give invariant values for the given dataset.

3.2 Benchmarking with Linked Data

As described before, Linked Data may comprise several distinct, connected knowledge graphs that may grow over time. This creates two challenges for benchmarking Linked Data processing systems. First, different systems and benchmark datasets might be based on different knowledge graphs. For example, some systems may use DBpedia IRIs [284, 285] while other systems work with Wikipedia article IDs [68] or article titles [183]. In the area of knowledge extraction, Cornolti et al. [68] suggest to rely on Wikipedia article IDs. They implemented the BAT framework based on the idea to translate all IRIs and other identifiers to Wikipedia article IDs. However, while the comparison of identifiers becomes an easy comparison of numbers, the approach can only be used for entities that have an article in the Wikipedia. Usbeck et al. [228, 286, 287] took the idea of the BAT framework to benchmark named entity recognition and linking systems further by implementing GERBIL. In addition to various other enhancements, GERBIL does not rely on the comparison of Wikipedia article IDs but on the comparison of IRIs. While this removes the close coupling to the Wikipedia it still requires that the benchmarked

systems and benchmark datasets rely on the same knowledge graph. In practice, this led to the implementation of IRI translations within the single adapters that have been implemented for the different systems and datasets. These translations ensure that the IRI comparisons within GERBIL are always based on DBpedia IRIs. A generic solution that is not bound to a particular knowledge graph would be preferable.

A second challenge arises from the growth of knowledge graphs and changes that are applied to them over time. Many benchmarking platforms rely on gold standards that have been manually created. The creation of such gold standards is expensive since it involves the work of human experts [226, 259]. Hence, datasets that have been made available are reused to save costs and compare the performance of different systems on the same dataset. However, while the development of Linked Data systems moves on, many datasets were created years ago using older versions of knowledge graphs. Hence, the gold standard of a dataset may contain the IRI that may not exist anymore in the latest version the reference knowledge graph from which it was originally derived. Jah et al. [138] evaluated 13 datasets and found outdated IRIs in all of them. In 3 of the datasets, more than 10% of the IRIs needed an update. These IRIs must be identified and either updated or marked as outdated.

Within this Section, we tackle both challenges. Our extended version of GERBIL [231] is the first benchmarking platform for knowledge extraction that takes the special features of Linked Data into account. Within this section, we present the two following main features. First, GERBIL can bridge the gap between systems and datasets that have been created for different knowledge graphs.[7] Second, it can identify outdated and faulty IRIs within a dataset.

In the following, we briefly describe GERBIL. After that, we describe the extensions that we added.

3.2.1 GERBIL and D2KB

GERBIL is designed as a benchmarking platform for knowledge extraction systems, i.e., systems that take natural language as input and provide structured data as output. It supports different experiment types ranging from the spotting of named entities within a text, over their linking to a given reference knowledge graph, to the extraction of complete triples from the given text.[8] In the following, we focus on the evaluation of the entity disambiguation task (D2KB, also known as named

[7] We assume that links between these knowledge graphs exist and can be found by GERBIL.
[8] The interested reader is referred to Röder et al. [231] for a detailed description.

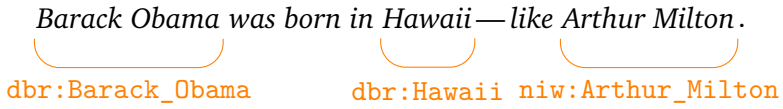


Figure 3.1.: An example document with three named entities and the IRIs of the ground truth.

entity linking). This task is defined as follows. A benchmarked system (dubbed annotator) receives a text that contains marked named entities. For these named entities, it should provide IRIs from a reference knowledge graph [68, 287]. The evaluation is mainly based on the comparison of the system's answer to the IRIs in the gold standard. An example for an input is shown in Figure 3.1. The text contains three named entities. Based on the example knowledge graph $\mathcal{G}_{\text{ex}}$ presented in Section 2.1.2, it is easy to assign the IRIs `dbr:Barack_Obama` and `dbr:Hawaii` to the first two named entities. However, there is no IRI in this graph that represents an entity named "Arthur Milton"—a person that might be known to the author of the example text but not to a broader group of people. Following Hoffart et al. [125], we name such an entity an emerging entity. For these entities, GERBIL generates a new IRI in a separate namespace. We us the `http://aksw.org/notInWiki` namespace with the prefix `niw` for that.[9] In the example, we assign the IRI `niw:Arthur_Milton` to the named entity "Arthur Milton". We will use these three IRIs as gold standard for the given example. Hence, the IRIs that would be assigned by a benchmarked annotation system would be compared to these three IRIs.

However, the comparison of two IRIs cannot easily be reduced to a simple string comparison. As described above, annotation systems use different reference knowledge graphs and IRIs from two different graphs could still point to the same real-world entity. Assume two annotation systems A1 and A2 that assign IRIs to the example text as shown in Figure 3.2. A1 relies on DBpedia IRIs while A2 uses Wikipedia IRIs and the prefix `ex:` for emerging entities. The result of A1 contains only one correct IRI. The second IRI refers to the island while the gold standard referred to the state.[10] The third annotation of A1 assigns the IRI of the English sportsman Arthur Milton who has not been born in Hawaii and does not match the emerging entity defined in the gold standard. It can be summarized that for the annotations of A1, a simple string-based matching would identify matches and mismatches between the system's answer and the gold standard. However, this is not the case for the result returned by A2. The first and second IRI refer to the correct real-world entities but use Wikipedia IRIs. The third IRI shows that A2 identified the third named entity

[9] This namespace for emerging entities is also used by Speck et al. [259].

[10] The island Hawaii is the largest island of the state Hawaii. However, the state covers more than 130 other islands.

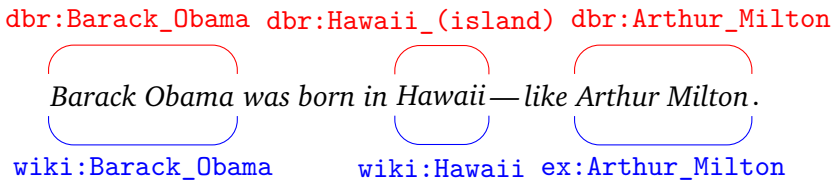


Figure 3.2.: The example document annotated by the two example systems A1 (red) and A2 (blue).

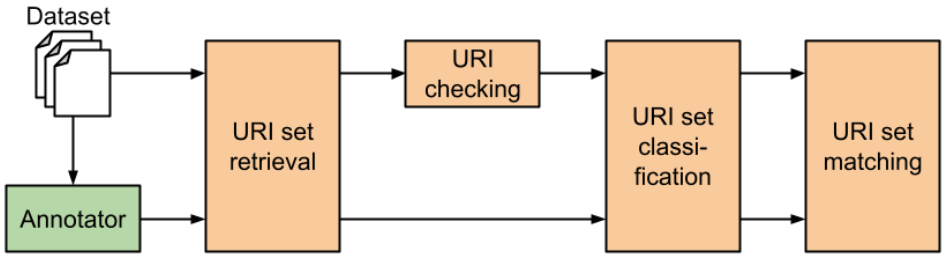


Figure 3.3.: Schema of the four components of the entity matching process.

correctly as an emerging entity. However, a string-based comparison would not find a single match. Hence, a more sophisticated matching is needed.

3.2.2 Extended IRI Matching

Our extension to GERBIL mainly covers the matching of IRIs. It tackles both challenges described above—different reference knowledge graphs and outdated IRIs—by representing the meaning of a named entity as a set of IRIs and an enhanced entity matching shown in Figure 3.3. It comprises the following four steps:

1. IRI set retrieval,
2. IRI checking,
3. IRI set classification, and
4. IRI set matching.

We explain these steps in more detail in the following.

IRI Set Retrieval

Since an entity can be described in several knowledge graphs using different IRIs, GERBIL assigns a set of IRIs to a single annotation representing the semantic meaning of this annotation. Initially, this set contains the single IRI that has been loaded from the dataset or read from an annotators response. The set is expanded by crawling

the Semantic Web graph using `owl:sameAs` links as well as redirects.[11] These links are retrieved using different modules that are chosen based on the domain of the IRI. The general approach we implemented dereferences the given IRI and tries to parse the returned triples. Although this approach works with every knowledge graph with dereferencable IRIs, we offer some additional modules. A DBpedia-Wikipedia bridge module transforms DBpedia IRIs into Wikipedia IRIs and vice versa. Additionally, we implemented a Wikipedia API client module that can retrieve redirects for Wikipedia IRIs. Moreover, one module can handle common errors like wrong domain names, e.g., the usage of `DBpedia.org` instead of `dbpedia.org`, and the transformation of an IRI into a URI and vice versa. The expansion of the set stops if all IRIs in the set have been used by these modules and no new IRI could be added.

IRI Checking

As explained above, IRIs in the ground truth of a benchmark dataset can be outdated. GERBIL tries to minimize the influence of outdated IRIs by checking every IRI in a given dataset for its existence in the reference knowledge graph. If an IRI cannot be found in the knowledge graph, it is marked as outdated. GERBIL offers two ways to search for an IRI. Either it is searched in a provided index that lists all available IRIs of a reference knowledge graph, or it is dereferenced. However, this is only possible for IRIs of knowledge graphs that abide by the Linked Data principles and provide dereferencable IRIs. All outdated IRIs are changed by replacing their namespace with a namespace that marks them as emerging entities.[12] However, it should be noted that an annotation that has an outdated IRI in the gold standard, may not end up with a set of IRIs that makes it an emerging entity. The previous step may have already derived a new IRI from the reference knowledge graph which would be part of the IRI set.[13] The editing of the outdated IRI would not affect the previously derived IRIs in the set.

[11] We use the prefix `owl` for theIRI `http://www.w3.org/2002/07/owl#`.

[12] This follows a similar strategy that Cornolti et al. [68] applied. They manually deleted outdated Wikipedia links from their datasets. In comparison, our strategy is better since 1) it keeps the information that an entity is mentioned in the text and 2) it is applied automatically without any manual effort.

[13] The Wikipedia API is a good example for a source of links from outdated IRIs to new IRIs since requests to old Wikipedia article titles are forwarded to the new articles as long as these old titles have not been reused.

IRI Set Classification

As explained before, the entities can be separated into two classes [125]. Entities can be either in the reference knowledge graph or they can be emerging entities. The third step of GERBIL's entity matching process assigns one of these two classes to each annotation based on the annotation's IRI set. If at least one IRI in the annotation's set belongs to the reference knowledge graph, it is marked as a known entity from that graph. If there is no such IRI, the entity is an emerging entity and the annotation is marked accordingly.

IRI Set Matching

The final step of checking whether two entity annotations match each other is to check whether their two IRI sets match. The match is based on two conditions. First, both sets have to have the same class assigned, i.e., either both have to be classified as belonging to the knowledge graph or both have to be classified as emerging entities. Second, if they have been classified as entities of the graph, the two sets have to overlap, i.e., there has to be at least one IRI that both sets have in common. It should be noted that this second condition is not applied if the sets are classified as emerging entities. The IRIs for these entities are typically generated and different systems may use different strategies for this generation.

With this IRI matching, our extended version of GERBIL addresses the two aforementioned challenges. However, as described in Section 3.1, GERBIL's architecture comes with several other drawbacks. It is based on the idea to benchmark knowledge extraction systems that have been deployed as Web services. While this is a lightweight approach that allows the easy comparison of prototypes or smaller systems, it does not support a fair benchmarking of large, distributed systems. In the following section, we will present requirements that should be fulfilled by such a benchmarking platform.

3.3 Requirements

We adopted a user-driven approach to develop our platform. Additionally to the goals of the HOBBIT project, the requirements were mainly derived from an online survey as well as a workshop—both described by Fundulaki [102].[14]

The survey had 61 expert participants representing their organizations. These experts were contacted via mail using several mailing lists of the Semantic Web community. During the survey, the participants were asked to add themselves to one ore more of three stake holder groups. 48 participants classified themselves as solution providers, i.e., they represent an organization which implements a Linked Data system. 46 participants added themselves to the group of technology users, i.e., people which are using Linked Data systems developed by a 3rd party. The third group—the scientific community which aims at identifying problems in existing solutions and developing new algorithms—comprised 47 participants. Asked for the target of the Linked Data systems they are developing or using, 50 participants stated to work in the area of storage and querying, 39 in the area of interlinking, 39 in classification and enrichment, 35 in discovery, 31 in extraction and 22 in reasoning. The survey further asked which benchmarks the participants use. This was further detailed with the size and type of datasets (synthetic, real-world or a combination of both) they use as well as the Key Performance Indicators (KPIs) they are interested in [102].

In 2016, a workshop was arranged within the programme of the Extended Semantic Web Conference [102]. 21 conference participants took part in the workshop and discussed the goals of the HOBBIT project as well as requirements. The participants were separated into 4 groups—Generation & Acquisition, Analysis & Processing, Storage & Curation as well as Visualisation & Services—covering the complete Linked Data life cycle. Each group discussed requirements which the benchmarks of this area as well as the benchmarking platform used to execute these benchmarks should fulfil. To distinguish the gathered user requirements from the FAIR data principles, we will abbreviate these user requirements with **U**.

3.3.1 Functional Requirements

U1 The main functionality of the platform is the execution of benchmarks.

U2 Benchmark results should be presented in human- and machine-readable form.

[14] Please note that [102] is also available via the Community Research and Development Information Service of the European Commission using the grand agreement ID of the HOBBIT project: 688227. See `https://cordis.europa.eu/project/rcn/199489/results/en`; last accessed on 03.08.2022.

U3 It should be possible to add new benchmarks and new systems.

U4 The platform should offer repeatable experiments and analysis of results.

U5 The KPIs should include the effectiveness, e.g., the accuracy, and the efficiency, e.g., runtime of systems.

U6 The platform should be able to measure the scalability of solutions. This leads to the need of a scalable generation of both—data the evaluation is based on as well as tasks a system has to execute.

U7 The platform should support the benchmarking of distributed systems.

U8 The platform should support the execution of benchmarking challenges. This includes 1) the creation of challenges within the platform, 2) the registration of users with their system for the challenge, 3) the execution of the challenge experiments at a predefined point in time, and 4) the summary of the experiment results for this challenge.

These functional requirements predefined the corner stones for the platforms architecture. In Section 3.4, it will be shown how the platform fulfills each of them.

3.3.2 Qualitative Requirements

U9 The benchmarks should be easy to use and interfaces provided should be as simple as possible.

U10 The platform should support different programming languages.

U11 The results should be archived safely for later reference.

U12 The platform needs to be robust regarding faulty benchmarks or systems.

Several requirements—especially **U1**–**U4** as well as **U8**—addressed fundamental functions of a benchmarking platform that supports the execution of benchmarking challenges and were directly derived from this goal. However, the results of the survey as well as the workshop show that the participants agreed to the goals of the project and that especially the repeatability of experiments (**U4**) is of importance to the community. The range of mentioned KPIs in the survey as well as in the results of the workshop let to **U5**. The need to measure the efficiency of systems is one reason for **U6**. The large range of dataset sizes used by the survey participants was another reason.[15] **U7** was an important requirement to ensure the ability to benchmark systems which achieve their scalability by horizontal scaling. **U9** was a result of the workshop. **U10** and **U11** were derived very early. Although not explicitly mentioned during the workshop, the usage of different programming languages within the community became evident. Additionally, **U11** was derived

[15] 3.6% of the survey participants used datasets with less than 10 thousand triples while 35.7% used datasets with more than 100 million triples [102].

from the usage of existing platforms which have already been accepted by the community, e.g., the citable IRIs of GERBIL [231]. The error tolerance of a software is a general requirement for most developments. However, with **U12** it gained additional attention because the platform allows the upload of third party software which might not be reliable.

We derived the degree of modularity and the error handling of the platform from these requirements (**U1**, **U3**–**U12**). The result analysis component and interfaces were designed to accommodate **U2** and **U9**–**U12**. Details are provided in Section 3.4.

3.3.3 FAIR Data Principles

From the beginning on, the platform was built to support the FAIR data principles [306].[16] The following list is a literal citation of Wilkinson et al. [306] and their summary of the principles:

- **F1** (Meta)data are assigned a globally unique and persistent identifier.
- **F2** Data are described with rich metadata (defined by R1 below).
- **F3** Metadata clearly and explicitly include the identifier of the data they describe.
- **F4** (Meta)data are registered or indexed in a searchable resource.
- **A1** (Meta)data are retrievable by their identifier using a standardized communications protocol.
  - **A1.1** The protocol is open, free, and universally implementable.
  - **A1.2** The protocol allows for an authentication and authorisation procedure, where necessary.
- **A2** Metadata are accessible, even when the data are no longer available.
- **I1** (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- **I2** (Meta)data use vocabularies that follow FAIR data principles.
- **I3** (Meta)data include qualified references to other (meta)data.
- **R1** Meta(data) are richly described with a plurality of accurate and relevant attributes.
  - **R1.1** (Meta)data are released with a clear and accessible data usage license.
  - **R1.2** (Meta)data are associated with detailed provenance.
  - **R1.3** (Meta)data meet domain-relevant community standards.

[16] `https://www.go-fair.org/fair-principles/`; last accessed on 03.08.2022.

The following section shows the design of the HOBBIT platform. Within this section, we explain how the platform fulfills the user requirements and how it supports the FAIR data principles.

3.4 Platform Architecture

3.4.1 Overview

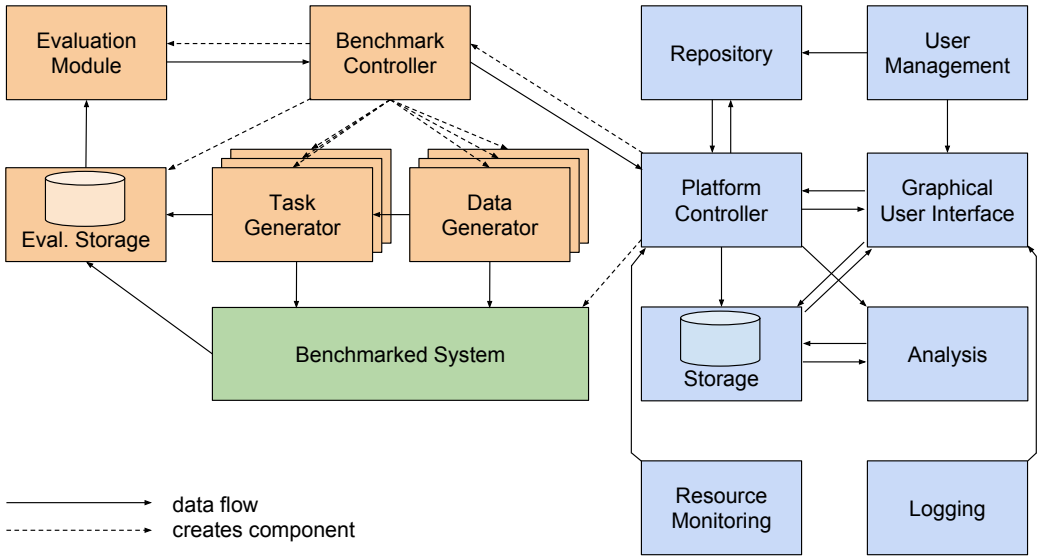


Figure 3.4.: Architecture of the HOBBIT platform

Figure 3.4 gives an overview of the architecture of the HOBBIT platform. The platform is based on a container architecture, i.e., the components are implemented as independent containers. This eases the adding of new benchmarks and systems (**U3**), which can be implemented using different languages (**U10**). Additionally, it eases the development and maintenance of the platform itself and adds a separation between the platform, benchmark, and system containers, thus limiting the influence of faulty program code to its container instead of decreasing the stability of the whole platform (**U11**, **U12**). Using containers for benchmark and system components also gives the possibility of scaling both by offering the deployment of additional containers across multiple machines (**U6**, **U7**). The communication between these components is ensured by means of a message bus. Choosing this established communication method eases the implementation of benchmarks and systems based on different programming languages (**U9**, **U10**).

3.4.2 Platform Components

The platform has several components (see blue elements in Figure 3.4). They offer the main functionality of the platform.

Platform Controller

The platform controller is the central component of the HOBBIT platform. Its main role is to coordinate the interaction of other components as needed. This mainly includes handling requests that come from the user interface component, starting and stopping of experiments, observing the health of the cluster, and triggering the analysis component. In addition, the controller manages a priority queue that contains user-configured experiments that are to be executed in the future. The execution order of experiment configurations is determined using 1) the time at which they have been configured by the user (following the first-in-first-out principle) and 2) the priority of experiments, which is derived from whether the said experiment is part of a scheduled challenge (higher priority) or not (**U8**). The internal status of the platform controller is stored in a database. This enables restarting the controller without losing its current status, e.g., the content of the experiment queue.

The platform controller uses features of Docker Swarm to observe the status of the cluster that is used to execute the experiments. E.g., if one of the nodes drops out of the cluster, the comparability between single experiments might not be given (**U4**). Thus, the platform controller needs to be aware of the number of working nodes that are available for the experiment. If there is no running experiment and the queue is not empty, the platform controller initiates the execution of an experiment and observes its state. If the experiment takes more time than a configured maximum, the platform controller terminates the benchmark components and the system that belongs to the experiment. By these means, it also ensures that faulty benchmarks or systems cannot block the platform (**U12**).

Storage

The storage component contains the experiment results and configured challenges. It comprises two containers—a triple store that uses the HOBBIT ontology to describe results and a handler for the communication between the message bus and the triple store. The storage component offers a public SPARQL endpoint with read-only

access which can be queried via HTTP/HTTPS (**U2**, **F4**, **A1**).[17] The write access is limited to the platform controller, the user interface, and the analysis component. The controller stores experiment results and manages running challenges. The user interface presents the available data to the user and enables the configuration of new challenges as well as the registration of systems for taking part in a challenge (**U8**). The analysis component requests experiment results from the storage and stores results of the analysis.

Ontology

The experiment results, the metadata of experiments and challenges as well as the results of the analysis component are stored as RDF triples [250] (**I1**). Where possible, we used established RDF vocabularies (**I2**, **R1.3**).[18] However, for describing the experiments and challenges in detail we created the HOBBIT ontology.[19] In the following, we use `ho` as prefix to shorten the ontology's namespace `http://w3id.org/hobbit/vocab#`.

The ontology offers classes and properties to define the metadata for the single benchmarks and benchmarked systems. The main schema of the ontology is depicted in Figure 3.5. For each benchmark or system a user would like to use within the platform, a metadata file has to be provided containing some general information. This includes the definition of an IRI for each benchmark and system (**F1**), a name, a description, and an IRI of the API offered by the benchmark and implemented by the system. Based on the API IRI the platform can map the available systems to the benchmarks to ensure that the system is applicable for a given benchmark. Additionally, a benchmark's metadata include parameters and KPIs. The parameters can be defined to be configurable through the user interface when starting an experiment and whether the parameters should be used as feature in the analysis component.

A system's metadata offers the definition of several system instances with different parameterizations. The analysis method can make use of the different parameter values of the instances to measure the impact of the parameters on the KPIs.

Experiments are described with triples regarding 1) provenance, 2) the experiment results, 3) the benchmark configuration, and 4) benchmark as well as 5) system

[17] Our endpoint can be found at `https://db.project-hobbit.eu/sparql`; last accessed on 03.08.2022.

[18] Namely, RDF [52], PROV-O [161], Data Cube [73], and XSD [60].

[19] The formal specification of the ontology can be found at `https://github.com/hobbit-project/ontology`; last accessed on 03.08.2022.

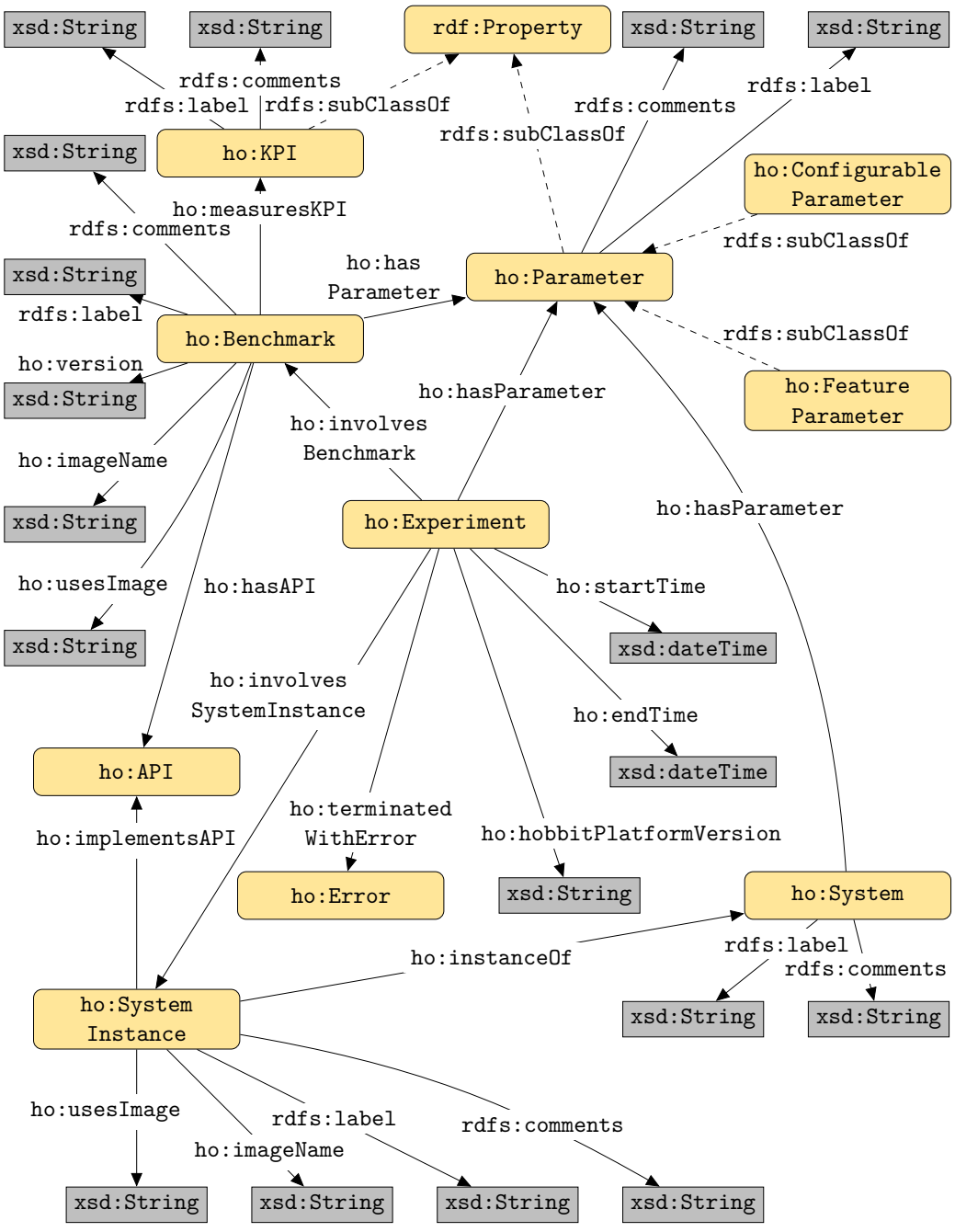


Figure 3.5.: Main concepts of the HOBBIT ontology.

metadata (**F2**, **F3**, **I3**, **R1**, **R1.2**). The provenance information covers—additionally to the metadata of the benchmark and system—the start and end time of the experiment as well as details about the hardware on which the experiment has been executed. The experiment results are generated by the implementation of the benchmark and typically contain results for the single KPIs which are defined in the benchmark's metadata. Together with the metadata of the benchmark and system, the values of the KPIs and their description are used by the analysis component (**U4**). The platform controller assigns an IRI to the experiment (**F1**) and copies the configuration of the benchmark as well as the metadata of the benchmark and system into the experiment's metadata. Note that this makes sure that even if a user removes a benchmark or system from the platform after executing an experiment their metadata is still available (**A2**).

Challenges which are carried out on the platform are modeled by separating them into single tasks. Each task has a benchmark with a certain parameterization and users can register their systems for the single tasks to take part in the challenge. A challenge and its tasks have a generated IRI (**F1**) and come with a label as well as a description. Additionally, the creator of the challenge can define the execution date and the publication date of the challenge as well as a link to a Web page giving further information about the challenge. The first date defines the point in time at which the execution of the single experiments of the challenge should start while the latter defines the day at which the results should be made public. The experiments that are part of a challenge, point to the challenge task for which they have been executed (**I3**). An overview of the concepts used to describe a challenge is given by Figure 3.6.

Essentially, the ontology offers classes and properties to store the configuration and the results of an experiment. IRIs are assigned to benchmarks, benchmarked software systems, and KPIs. Moreover, benchmark configurations as well as benchmark and system features, e.g., a certain parameterization, can be described. In addition to experiments, the ontology allows for the description of challenges, tasks in challenges and benchmarks associated with these tasks.

Analysis

This component is triggered after an experiment has been carried out successfully. Its task is to enhance the benchmark results by combining them with the features of the benchmarked system(s) and the data or task generators. This combination can lead to additional insights, e.g., strengths and weaknesses of a certain system (**U4**).

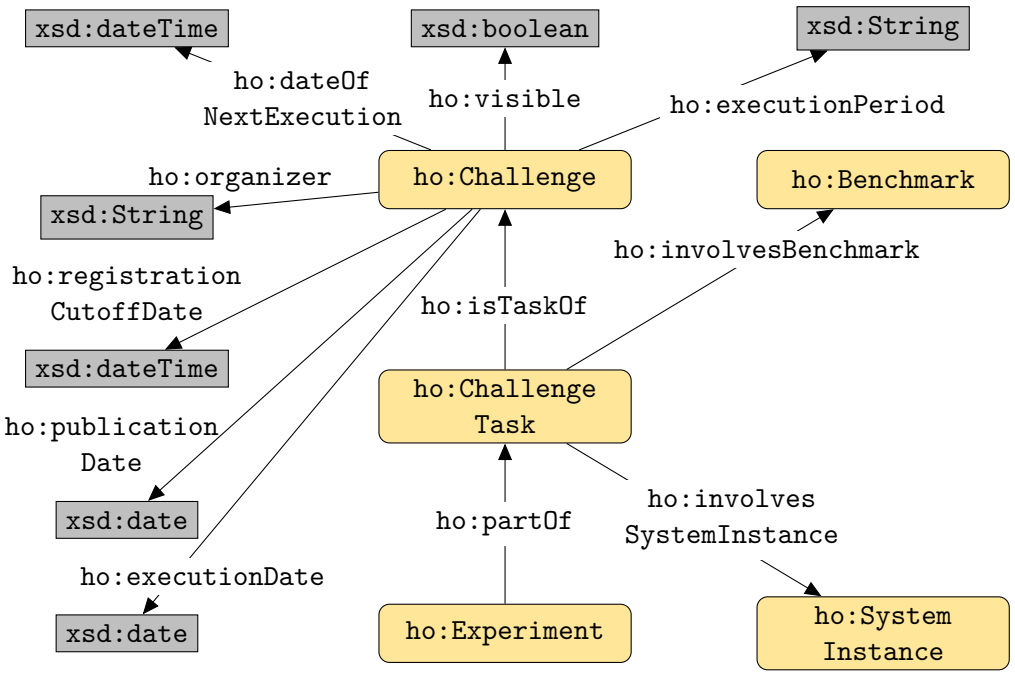


Figure 3.6.: Concepts of the HOBBIT ontology to describe a challenge. The `rdfs:label` and `rdfs:comment` triples for the `ho:Challenge` and `ho:ChallengeTask` have been left out.

While the component uses the results of benchmarks, it is modeled independently from any benchmark implementation.

Graphical User Interface

The graphical user interface component handles the interaction with the user via HTTP or HTTPS (**A1**). It retrieves information from the user management that allows different roles enabling the user interface to offer functionality for authenticated users as well as a guest role for unauthenticated users. For example, a guest is only allowed to read the results of experiments and analysis (**U2**). Since the number of experiments is steadily increasing, the user interface offers a filter and sorting mechanism to increase the findability (**F4**). Experiments are currently visualized as table containing their metadata, the parameter values and the KPI values. This table view can also be used to compare several experiments with each other. Additionally, plots as shown in Figure 3.7 are generated where applicable.[20]

Authenticated users have additional rights ranging from starting experiments to organizing challenges, i.e., define experiments with a certain date at which they will

[20]The example is part of the experiment `https://w3id.org/hobbit/experiments#1540829047982`; last accessed on 03.08.2022.

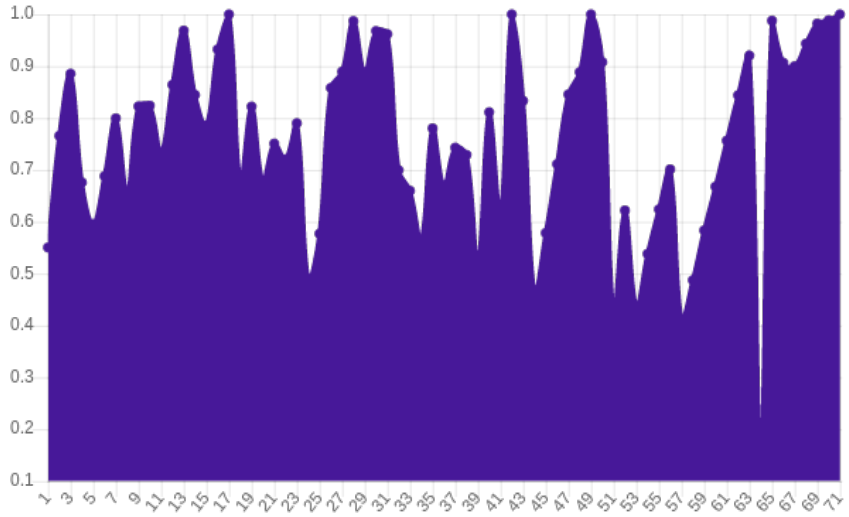


Figure 3.7.: A screenshot of a plot generated for a KPI. It shows the F1-measure the Jena Fuseki triple store achieved for 71 consecutive select queries during a run of the Odin benchmark [104].

be executed (**U1**, **U8**). Additionally, experiments and challenges have dereferencable IRIs assigned, i.e., a user can copy the IRI of an experiment or a challenge into the browser's URL bar and the server shows the details of this resource (**F1**, **A1**). For our online instance, we offer w3id IRIs to enable static URLs that can be redirected.[21]

For each benchmark, a report can be generated. This comprises 1) a brief overview over the results of the last experiments carried out with the benchmark, 2) scatter plots that compare values of features and KPIs, and 3) plots showing the correlation between benchmark features and the performance achieved by the single systems. Such a plot is shown in Figure 3.8.

If the license of the data has been configured in the triple store, the information is shown in the user interface (**R1.1**). The data of our online instance is licensed under the Creative Commons Attribution 4.0 International Public License.[22]

Message Bus

This component contains the message bus system. Three different communication patterns are used. First, labeled data queues simply forward data, e.g., the data generated by the mimicking algorithm is transferred from several data generators to

[21] See `https://w3id.org/`; last accessed on 03.08.2022.

[22] License: `https://creativecommons.org/licenses/by/4.0/legalcode`; last accessed on 03.08.2022.. The license statement of our online instance can be found at `https://master.project-hobbit.eu/home`; last accessed on 03.08.2022.

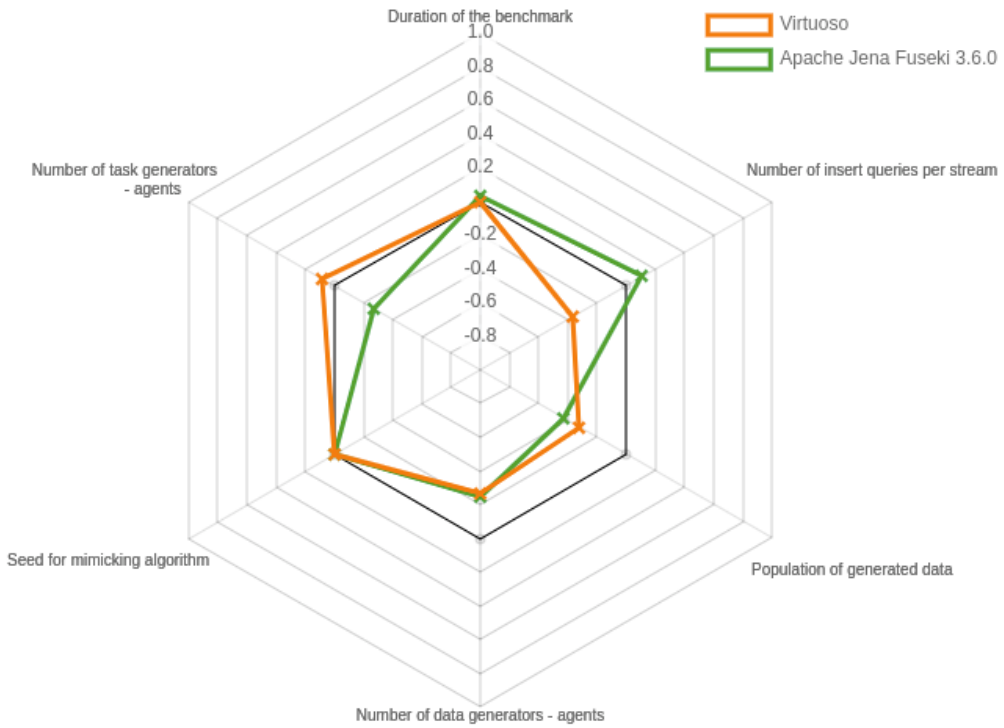


Figure 3.8.: An example of a diagram showing the Pearson correlations between the different parameters of the Odin benchmark [104] and the micro F1-measure achieved by the two triple stores Virtuoso and Jena Fuseki.

several task generators. The second pattern works like remote procedure calls. The queue has one single receiving consumer that executes a command, e.g., a SPARQL query, and sends a response containing the result. Third, a central broadcasting queue is used (`hobbit.command`). Every component connected to this queue receives all messages sent by one of the other connected components. This queue is used to connect the loosely coupled components and orchestrate their activities.

User Management

The user management is based on Keycloak.[23] It allows the upload of private systems which cannot be seen by other users. Additionally, the platform makes use of different user roles to enable single users to create challenges. Note that the user management offers a guest role that enables unregistered users to see the publicly available experiment results.

[23] `https://www.keycloak.org/`; last accessed on 03.08.2022.

Repository

The repository contains all available benchmarks and systems. For our online instance, the repository is a Gitlab instance which can be used by registered users to upload Docker images and define the metadata of their benchmarks and systems (**U3**, **R1.3**).[24] Note that users can define the visibility of their systems, i.e., the platform supports publicly accessible systems and benchmarks that can be used by every registered user as well as private systems. However, the experiment results (including the system's metadata) will always be made public.

Resource Monitoring

The resource monitoring component uses Prometheus to collect information about the hardware resources used by the benchmarked system.[25] The benchmark can request this information to include it into its evaluation. At the moment, the CPU time, the disk space, and the amount of RAM used by the system can be monitored. Based on the architecture of Prometheus, this list of metrics can be further extended.

Logging

The logging comprises three components—Logstash, Elasticsearch, and Kibana.[26] While Logstash collects the log messages from the single components, Elasticsearch is used to store them inside a full text index. Kibana offers the user interface for accessing this index. The logs are kept private. However, owners of systems or benchmarks can download the logs of their components for a particular experiment from the user interface.

3.4.3 Benchmark Components

These components are part of given benchmarks and have been colored orange in Figure 3.4. Hence, they are instantiated for a particular experiment and are destroyed when the experiment ends. A benchmark execution has three phases—an initialization phase, a benchmarking phase, and an evaluation phase. The

[24] `https://about.gitlab.com/`; last accessed on 03.08.2022.
[25] `https://prometheus.io/`; last accessed on 03.08.2022.
[26] `https://www.elastic.co/de/products/logstash`, `https://www.elastic.co/de/products/elasticsearch`, and `https://www.elastic.co/de/products/kibana`; last accessed on 03.08.2022..

phases are described in more detail in Section 3.4.5. It should be noted that the components described in this section represent our suggestion for the structure of a benchmark. However, the HOBBIT platform supports a wide range of possible benchmark structures as long as a benchmark implements the necessary API to communicate with the platform controller.

Benchmark Controller

The benchmark controller is the central component of a benchmark. It communicates with the platform controller and it creates and controls the data generators, task generators, evaluation-storage, and evaluation-module.

Data Generator

Data generators are responsible for supplying the other components with the data necessary for the experiment. Depending on the benchmark implementation, there are two types of generators. Either, a given dataset, e.g., a real-world dataset, is loaded from a file or the component encapsulates an algorithm able to generate the necessary data. Importantly, data generators can be run in a distributed fashion to ensure that the platform can create the necessary data volumes or data velocity. Typically, data generators are created by the benchmark controller and configured using benchmark-specific parameters. They generate data based on the given parameters, send said data to the task generators and the system adapter, and terminate when the required data has been submitted.

Task Generator

Task generators get data from data generators, generate tasks that can be identified with an ID and send these tasks to the system adapter. Each task represents a single problem that has to be solved by the benchmarked system (e.g., a SPARQL query that should be answered by a triple store). The expected response for the generated task is sent to the evaluation storage. Like data generators, task generators can be scaled to run in a distributed fashion.

Evaluation Storage

This component stores the gold standard results as well as the responses of the benchmarked system during the benchmarking phase. During the evaluation phase it sends this data to the evaluation module. Internally, the output of a benchmark is stored as a set of key-value pairs. The task IDs are used as keys. Each value comprises 1) the expected result, 2) the result calculated by the benchmarked system, 3) the timestamp at which the task was sent to the system by a task generator, and 4) the timestamp at which the response was received by the evaluation storage.

Evaluation Module

The evaluation module is created by the benchmark controller at the beginning of the evaluation phase and requests results from the evaluation storage. It evaluates them by computing the KPIs associated with the benchmark. It should be noted that the decision which KPIs will be used is mainly up to the benchmark developer. Both, the effectiveness as well as the efficiency of systems can be measured (**U5**). After computing the KPIs, the component summarizes the evaluation results and sends them to the benchmark controller before it terminates.

3.4.4 Benchmarked System Components

Each system to be benchmarked using the HOBBIT platform has to implement the API of the benchmark to be used and the API of the platform. Since systems are typically not developed for the mere sake of being benchmarked with our platform, each system is usually connected to the platform by means of a system adapter container. The system adapter serves as a proxy translating messages from the HOBBIT platform to the system to be benchmarked and vice versa. The system adapter of each of the systems to benchmark is instantiated by the platform controller when an experiment is started. Adapters can create additional containers that might contain components of the benchmarked system. Thereafter, they send a ready signal to the platform controller to indicate that they are ready to be benchmarked. They receive incoming data and tasks, forward them to the system and send its responses to the evaluation storage. Adapters stop the benchmarked system and terminate after they receive a command indicating that all tasks have been completed.

3.4.5 Benchmark Workflow

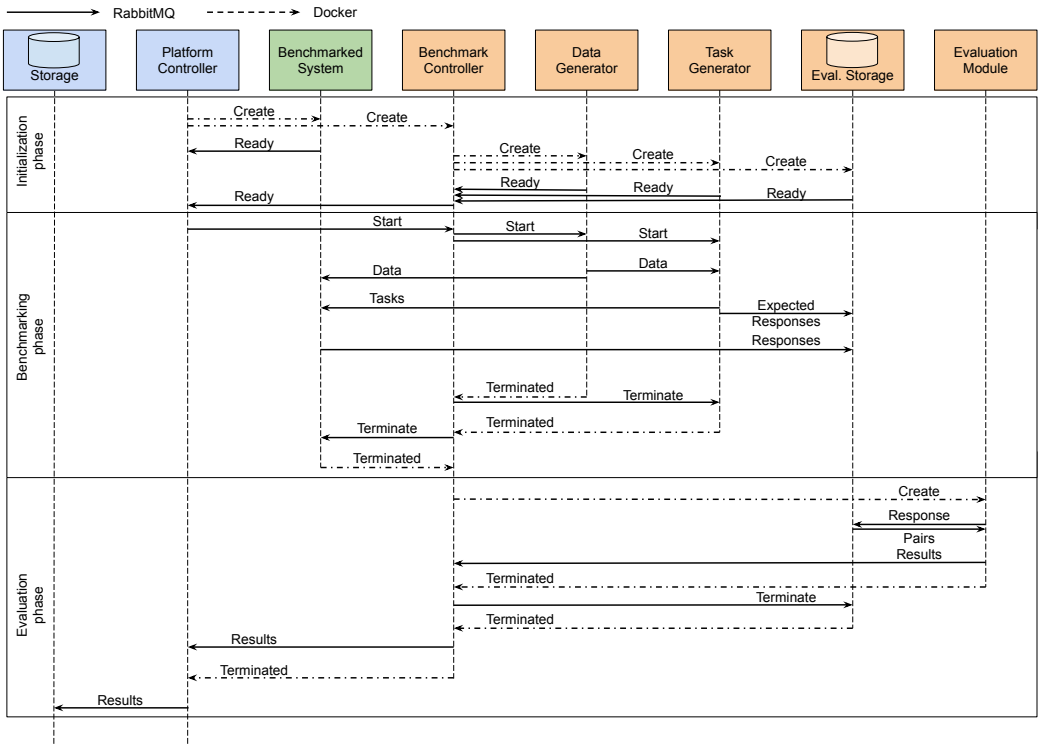


Figure 3.9.: Simplified overview of the general benchmarking workflow. Parts of the platform (e.g., the user interface) are left out and the benchmark controller creates other containers directly, without sending requests to the platform controller. Solid arrows indicate a communication via the message bus while dashed arrows represent an interaction with Docker swarm.

Since the platform was designed for executing benchmarks (**U1**), we defined a typical workflow of benchmarking a Big Linked Data system. The workflow is abstracted to make sure that it can be used for benchmarking all steps of the Linked Data life cycle. Figure 3.9 shows a sequence diagram containing the steps as well as the type of communication that is used. Note that the orchestration of the single benchmark components is part of the benchmark and can be different across different benchmark implementations.

Initialization Phase

At the beginning of the benchmarking process, the platform controller makes sure that a benchmark can be started. This includes a check to make sure that all hardware nodes of the cluster are available. The platform controller then instantiates the system adapter. The said adapter first initializes, then starts the system to be

benchmarked and makes sure that it is working properly. Finally, the adapter sends a message to the platform controller to indicate that it is ready. Once the system adapter has been started, the platform controller generates the benchmark controller. The task of the benchmark controller is to ensure that the data and tasks for a given benchmark are generated and dispatched according to a given specification. To achieve this goal, the controller instantiates the data and task generators as well as the evaluation storage. It then sends a message to the platform controller to indicate that it is ready.

Benchmarking Phase

The platform controller waits until both the system adapter and the benchmark controller are ready before starting the benchmarking phase by sending a start signal to the benchmark controller which starts the data generators. The data generators start the data generation algorithms to create the data that will underlie the benchmark. The data is sent to the system adapter and to the task generators. The task generators generate the tasks and send them to the system adapter, which triggers the required processing of the data in the system. The system response is forwarded to the evaluation storage by the system adapter. The task generators store the corresponding expected result in the evaluation storage. After the data and task generators finish their work, the benchmarking phase ends and both the generators and the system adapter terminate.

Evaluation Phase

During the evaluation phase, the benchmark controller creates the evaluation module. The evaluation module loads the results from the evaluation storage. This is carried out by requesting the result pairs, i.e., the expected result and the result received from the system for a single task, from the storage. The evaluation module uses these pairs to evaluate the system's performance and to calculate the KPIs. The results of this evaluation are returned to the benchmark controller before the evaluation module and storage terminate. The benchmark controller adds information for repeating the experiment, e.g., its parameters, to the evaluation results, sends them to the platform controller and terminates. Note that this makes sure that all the data is still available, although the benchmark or the benchmarked system are deleted from the servers (**A2**). After the benchmark controller has finished its work, the platform controller can add additional information to the result, e.g., the configuration of the hardware, and store the result. Following this, a new evaluation

can be started. The platform controller sends the IRI of the new experiment result to the analysis component. The analysis component reads the evaluation results from the storage, processes them and stores additional information in the storage.

Importantly, the platform allows for other orchestration schemes. For example, it is possible to generate all the data in a first step before the task generators start to generate their tasks based on the complete data. In another variation, the task generators can also be enabled to generate a task, wait for the response of the system and then send the subsequent task.

3.5 Mimicking real-world RDF Graphs

As described in the previous sections, benchmarking an Linked Data system typically relies on data, and the creation and design of datasets is a crucial part of the design of a benchmark [85]. For example, given that the performance of triple stores changes across dataset versions [67, 209], there is a need to predict the future performance of storage solutions given existing versions of a dataset. Such a prediction can facilitate the deployment of reliable knowledge graph infrastructures, the timely acquisition and alteration of software components, and the maintenance of quality-of-service requirements.

In this Section, we hence address the challenge of *predicting the topology of future versions of knowledge graphs given current versions*. We use SPARQL queries as a proxy to evaluate the quality of our prediction. Since benchmarking has been traditionally associated with storage [109], we developed the mimicking algorithm with a focus on querying performance. However, this is not an intrinsic limitation of the algorithm, which aims to mimic graph topology. This task differs from that addressed by current RDF generators, which assume a particular dataset or ontology (e.g., universities) and generate data based thereupon [14, 16, 89, 114, 248, 272].

We formalize the problem as follows: Given versions $\mathscr{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_{|\mathscr{G}|}\}$ of a knowledge graph (e.g., WikiData, DBpedia, MusicBrainz), we aim to learn a synthetic dataset generator which allows the prediction of the performance and ranking of storage solutions on a version of $\mathscr{G}$ of size ν_R, where ν_R is the number of IRI resources the generated version should have. We use training data in form of $\mathscr{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_{|\mathscr{G}|}\}$ to learn *graph-specific invariants*, which we define as functions Λ whose results have a low variance on $\mathscr{G}$ and a high variance on other sets of graphs disjoint from $\mathscr{G}$. We learn these functions using a refinement operator ρ for arithmetic functions. We show that our operator is finite, redundant, and complete.

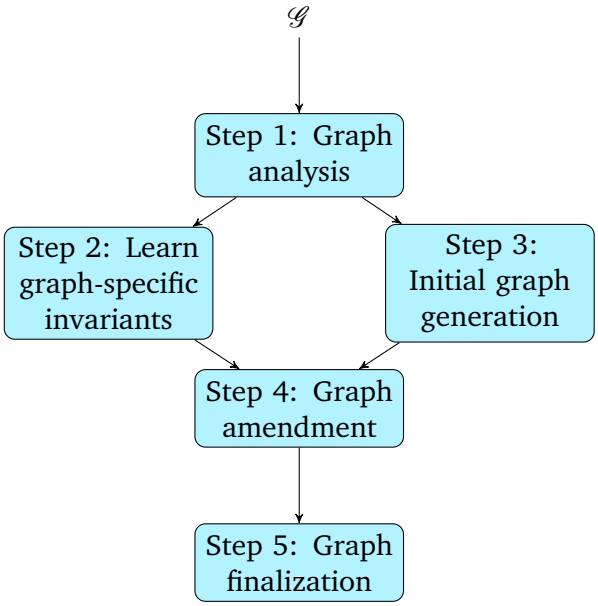


Figure 3.10.: Overview of the 5 steps of LEMMING.

Our experiments show that our approach is able to generate datasets with which the ranking on real datasets can be approximated with a root mean squared error on ranks under 0.15.

Figure 3.10 shows an overview of our approach LEMMING. It takes the set of versions $\mathscr{G}$ as input. It should be noted that we assume that all graphs in the given set are fully materialized, i.e., all implicit triples that can be inferred based on the ontology of the graph have been made explicit. Our approach comprises the five steps depicted in Figure 3.10. First, the given graphs are analyzed to gather necessary statistics. After that, the graph-specific invariants Λ for $\mathscr{G}$ are learned using a refinement operator ρ. In parallel, an initial graph of size ν_R is generated, which is further refined to meet the values of the expressions Λ in Step 4. Finally, the graph is finalized by adding literals and exporting it to RDF. These five steps are further detailed in the following.

3.5.1 Graph Analysis

First, the given set of graphs $\mathscr{G}$ is analyzed. We assume that all graphs in $\mathscr{G}$ are fully materialized. We separate the triples of each given graph into two sets. Similar to datatype and object properties defined in Definitions 2.12 and 2.13, respectively, we distinguish between datatype and object triples. Datatype triples have a literal as

object while object triples have a blank node or IRI as object. For a given knowledge graph $\mathcal{G}$, these two subsets of $T_{\mathcal{G}}$ are defined as follows:

$$T_{L\mathcal{G}} = \{(s,p,o)\,|(s,p,o) \in T_{\mathcal{G}} \wedge o \in \mathfrak{L}\}\,, \tag{3.1}$$

$$T_{R\mathcal{G}} = \{(s,p,o)\,|(s,p,o) \in T_{\mathcal{G}} \wedge o \in \mathfrak{I} \cup \mathfrak{B}\}\,. \tag{3.2}$$

We also define a subset of nodes of a given knowledge graph that contains the blank nodes and IRI resources but not the literals as follows:

$$V_{R\mathcal{G}} = V_{\mathcal{G}} \setminus L_{\mathcal{G}} = R_{\mathcal{G}} \cup B_{\mathcal{G}}\,. \tag{3.3}$$

We start the analysis with the calculation of several metrics for the object triple parts of the given graphs. The density of each knowledge graph $\mathcal{G}_i \in \mathscr{G}$, denoted, $\delta_{\mathcal{G}_i}$ is determined based on the object triples as follows:

$$\delta_{\mathcal{G}_i} = \frac{|T_{R,\mathcal{G}_i}|}{|V_{R,\mathcal{G}_i}|}\,. \tag{3.4}$$

Let $C_{\mathscr{G}}$ be the set of all classes that exist in $\mathscr{G}$ and $C \in 2^{|C_{\mathscr{G}}|}$ a set of classes. We determine the distribution over sets of classes $C \in 2^{|C_{\mathscr{G}}|}$ by calculating the probability that a vertex is an instance of exactly all classes in C. This probability is defined as

$$\mathbb{P}_{\mathcal{G}_i}(C) = \frac{\left|\left\{v_j | v_j \in V_{R,\mathcal{G}_i} \wedge C = \mathfrak{c}_{\mathcal{G}_i}(v_j)\right\}\right|}{\left|V_{R,\mathcal{G}_i}\right|}\,, \tag{3.5}$$

where $\mathfrak{c}_{\mathcal{G}_i}(v_j)$ is the mapping function that derives the set of all classes for the node v_j according to knowledge graph $\mathcal{G}_i$ as defined in Section 2.1.2. In a similar way, the probability that an edge has p as property is calculated using

$$\mathbb{P}_{\mathcal{G}_i}(p) = \frac{\left|\left\{(s,p,o) | (s,p,o) \in T_{R,\mathcal{G}_i}\right\}\right|}{\left|T_{R,\mathcal{G}_i}\right|}\,. \tag{3.6}$$

For pairs of class sets (C_s, C_o), the probability that the subject and object vertices of an edge are instances of the classes in C_s and C_o, respectively, is determined using

$$\begin{aligned}&\mathbb{P}_{\mathcal{G}_i}((C_s,C_o)|\,p)\\ &= \frac{\left|\left\{(s,p,o) | (s,p,o) \in T_{R,\mathcal{G}_i} \wedge C_s \subseteq \mathfrak{c}_{\mathcal{G}_i}(s) \wedge C_o \subseteq \mathfrak{c}_{\mathcal{G}_i}(o)\right\}\right|}{\sum\limits_{C_j \subseteq C_{\mathscr{G}}} \sum\limits_{C_k \subseteq C_{\mathscr{G}}} \left|\left\{(s,p,o) | (s,p,o) \in T_{R,\mathcal{G}_i} \wedge C_j \subseteq \mathfrak{c}_{\mathcal{G}_i}(s) \wedge C_k \subseteq \mathfrak{c}_{\mathcal{G}_i}(o)\right\}\right|}.\end{aligned} \tag{3.7}$$

We also collect which types of triples occur in the graph, i.e., which combination of sets of classes and property occur. We will use these observed combinations as constraints for our graph generation process. Let (C_s, p, C_o) be a single constraint, i.e., a single combination of a set of classes C_s for the subject, a set of classes C_o for the object, and a property p for the predicate of a triple. The set of constraints collected on a given graph $\mathcal{G}_i$ is defined as follows:

$$\Omega_{\mathcal{G}_i} = \left\{(C_s, p, C_o) | (s, p, o) \in T_{R,\mathcal{G}_i} \wedge C_s = \mathfrak{c}_{\mathcal{G}_i}(s) \wedge C_o = \mathfrak{c}_{\mathcal{G}_i}(o)\right\} . \tag{3.8}$$

For each datatype property $p \in P_{L,\mathcal{G}_i}$, we collect the average number of outgoing edges with said property that the instances of a class set C_j in $\mathcal{G}_i$ have. We call this number $\mathrm{d}_{C_j,p,\mathcal{G}_i}$ define it as follows:

$$\mathrm{d}_{C_j,p,\mathcal{G}_i} = \frac{\left|\left\{(s, p, o) | (s, p, o) \in T_{L,\mathcal{G}_i} \wedge C_j \subseteq \mathfrak{c}_{\mathcal{G}_i}(s)\right\}\right|}{|\{s | C_j \subseteq \mathfrak{c}_{\mathcal{G}_i}(s)\}|}. \tag{3.9}$$

After analyzing the single graphs, the analysis results are summarized as follows:

$$\delta_{\mathscr{G}} = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{G}_i \in \mathscr{G}} \delta_{\mathcal{G}_i} , \tag{3.10}$$

$$\mathbb{P}_{\mathscr{G}}(C) = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{G}_i \in \mathscr{G}} \mathbb{P}_{\mathcal{G}_i}(C) , \tag{3.11}$$

$$\mathbb{P}_{\mathscr{G}}(p) = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{G}_i \in \mathscr{G}} \mathbb{P}_{\mathcal{G}_i}(p) , \tag{3.12}$$

$$\mathbb{P}_{\mathscr{G}}((C_s, C_o) | p) = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{G}_i \in \mathscr{G}} \mathbb{P}_{\mathcal{G}_i}((C_s, C_o) | p) , \tag{3.13}$$

$$\Omega_{\mathscr{G}} = \bigcup_{\mathcal{G}_i \in \mathscr{G}} \Omega_{\mathcal{G}_i} , \tag{3.14}$$

$$\mathrm{d}_{C_j,p,\mathscr{G}} = \frac{1}{|\mathscr{G}|} \sum_{\mathcal{G}_i \in \mathscr{G}} \mathrm{d}_{C_j,p,\mathcal{G}_i} . \tag{3.15}$$

In addition to that, we gather the degrees of the vertices that are instances of all classes over all graphs in $\mathscr{G}$. The degrees are used to determine the degree distribution $\varsigma_{C_j,\mathscr{G}}$ for each C_j. This allows the usage of different types of distributions for different sets C_j. For each property p in the datatype triples $T_{L,\mathscr{G}}$, we gather data about the literal values these triples of this property have as object. This data is used to create a literal value distribution $\ell_{p,\mathscr{G}}$ for each datatype property.

3.5.2 Learning Graph Invariants

Our approach to learning graph invariants is based on a refinement operator ρ, which uses a specificity function as heuristic to measure the quality of an arithmetic expression. In the following, we begin by presenting ρ and prove that it is finite, redundant, and complete. We then present how we compute the specificity of expressions. Finally, we combine the refinement operator and the specificity function to learn graph invariants.

Operator

Let $\mathbb{A}(\mathfrak{F})$ be the space of all arithmetic expressions over a finite set $\mathfrak{F}$ of predefined real-valued functions over the set of all RDF graphs and a finite set of binary arithmetic operations $\mathscr{O} = \{+, -, \times, /\}$. We denote the i-th element of $\mathfrak{F}$ with $\mathfrak{f}_i$.

Example 3.1. *We can imagine $\mathfrak{F}$ to be the set of functions $\{\mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\max}}\}$, which return the minimal ($\mathrm{d}_{\min}$) and maximal ($\mathrm{d}_{\max}$) degree of resources in a graph, respectively.*

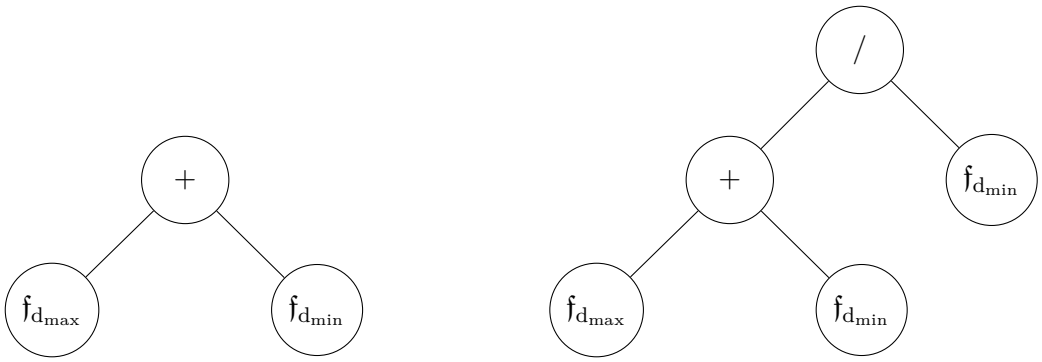
Every arithmetic expression $\lambda \in \mathbb{A}(\mathfrak{F})$ created with the binary operators $\mathscr{O}$ can be naturally represented as a binary expression tree [219]. We say that an expression λ_1 is subsumed by an expression λ_2 (denoted $\lambda_1 \sqsubseteq \lambda_2$) iff λ_1's tree representation is a subtree of λ_2's tree representation.

Example 3.2. $\lambda_1 = (\mathfrak{f}_{\mathrm{d}_{\max}} + \mathfrak{f}_{\mathrm{d}_{\min}})$ *is subsumed by* $\lambda_2 = (\mathfrak{f}_{\mathrm{d}_{\max}} + \mathfrak{f}_{\mathrm{d}_{\min}})/\mathfrak{f}_{\mathrm{d}_{\min}}$.

The subsumption relation defines a partial ordering over $\mathbb{A}(\mathfrak{F})$. Let $\circledast \in \mathscr{O}$ be a binary operator. We define the operator $\rho : \mathbb{A}(\mathfrak{F}) \to 2^{\mathbb{A}(\mathfrak{F})}$ as follows:

$$\rho(\lambda) = \begin{cases} \mathfrak{F} & \text{if } \lambda \text{ is the empty expression } \lambda_\emptyset, \\ \bigcup\limits_{\mathfrak{f}_i \in \mathfrak{F}} \bigcup\limits_{\circledast \in \mathscr{O}} \{\lambda \circledast \mathfrak{f}_i\} & \text{else.} \end{cases} \tag{3.16}$$

Example 3.3. *Let* $F = \{\mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\max}}\}$. *Then* $\rho(\mathfrak{f}_{\mathrm{d}_{\min}}) = \{\mathfrak{f}_{\mathrm{d}_{\min}} + \mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\min}} - \mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\min}} \times \mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\min}} / \mathfrak{f}_{\mathrm{d}_{\min}}, \mathfrak{f}_{\mathrm{d}_{\min}} + \mathfrak{f}_{\mathrm{d}_{\max}}, \mathfrak{f}_{\mathrm{d}_{\min}} - \mathfrak{f}_{\mathrm{d}_{\max}}, \mathfrak{f}_{\mathrm{d}_{\min}} \times \mathfrak{f}_{\mathrm{d}_{\max}}, \mathfrak{f}_{\mathrm{d}_{\min}} / \mathfrak{f}_{\mathrm{d}_{\max}}\}$.



(a) Tree of λ_1. **(b)** Tree of λ_2.

Figure 3.11.: A graphical representation of the two binary expression trees of Example 3.2's expressions.

We call two arithmetic expressions λ_1 and λ_2 in $\mathbb{A}(\mathfrak{F})$ equivalent iff they return the same value for all input graphs. Based on this definition of equivalence, we can show that ρ is a finite, redundant, and complete refinement operator over $(\mathbb{A}(\mathfrak{F}), \sqsubseteq)$:[27]

ρ is a refinement operator. By virtue of the construction of ρ, it is evident that $\forall\lambda \in \mathbb{A}(\mathfrak{F})\ \forall\lambda' \in \rho(\lambda) : \lambda \sqsubseteq \lambda'$. Given that $\forall\lambda \in \mathbb{A}(\mathfrak{F}) : \lambda_\emptyset \sqsubseteq \lambda$ because the tree representation of $\lambda_\emptyset$ is the empty tree, we can conclude that $\forall\lambda' \in \rho(\lambda) : \lambda \sqsubseteq \lambda'$. By virtue of the definition of refinement operators given by van der Laag et al. [294], we can conclude that ρ is a *refinement operator*.

ρ is finite. A refinement operator is called *finite* if the set of one-step refinements, i.e., the number of elements created with a single application of the refinement operator for a given input, is finite [294]. The *finiteness* of ρ is given by $|\rho(\lambda)| = |\mathscr{O}||\mathfrak{F}| < \infty$ for all non-empty expressions and $|\rho(\lambda_\emptyset)| = |\mathfrak{F}| < \infty$ for the empty expression.

ρ is redundant. We call a refinement operator *redundant* if at least two different sequences of application of the refinement operator can lead to the same expression [294]. ρ is *redundant* because there are two refinement paths from $\lambda_\emptyset$ to the equivalent expressions $\mathfrak{f}_1 + \mathfrak{f}_2$ and $\mathfrak{f}_2 + \mathfrak{f}_1$, i.e., $\lambda_\emptyset \rightarrow \mathfrak{f}_1 \rightarrow \mathfrak{f}_1 + \mathfrak{f}_2$ and $\lambda_\emptyset \rightarrow \mathfrak{f}_2 \rightarrow \mathfrak{f}_2 + \mathfrak{f}_1$.

ρ is complete. A refinement operator is called *complete* if it can generate an expression λ equivalent to any $\lambda' \in \mathbb{A}(\mathfrak{F})$ [294].[28] Equation 3.3 shows that ρ refines a given expression λ by appending a binary operation and a function on the right side. If λ is represented as a binary expression tree the refinement changes this tree by using

[27] For the sake of space, we refer the interested reader to van der Laag et al. [294] for more details on refinement operators.

[28] Our definition of completeness covers global completeness. We do not claim that our refinement operator offers local completeness [195].

the tree's root node as left child of the newly added operator and making the new operator the new root node with the added function as it's right child. Figure 3.11 shows an example based on the two expressions of Example 3.2. This leads to unbalanced trees since the left subtree of a node might be a complex expression while the right node is always a single function. We name this a left deep tree. Hence, ρ is complete if it is possible that every binary expression tree over $\mathscr{O}$ can be represented as such a left deep tree. It is possible to transform many expressions directly into a left deep tree. Section A.1 lists several basic examples. However, since our refinement operator is able to generate polynomials, it can create Taylor series that approximate expressions that cannot be transformed directly into a left deep tree [304].[29]

Specificity

We can compute how characteristic an expression λ is for $\mathscr{G}$ by measuring the invariance of its values over all graphs in $\mathscr{G}$ and by comparing it with negative example graphs. We begin by using a set of graph generators $\mathfrak{G}$ for generating a set of negative examples $\tilde{\mathscr{G}}$ made up of $|\mathfrak{G}| \times |\mathscr{G}|$ graphs $\left\{\tilde{\mathcal{G}}_1, \ldots, \tilde{\mathcal{G}}_{|\mathfrak{G}|\times|\mathscr{G}|}\right\}$. $\mathfrak{G}$ can comprise any off-the-shelf graph generator. During the generation, we ensure that for each generator in $\mathfrak{G}$, $\forall i \in [1, |\mathscr{G}|] : |V_{\mathcal{G}_i}| = |V_{\tilde{\mathcal{G}}_i}|$. The set of negative examples is used to contrast the positive examples found in $\mathscr{G}$ during the learning of the graph-specific invariants. First, we use the following variance-inspired measure to compute how close λ is to being an invariant of $\mathscr{G}$:

$$h(\lambda, \mathscr{G}) = 1 - \frac{\sum_{i=1}^{|\mathscr{G}|}\sum_{j=1}^{|\mathscr{G}|}(\lambda(\mathcal{G}_i) - \lambda(\mathcal{G}_j))^2}{\max\left(\left\{\left(\lambda(\mathcal{G}_1)\right)^2, \ldots, \left(\lambda(\mathcal{G}_{|\mathscr{G}|})\right)^2\right\}\right)|\mathscr{G}|(|\mathscr{G}|-1)}\,. \tag{3.17}$$

For invariants, $h(\lambda, \mathscr{G}) = 1$. h treats expressions of all lengths the same. For the sake of computational efficiency, we would want h to prefer shorter expressions over longer ones. To achieve this goal, we extend h by defining h' as follows:

$$h'(\lambda, \mathscr{G}) = h(\lambda, \mathscr{G}) - u|\lambda|\,. \tag{3.18}$$

where $|\lambda|$ is the number of arithmetic operators in λ and $u \in [0, 1]$ is a small constant.[30]

[29] Note that for some expressions the Taylor series does not converge [304]. Hence, the refinement operator would have to generate an infinite long polynomial to approximate such an expression.

[30] We set $u = 0.1$ in all experiments.

While h' captures how close λ is to being a short invariant on $\mathscr{G}$, it fails to capture how specific this expression is for $\mathscr{G}$. For example, while the expression $\mathfrak{f}_1 - \mathfrak{f}_1$ is a trivial invariant for $\mathscr{G}$, it is also an invariant for any non-empty set of graphs. We alleviate this problem by using the following function:

$$\mathfrak{s}(\lambda, \mathscr{G}, \tilde{\mathscr{G}}) = \frac{2h'(\lambda, \mathscr{G})(1 - h'(\lambda, \tilde{\mathscr{G}}))}{h'(\lambda, \mathscr{G}) + (1 - h'(\lambda, \tilde{\mathscr{G}}))}\,. \tag{3.19}$$

$\mathfrak{s}(\lambda, \mathscr{G}, \tilde{\mathscr{G}})$ is the harmonic mean of $h'(\lambda, \mathscr{G})$ and $1 - h'(\lambda, \tilde{\mathscr{G}})$ and is a measure of the specificity of λ as an invariant for $\mathscr{G}$. For $u = 0$, $\mathfrak{s}(\lambda, \mathscr{G}, \tilde{\mathscr{G}}) = 1$ if λ is an invariant of $\mathscr{G}$ (i.e., $h'(\lambda, \mathscr{G}) = 1$) and not an invariant for $\tilde{\mathscr{G}}$ (i.e., $h'(\lambda, \tilde{\mathscr{G}}) = 0$).

Learning Approach

The invariants for $\mathscr{G}$ can be learned as follows. We begin by generating $\tilde{\mathscr{G}}$ using off-the-shelf graph generators. As suggested by previous publications on negative sampling (see, e.g., [264]), the choice of the models should not affect our results and is hence not further analyzed in this work. We initialize the set of candidate expressions Λ with $\{\lambda_\emptyset\}$. The set Λ' of seen expressions is initialized with $\emptyset$. We then iterate the following three steps a predefined number of times:[31]

1. Selection: $\lambda_{\max} = \underset{\lambda \in \Lambda \setminus \Lambda'}{\operatorname{argmax}}\, \mathfrak{s}(\lambda, \mathscr{G}, \tilde{\mathscr{G}})$.[32]
2. Refinement: $\Lambda = \Lambda \cup \rho(\lambda_{\max})$.
3. Update: $\Lambda' = \Lambda' \cup \{\lambda_{\max}\}$.

Finally, we select the set $\Lambda_{\max}$ comprising the best performing expressions as our final output.

3.5.3 Initial Graph Generation

The initial graph generation step creates a first graph $\dot{\mathcal{G}}_0$ of the target size. To generate the graph, the number of edges is computed based on the given number of IRI vertices ν_R and the average density $\delta_{\mathscr{G}}$. After that, the classes and properties are assigned to the vertices and edges based on the class and property distributions, respectively. It should be noted that the class assignment results in the two mappings $\mathfrak{c}_{\dot{\mathcal{G}}_0}$ and $\mathfrak{i}_{\dot{\mathcal{G}}_0}$ as defined in Section 2.1.2. Finally, the edges are used to connect the

[31] In our experiments, we use 50 iterations.

[32] Given that ρ is redundant, we exploit the commutativity and the associativity of some arithmetic operations to detect and remove duplicate expressions from Λ in our implementation.

vertices. We achieve this by applying the following three steps for each edge. First, the set of possible classes for the subject and the object of the edge are determined. Second, the classes of the two endpoints (C_s and C_o) of the edge are chosen from these sets. Third, two instances are chosen which will be connected by the edge from the two sets of vertices that are instances of the chosen classes.

Class Set Selection

This first step uses the previously collected constraints $\Omega_{\mathscr{G}}$. Let p be a property and $\omega_s = (p, C_o)$ be a function that returns a set of sets of classes whose instances are potential subjects of edges with p and a an object v_i with $\mathfrak{c}_{\dot{\mathcal{G}}_0}(v_i) = C_o$. Let $\omega_o = (p, C_o)$ be a similar function for potential object classes. We define the two function as follows:

$$\omega_s(p, C_o) = \{C_s | (C_s, p, C_o) \in \Omega_{\mathscr{G}}\}, \tag{3.20}$$

$$\omega_o(C_s, p) = \{C_o | (C_s, p, C_o) \in \Omega_{\mathscr{G}}\}. \tag{3.21}$$

Both functions can be used as $\omega_s(p, *)$ and $\omega_o(*, p)$ where $*$ donates any set of classes.

Endpoint Class Definition

We propose three different approaches for selecting the set of classes of the two endpoints of a given edge. The approach *Uniform Class Selection (UCS)* randomly draws C_s from $\omega_s(p, *)$ using a uniform distribution. In the same way, C_o is chosen from $\omega_o(C_s, p)$.

The approach *Biased Class Selection (BCS)* relies on the $\mathbb{P}_{\mathscr{G}}\left((C_s, C_o) | \, p\right)$ probabilities of the different class sets to sample the class sets for the subject and object of the edge. For each set of classes $C_i \in \omega_s(p, *)$ the probability $\mathbb{P}_{\mathscr{G}}\left((C_s, *) | \, p\right)$ is used. It is determined as follows:

$$\mathbb{P}_{\mathscr{G}}\left((C_i, *) | \, p\right) = \sum_{C_j \in \omega_o(C_i, p)} \mathbb{P}_{\mathscr{G}}\left((C_i, C_j) | \, p\right). \tag{3.22}$$

Based on these probabilities, a class set C_s is sampled for the subject of the edge. Based on C_s, a set of classes is sampled for the object of the edge. For each possible class set $C_i \in \omega_o(C_s, p)$ the probability $\mathbb{P}_{\mathscr{G}}\left((C_s, C_i) | \, p\right)$ is used for that.

While UCS and BCS sample C_s before C_o, the *Clustered Class Selection (CCS)* samples both class sets at the same time. For each possible class set pair (C_i, C_j) with $(C_i, p, C_j) \in \Omega_{\mathscr{G}}$, its probability $\mathbb{P}_{\mathscr{G}}\left((C_i, C_j)|\, p\right)$ is used.

Vertex Selection

After the classes for the subject and object vertices of the edge are chosen, the two single vertices with these classes have to be sampled. For sampling two vertices, the *Uniform Instance Selection (UIS)* assigns a uniform probability to all vertices of the sets $\mathfrak{i}_{\dot{\mathcal{G}}_0}(C_s)$ and $\mathfrak{i}_{\dot{\mathcal{G}}_0}(C_o)$, respectively.

In contrast, the *Biased Instance Selection (BIS)* approach uses the degree distributions $\varsigma_{C_j,\mathscr{G}}$ to assign degree weights to the single vertices. For each vertex $v_i \in V$, a degree weight $\mathfrak{w}_i$ is sampled from $\varsigma_{\mathfrak{c}_{\dot{\mathcal{G}}_0}(v_i),\mathscr{G}}$. Based on these weights, a probability $\mathbb{P}\left(v_i|\, C_j\right)$ is assigned to each vertex to be chosen when sampling a vertex for a given set of classes C_j. The probability is defined as

$$\mathbb{P}\left(v_i|\, C_j\right) = \mathfrak{w}_i \Big/ \sum_{v_k \in \mathfrak{i}_{\dot{\mathcal{G}}_0}(C_j)} \mathfrak{w}_k\,. \tag{3.23}$$

The chosen vertices are connected by the given edge. However, if both vertices are already connected with an edge that has the same property p two new vertices have to be sampled. By combining the three approaches for selecting the subject and object classes for an edge with the two techniques to select the single vertices, six approaches are obtained: UCS-UIS, UCS-BIS, BCS-UIS, BCS-BIS, CCS-UIS, and CCS-BIS.

3.5.4 Graph Amendment

The initial graph $\dot{\mathcal{G}}_0$ is further amended iteratively based on the set of characteristic expressions determined on the set of original graphs. To this end, we define an error score that is used to measure the difference between the values of the invariant expressions for the original graphs $\mathscr{G}$ and the generated graph $\dot{\mathcal{G}}_k$. Let Λ_{max} be the set of the best invariant expressions learned on $\mathscr{G}$ as described in Section 3.5.2. Let μ_i be the average value the expression λ_i returns for the original graphs and let σ_i be its standard deviation. Let $\mathfrak{a}(\dot{\mathcal{G}}_k, \lambda_i, \mu_i, \sigma_i)$ be the difference function defined as follows:

$$\mathfrak{a}(\dot{\mathcal{G}}_k, \lambda_i, \mu_i, \sigma_i) = \frac{(\lambda_i(\dot{\mathcal{G}}_k) - \mu_i)^2}{\sigma_i^2}\,. \tag{3.24}$$

Let $\varepsilon(\dot{\mathcal{G}}_k)$ be the error of graph $\dot{\mathcal{G}}_k$ with respect to the characteristic expressions defined as follows:

$$\varepsilon(\dot{\mathcal{G}}_k) = \sum_{i=1}^{|\Lambda_{\max}|} \mathfrak{a}(\dot{\mathcal{G}}_k, \lambda_i, \mu_i, \sigma_i)\,. \tag{3.25}$$

The target of the amendment phase is to optimize for the error score of the generated graph by successive modifications. We achieve this goal by using a greedy approach. In each iteration, the algorithm generates two new versions of $\dot{\mathcal{G}}_k$ by adding or removing a random edge, respectively. Thereafter, the graph with the lower error score is used for the next iteration. The amendment phase ends when a maximum number of iterations is reached or no improvement has been achieved for several iterations. The removal of an edge randomly chooses an edge and removes it. The addition of a new edge starts with choosing a property p for the edge following the property distribution. Based on the chosen property, the same steps as during the generation of the initial graph are executed to assign subject and object vertices to the newly generated edge.

3.5.5 Graph Completion

The completion phase takes the result graph $\dot{\mathcal{G}}_k$ of the amendment phase as input and extends it to form the final, complete graph $\dot{\mathcal{G}}$. First, datatype edges are created. For each set of classes C_j and each datatype property $p \in P_{L,\mathcal{G}}$, the number of p edges the instances of C_j should have is determined by multiplying the number of instances with the average degree $\mathrm{d}_{C_j,p,\mathcal{G}}$. Second, for each datatype edge, a literal is generated by sampling a literal value from the previously learned distribution $\ell_{p,\mathcal{G}}$. Third, the datatype edges are connected to a resource node within the graph. We sample the vertex for this connection from the set of instances of C_j.

Finally, the graph is transformed into an RDF graph representation. To this end, each resource vertex of the graph receives a generated IRI. With these IRIs, the graph can be transformed into an RDF triple representation. After that, the `rdf:type` triples are generated, i.e., for each vertex $v_j \in V_{\dot{\mathcal{G}}_k}$ and class $c \in \mathfrak{c}_{\dot{\mathcal{G}}_k}(v_j)$ an RDF triple $(v_j,$ `rdf:type` $, c)$ using the IRIs of v_j and c, respectively.

3.6 Evaluation

The HOBBIT platform has already been used successfully in a large number of challenges (see Section 3.7). Still, we evaluated our architecture in two different

Table 3.2.: Platform benchmark results on a single machine (1 – 3) and a cluster (4, 5) (std. dev. = standard deviation).

Experiments	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Data generators	2	2	2	1	3
Task generators	1	1	1	1	1
Queries	1,000	2,000	5,000	100,000	300,000
Avg. query runtime (in ms)	7,058	17,309	33,561	38,810	59,828
Query runtime std. dev.	686	4,493	3,636	22,517	24,540
Overall runtime (in s)	11.2	32.4	51.5	2,086	2,536
Queries per second (avg.)	44.9	31.0	48.6	865.1	774.2

respects. First, we simulated benchmarking triple stores using HOBBIT. These experiments had two goals. First, we wanted to prove that the HOBBIT platform can be used on single, lightweight hardware (e.g., for development purposes or for benchmarks where the scalability and runtime are not of importance) as well as in a distributed environment. Second, we wanted to evaluate the throughput of storage benchmarks. In addition, we benchmarked several knowledge extraction tools and studied the runtime performance of these systems for the first time.

In a third experiment, we used our graph mimicking algorithm LEMMING to generate several synthetic graphs using the different variants of our approach. We benchmarked triple stores based on the generated synthetic graphs and compared their performance and ranking with the benchmark results we achieved on a held-out dataset.

3.6.1 Triple Store Benchmark

To configure our simulation, we derived message characteristics from real data using the Linked SPARQL Queries Dataset [242]—a collection of SPARQL query logs. This collection of real query logs suggests that 1) the average length of a SPARQL query is 545.45 characters and 2) the average result set comprises 122.45 bindings. We assumed that the average size of a single result is 100 characters leading to a result set size of approximately 12,200 characters which is created for every request by our triple store simulation.

The platform was deployed on a small machine and on a server cluster.[33] The single benchmark runs are shown in Table 3.2. We executed the benchmark with three

[33] Single machine specifications: Dual Intel Core i5, 2.5 GHz, 4 GB RAM.
Cluster specifications: 1 master server (1xE5-2630v4 10-cores, 2.2GHz, 128GB RAM) hosting platform components (including RabbitMQ message broker), 1 data server (1xE5-2630v3 8-cores, 2.4GHz,

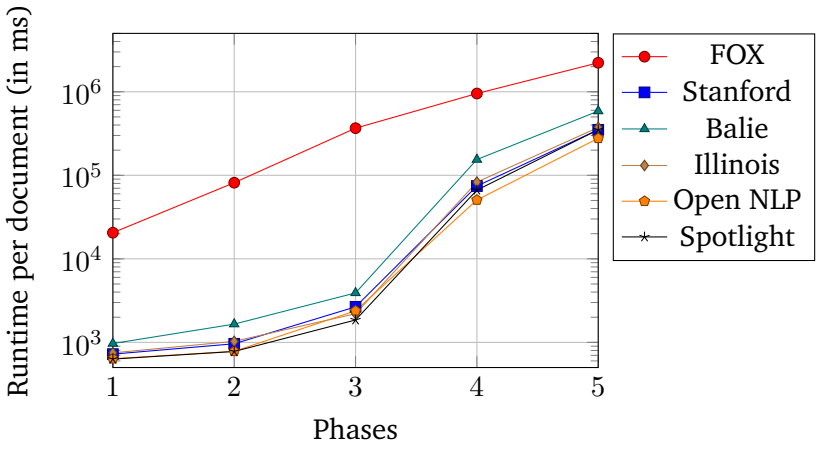


Figure 3.12.: Average runtime per document achieved by systems during the different phases.

Table 3.3.: The effectiveness of the benchmarked systems (Micro measures).

System	**Precision** (Pr_{mic})	**Recall** (Re_{mic})	**F1-measure** ($F1_{mic}$)
Balie	0.321	0.293	0.306
FOX	0.505	0.543	0.523
Illinois	0.524	0.614	0.565
OpenNLP	0.351	0.233	0.280
Spotlight	0.513	0.411	0.456
Stanford	0.548	0.662	0.600

different numbers of queries on the smaller machine and two larger numbers of queries on the cluster. Our results show that the platform can run even on the minimalistic single machine chosen for our evaluation. Hence, the HOBBIT platform can be used locally for smoke tests and development tests. In addition, our results also clearly indicate the need for a platform such as HOBBIT by pointing to the necessity to deploy benchmarking platforms in a large-scale environment to test some of the Big Linked Data systems. Experiments with 5000 queries run on the small machine clearly show an increase in the average runtime per query and the standard deviation of the query runtimes due to a traffic jam in the message bus queues. In contrast, our results on the cluster show that we are able to scale up easily and run 20 times more queries per second than on the single machine.

3.6.2 Knowledge Extraction Benchmark

For our second evaluation, we used Task 1B of the Open Knowledge Extraction challenge 2017 [258] as use case. This task comprises the problem of spotting named entities from a given text and linking them to a given knowledge graph. All experiments were run on our cluster. We benchmarked the following named entity recognition tools:

1. FOX [257],
2. The Ottawa Baseline Information Extraction (Balie) [196],
3. The Illinois Named Entity Tagger (Illinois) [222],
4. The Apache OpenNLP Name Finder (OpenNLP) [27],
5. The Stanford Named Entity Recognizer (Stanford) [97], and
6. DBpedia Spotlight (Spotlight) [179].

The entities that were found in the text by any of the tools are linked to a given knowledge graph using AGDISTIS [190]. In our experiment, we used DBpedia 2015 as the reference knowledge graph.[34]

The aim of the benchmark was to measure the scalability and the accuracy of these systems under increasing load, an experiment which was not possible with existing benchmarking solutions. We used a gold standard made up of 10,000 documents generated using the BENGAL generator included in the HOBBIT platform.[35] The evaluation module was based on the evaluation used for the Open Knowledge Extraction challenge [258] and measured the runtime for single documents as well as the result quality in terms of micro precision, recall, and F1-measure. We used 1 data and 1 task generator for our benchmark. The data generator was configured to run through 5 velocity phases (2000 documents/phase) with differing delays between single documents in each phase. The delays between the documents were set to $\{1s, \frac{1}{2}s, \frac{1}{4}s, \frac{1}{8}s, 0s\}$ leading to an increasing workload of $\{1, 2, 4, 8, \approx 800\}$ documents per second.

The results presented in Figure 3.12 show that all approaches scale well when provided with enough hardware. As expected, FOX is the slowest solution as it relies on calling 5 underlying fully-fledged entity recognition tools and merging their results. Our results also indicate that a better load balancing could lead to even better runtimes. In particular, the runtime per document starts to increase as soon as the tool cannot handle the incoming amount of documents in time and the

64GB RAM) hosting storages, 6 nodes (2xE5-2630v3 8-cores, 2.4GHz, 256GB RAM) divided into two groups hosting either components of the benchmark or the benchmarked system.

[34] `http://dbpedia.org`; last accessed on 03.08.2022.

[35] `http://github.com/dice-group/bengal`; last accessed on 03.08.2022.

documents start to be queued (see Phase 2 to 4). Additionally, the results show that Balie is slower than the other fully-fledged entity recognition tools. Given that Balie also has the lowest F1-score (see Table 3.3) it can be argued that removing Balie from FOX could be an option to increase its efficiency.[36]

3.6.3 Graph Mimicking Experiment

We evaluate the graph generation LEMMING based on three different real-world datasets and four different triple stores. The main aim of our evaluation is to measure the performance of the selected triple stores based on our generated datasets and compare it with that achieved by the same triple stores on an unseen version of the dataset.

The three datasets we use, i.e., Semantic Web Dog Food (SWDF), Linked Geo Data (LGD), and the International Chronostratigraphic Chart (ICC), are such that at least three different versions are available. We use the latest version of each dataset as held-out graph and its size as input parameter ν_R for our generation algorithm. We use the six versions of LEMMING and compare it with a baseline algorithm to generate graphs. Since all approaches are based on sampling mechanisms, we execute each algorithm three times. After that, we evaluate four reference triple stores—Virtuoso, Blazegraph, Fuseki, and GraphDB—on the held-out as well as the generated datasets using IGUANA [67].[37] IGUANA is a generic SPARQL benchmark execution framework. It can be used to benchmark different triple stores with different datasets in a comparable way. During the benchmarking, we measure the query mixes per hour (QMpH) and queries per second (QpS). QpS is measured for each query while QMpH summarizes the overall performance of a triple store over all queries. The similarity between the measured values is calculated using the Spearman rank correlation (SRC) for the QMpH values and the root mean squared error (RMSE) for the QpS values.

[36] All experiment results are available at `https://master.project-hobbit.eu/experiments/1501852310576,1501852574348,1501852527351,1501852487461,1501852242060,1501852152692`; last accessed on 03.08.2022.

[37] The stores are available at `https://github.com/openlink/virtuoso-opensource/releases`, `https://blazegraph.com/`, `https://jena.apache.org`, and `https://graphDB.ontotext.com/`, respectively (last accessed on 03.08.2022).

Table 3.4.: Features of the target graphs of the different datasets.

	SWDF	LGD	ICC
Triples	445 821	3 387 842	12 742
Resources (ν_R)	45 423	591 649	1 423
Queries	20	43	27

Datasets

SWDF comprises data about Semantic Web conferences from 2001 to 2015.[38] The data mainly focuses on persons, events, papers, and organizations related to these conferences. Since the dataset is designed to build one version upon the previous version we define it to have 15 versions—one version per year. Each version comprises the previous version and the data of the conferences of the next year. The last version of 2015 is the held-out version. The LGD dataset is a subset of the Linked Geo Dataset [263]. We use the `Military` and `Craft` files of the three consecutive versions of 2013, 2014, and 2015. The latter is used as held-out version. The third dataset, ICC, represents the chronostratigraphic chart as RDF [69, 70, 71, 72]. This chart defines the geological time intervals including their names, start, and end dates as well as their relations to each other. The dataset has been updated several times leading to twelve versions in the years 2004–2018. All versions of all three datasets are preprocessed by materializing all implicit knowledge that can be inferred based on the ontology of the datasets. Table 3.4 shows the features of the target graphs.

We use LSQ [242] to retrieve real user queries to the datasets from query logs. We use FEASIBLE [243] to generate benchmark queries from the LSQ queries, which can be used to benchmark the triple stores based on the different datasets. Table 3.4 shows the number of queries generated for the different datasets. For each dataset, the ontology is retrieved. If a dataset makes use of more than one ontology, the intersection of the ontologies is used. For each query, every IRI that is not contained in the respective ontology (i.e., each IRI that is neither a class nor a property) is replaced by a template variable [67]. IGUANA replaces these variables on the fly with resources from the graph used for benchmarking. This leads to several queries with different resources. It is ensured that only queries with a non-empty result are used for the benchmarking. This allows the usage of queries comprising instance IRIs although the target graph as well as the generated graphs have different instance IRIs.

[38] `https://old.datahub.io/dataset/semantic-web-dog-food`; last accessed on 03.08.2022.

Table 3.5.: Set of metrics $\mathfrak{F}$ used for the search of invariant expressions.

Metric	Description
`#edges`	The number of edges.
`#vertices`	The number of vertices.
`avgDegree`	Average vertex degree.
`maxInDegree`	The highest in-degree of a vertex found in the graph.
`maxOutDegree`	The highest out-degree of a vertex found in the graph.
`stdDevInDegree`	Standard deviation of the vertex in-degrees.
`stdDevOutDegree`	Standard deviation of the vertex out-degrees.
`#eTriangles`	Number of unique triangles formed by three edges.
`#vTriangles`	Number of unique triangles formed by three vertices.

Configuration

Table 3.5 shows the set $\mathfrak{F}$, i.e., the set of metrics that are used to learn the invariant expressions of the input graphs. The refinement operator is configured to use 50 iterations for its search with $u = 0.1$. As a set of graph generators $\mathfrak{G}$ for negative examples $\tilde{\mathscr{G}}$ we use generators for star, ring, grid, clique, and bipartite graphs. Our algorithm is configured to use a maximum of 50 000 iterations to reduce the error score during the amendment phase. The phase ends earlier if the error score does not improve for 5 000 iterations. Further, we configure the BIS approaches to rely on Poisson distributions to mimic the degree distributions $\varsigma_{C_j,\mathscr{G}}$. The distribution parameters are learned for each C_j individually.

Depending on the datatype of literals, we configure the algorithm to use different literal value distribution types. For datatype properties with literals that have a numeric, `Date` or `DateTime` datatype, we determine the minimum and maximum values. After that, we define the literal value distribution $\ell_{p,\mathscr{G}}$ for such a time-related property p as uniform distribution in the determined range of the minimum and maximum value. All other literals are treated as datatype string. For the generation of such literals, we define a distribution that always returns a new string making all string-based literals unique.

Baseline

An analysis of the datasets showed that they do not have a common type of degree distribution, i.e., it is not possible to assign them to a common class of graphs like scale-free or Poisson graphs. However, since it has been shown that the degree distributions of a large number of RDF datasets follow a power-law distribution [94,

Table 3.6.: Invariant characteristic expressions per dataset.

	ID	Expression
SWDF	λ_1	maxInDegree /((#vertices × stdDevOutDegree)+ maxInDegree)
	λ_2	maxInDegree /(#vertices + maxInDegree − stdDevOutDegree)
	λ_3	maxInDegree /((#vertices / maxInDegree) + maxInDegree)
	λ_4	maxInDegree /(#edges+ maxInDegree − stdDevOutDegree)
	λ_5	maxInDegree /((#vertices / stdDevOutDegree)+ maxInDegree)
LGD	λ_1	(2× #vertices − #edges)/(#edges × avgDegree)
	λ_2	#vertices /(#edges × avgDegree^2)
	λ_3	(#vertices − #edges)/(#edges + maxOutDegree − #vertices)
	λ_4	#vertices /(#edges ×(avgDegree −1.0))
	λ_5	(#vertices − #edges)/(#edges + maxInDegree − #vertices)
ICC	λ_1	maxInDegree /((#edges − #vertices)+ maxInDegree)
	λ_2	maxInDegree /((#vertices / maxOutDegree)+ maxInDegree)
	λ_3	maxInDegree /((#vertices − #edges)+ maxInDegree)
	λ_4	maxInDegree /((#vertices × stdDevInDegree)+ maxInDegree)
	λ_5	maxInDegree /((#vertices / maxInDegree)+ maxInDegree)

312] we use an implementation of the Barabasi-Albert model [9]. This algorithm adds one node after the other to the graph by creating $\delta_{\mathscr{G}}$ new directed edges. The direction of the edge is sampled from a Bernoulli distribution with the probability 0.5 for both cases. The second vertex for each edge is sampled based on the degree of the vertices, i.e., the higher the degree, the higher the probability that a vertex is chosen. After the generation of the graph, the properties and node types are sampled from $\mathbb{P}_{\mathscr{G}}(p)$ and $\mathbb{P}_{\mathscr{G}}(C)$, respectively.

Results

Tables 3.6 and 3.7 summarize the results of the graph generation process. Table 3.6 shows the graph invariants per dataset. Table 3.7 shows the final error score $\varepsilon(\dot{\mathcal{G}})$ and the runtimes of the different graph generation approaches—the baseline (BL) and the six combinations of the three class selection variants Uniform Class Selection (UCS), Biased Class Selection (BCS), and Clustered Class Selection (CCS) as well as the two instance selection variants Uniform Instance Selection (UIS) and Biased Instance Selection (BIS). Table 3.8 shows a summary of the triple store evaluation. It contains the rank correlation of the QMpH values and the RMSE for the QpS values.

Tables 3.6 and 3.7 summarize the results of the graph generation process. Table 3.6 shows the graph invariants per dataset, which clearly differ across datasets. A

Table 3.7.: Average results of the different expressions on the original graphs, and difference of the average values on the target graph and the generated graphs in percentages of the original graphs' average. BL marks the baseline approach. The last two lines of the results on a dataset contain the average error scores $\varepsilon(\dot{\mathcal{G}})$ of the generated graphs and the average runtimes. †,††,†††—1, 2 or all 3 runs terminated before reaching the maximum number of iterations, respectively.

	Exp.	Original graphs	Target graph	UCS		BCS		CCS		BL
				UIS	BIS	UIS	BIS	UIS	BIS	
SWDF	λ_1	0.0926	-16.05%	-61.86%	-63.24%	3.48%	0.00%	0.00%	0.00%	-99.18%
	λ_2	-0.1751	-15.72%	0.79%	0.69%	-0.59%	-0.09%	-0.09%	-0.09%	-99.33%
	λ_3	0.9997	0.03%	0.03%	0.03%	0.03%	0.03%	0.03%	0.03%	-35.10%
	λ_4	0.1294	-11.98%	0.77%	0.69%	-0.26%	0.12%	0.12%	0.12%	-99.09%
	λ_5	0.9965	0.15%	0.28%	0.29%	-0.11%	-0.11%	0.00%	-0.06%	-35.50%
	Error $\varepsilon(\dot{\mathcal{G}})$			0.3076	0.3195	0.0034	0.0011	**0.0002**	0.0004	1.4299
	Runtime (in hours)			3.4	3.5	† 2.9	3.5	††† 0.9	††† 0.8	0.0
LGD	λ_1	-0.1250	-0.06%	0.02%	0.03%	0.02%	0.02%	-0.03%	0.02%	0.03%
	λ_2	0.0160	-11.17%	-1.00%	-1.43%	-0.96%	-0.13%	4.19%	-0.07%	-4.00%
	λ_3	-0.9978	0.12%	-0.04%	-0.07%	0.17%	0.17%	-0.02%	0.21%	0.18%
	λ_4	0.0851	-8.77%	-0.77%	-1.11%	-0.74%	-0.09%	3.26%	-0.04%	-3.11%
	λ_5	-0.7857	0.68%	0.04%	0.00%	0.04%	-0.03%	-0.43%	-0.04%	27.22%
	Error $\varepsilon(\dot{\mathcal{G}})$			**0.0008**	0.0014	0.0038	0.0035	0.0046	0.0051	6.7174
	Runtime (in hours)			†† 36.1	† 56.0	48.4	†† 49.1	††† 35.3	†† 51.7	0.1
ICC	λ_1	0.0797	15.63%	0.04%	0.04%	0.04%	0.04%	0.37%	0.04%	-87.79%
	λ_2	0.9936	-0.03%	-0.11%	0.00%	0.00%	0.00%	0.00%	-0.17%	-14.74%
	λ_3	-0.0948	19.05%	-0.04%	-0.04%	-0.04%	-0.04%	0.35%	-0.04%	-89.54%
	λ_4	0.0168	3.02%	0.00%	-0.05%	-1.18%	-0.05%	-5.85%	-0.23%	-56.29%
	λ_5	0.9975	0.02%	0.04%	0.06%	0.04%	0.04%	0.04%	0.03%	-14.18%
	Error $\varepsilon(\dot{\mathcal{G}})$			0.0026	0.0020	0.0008	**0.0007**	0.0022	0.0056	124.8374
	Runtime (in seconds)			††† 290.7	††† 153.3	††† 190.0	††† 111.3	505.3	††† 218.0	0.6

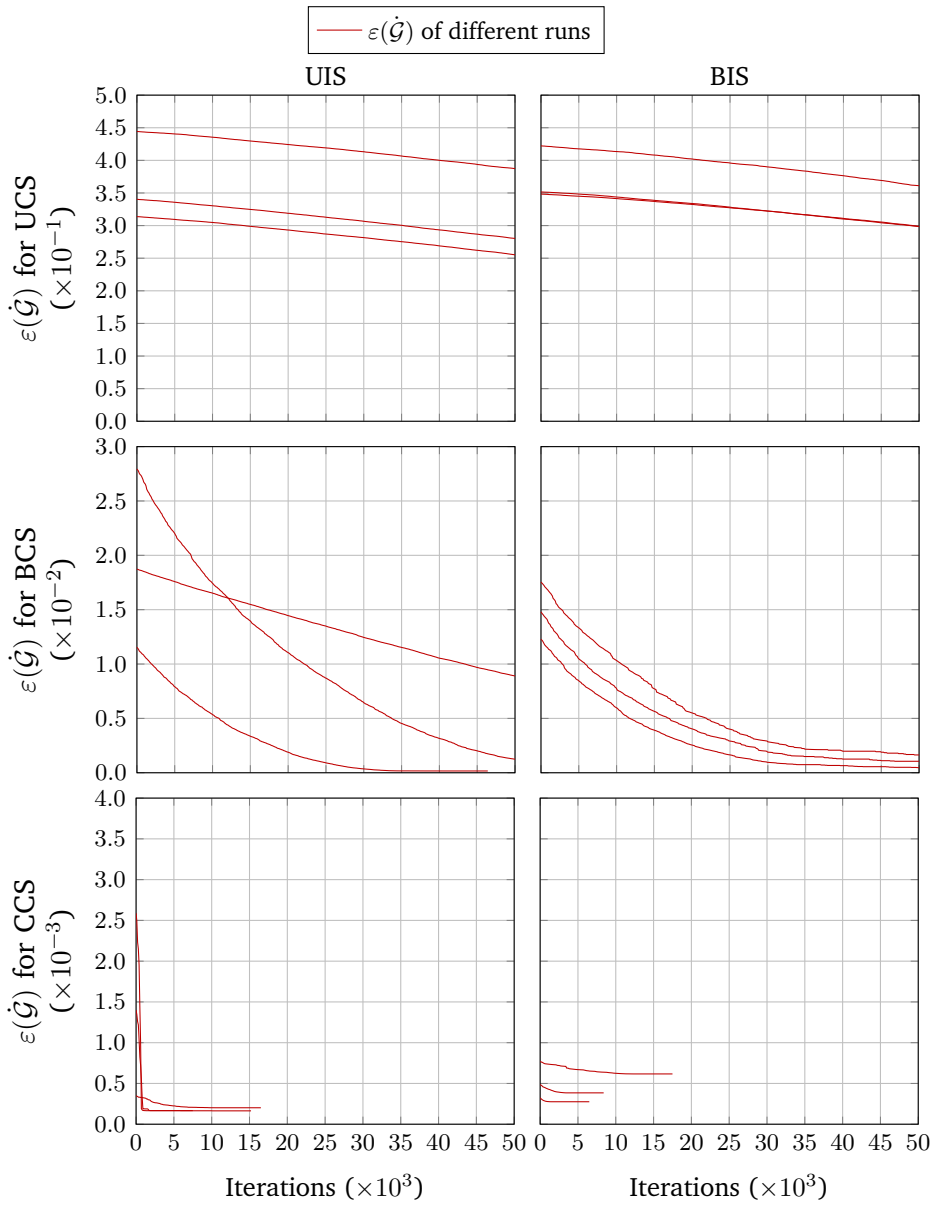


Figure 3.13.: The course of error values during the amendment phase for the SWDF dataset and all variants of LEMMING. The three rows are the class selection variants while the two columns show the instance selection variants.

comparison of the values of the graph invariants for the original graphs, the target graph and the generated graphs is shown in Table 3.7. The difference between the values of the invariants for original graphs and the target graphs are low for most of the graph invariants we learned. These results corroborate the assumption of the existence of graph invariants for RDF datasets.

Table 3.8.: SRC of the triple store systems based on their QMpH on the generated graphs compared to the ranking on the target graph and average RMSE values of the QpS values measured on the target graph and the generated graphs.

Approach	SWDF		LGD		ICC	
	SRC	RMSE	SRC	RMSE	SRC	RMSE
UCS-UIS	0.87	81.1	1.00	**111.8**	1.00	280.9
UCS-BIS	0.93	82.0	1.00	115.3	0.93	**219.6**
BCS-UIS	0.93	88.2	1.00	115.3	1.00	261.2
BCS-BIS	0.93	**64.1**	1.00	117.6	1.00	242.8
CCS-UIS	1.00	72.7	1.00	117.3	1.00	255.6
CCS-BIS	0.93	95.8	1.00	115.9	0.93	229.0
BL	0.32	170.5	1.00	159.1	0.93	222.6

Table 3.7 also shows the overall error $\varepsilon(\dot{\mathcal{G}})$ and the runtimes of the different graph generation approaches. Note that for all three datasets the baseline leads to the generation of graphs with the highest error score $\varepsilon(\dot{\mathcal{G}})$. A comparison of the errors $\varepsilon(\dot{\mathcal{G}})$ of our generation approaches (see Table 3.7) suggests that none is better overall. Still, our results suggest that the different approaches for selecting the tail and head classes for an edge (UCS, BCS, and CCS) have a higher influence on the overall error than the technique to select the single vertices (UIS and BIS). With respect to runtime, all three approaches take several hours for the generation of larger graphs. As expected, the runtimes are shorter for the small ICC graph. The majority of the time is used in the amendment phase. Hence, some approaches lead to shorter runtimes if the amendment phase is stopped earlier after 5 000 iterations without any improvement. The course of the error values for the SWDF dataset is shown in Figure 3.13.[39]

Table 3.8 shows a summary of the triple store evaluation, i.e., the rank correlation of the QMpH values and the RMSE for the QpS values. The QMpH values suggest that the benchmark on SWDF is harder than that on LGD. On both target graphs, Virtuoso shows the best performance with 1,018 and 1,460 QMpH respectively. In contrast, ICC seems to be less hard since all triple stores achieve values up to 434,911 QMpH (GraphDB). The results in Table 3.8 suggest that our approaches show a much better performance than the baseline for the hard SWDF dataset. For the LGD dataset, all generators achieve the same ranking of the triple stores. However, the average RMSE value of the baseline is significantly higher.[40] This is caused by much higher runtimes of the benchmark queries on the BL graphs than on the target graph. For

[39]The figures for the LGD and ICC datasets can be found in the appendix as Figures A.2 and A.3, respectively.

[40]We use a Wilcoxon signed rank test with a threshold of 0.1%.

the ICC dataset, the prediction of the order of the triple stores seems to be trivial as well. However, because of the higher QpS values achieved by all triple stores, a small difference in the query runtime leads to large differences in the calculated QpS values and, hence, to large RMSE values. Although the UCS-BIS approach achieves the smallest RMSE value, its difference to the baseline as well as several other approaches is not significant.[41] Overall, our results suggest that LEMMING is consistently better than the off-the-shelf approach. In addition, the differences across the benchmarks propound that the difference in the performance of LEMMING and the baseline is positively correlated with the difficulty of the benchmark.

3.7 Application

The HOBBIT platform is now being used by more than 3000 registered users that already have executed more than 16000 experiments with more than 40 benchmarks.[42] The HOBBIT platform was also used to carry out 14 benchmarking challenges for Big Data applications. It was used for the Grand Challenge of the 11th and 12th ACM International Conference on Distributed and Event-Based Systems (DEBS 2017 and 2018) [112, 113]. The 2017 challenge was aimed at event-based systems for real-time analytics. Overall, more than 20 participating systems had to identify anomalies from a stream of sensor data.

The Open Knowledge Extraction Challenges 2017 and 2018 used the platform for benchmarking Named Entity Recognition, Entity Linking, and Relation Extraction approaches [258, 259]. For one of the challenge tasks a setup similar to our evaluation in Section 3.6.2 was used. This evaluation revealed that the scalability of some systems decreases drastically under realistic loads. While some of the benchmarked solutions were able to answer single requests efficiently, they became slower than competing systems when challenged with a large amount of requests [258].

The Mighty Storage Challenges 2017 and 2018 focused on benchmarking triple stores [103, 104]. Their RDF data ingestion tasks showed that most triple stores are unable to consume and retrieve triples (e.g., sensor or event data) efficiently. This insight suggests that current triple stores need to significantly improve in their scalability before they can be used for Big Data applications out of the box. The derivation of this insight was made possible by HOBBIT's support of distributed systems and its distributed implementation that allows the generation of enough data and queries to overload the triple stores.

[41] We use a Wilcoxon signed rank test with with a threshold of 2%.

[42] See `http://master.project-hobbit.eu/experiments`; last accessed on 03.08.2022.

3.8 Limitations and Future Work

The HOBBIT benchmarking platform showed its applicability during several challenges and experiments described above. However, the platform comes with some limitations and space for future enhancements which will be discussed in this section.

The FAIR data principles are focusing on data management. Thus, not all of them can be solely realised by the implementation of the platform. There are principles that are at least partly in the responsibility of the organisation hosting the platform. The license for the experiment results has to be defined by the hosting organization (**R1.1**). Similarly, the combination of globally unique, persistent identifiers (**F1**) and making them retrievable (**A1**) is supported by the platform implementation and our online instance is deployed to enable this feature. However, the hosting party of a new instance will have to define another persistent IRI namespace for experiments and organize the redirection of requests from this namespace to the newly deployed instance.

Another limitation can be seen in the fulfillment of **F2**, **R1.3**, and **I2**. The platform is programmed to enhance the metadata of an experiment by adding all metadata that the platform has about itself as well as the metadata of the benchmark and system with which the experiment has been executed. However, since the benchmark and system metadata are user defined their richness as well as the used vocabularies are mainly depending on the user.

The design of the platform comes with two bottlenecks which we addressed by using horizontal scaling. Firstly, the message bus which is used for the communication might not be able to handle all the data in a reasonable amount of time. We handled this issue by using RabbitMQ as message broker.[43] It supports the deployment of a cluster of message brokers increasing the possible throughput. Secondly, the evaluation storage may reduce the benchmarked system's performance by consuming the results at a much lower pace than the system is sending them. To avoid this situation, we chose RIAK—a key-value store which can be deployed as cluster.[44] This enables the consumption of several system results in parallel.

An important limitation of the platform is the necessary knowledge about several technologies and the platform APIs which is demanded. While viewing and searching for experiment results is straight forward, the deployment of a new benchmark or a new system can cause some effort for users, which have not worked with the platform

[43] `https://www.rabbitmq.com`; last accessed on 03.08.2022.
[44] `https://riak.com/products/riak-kv/`; last accessed on 03.08.2022.

before. Especially for complex benchmarks the workflow described in Section 3.4.5 may have to be adapted. We created base implementations for different benchmark and system components, developed example benchmarks and systems as open source projects, created video tutorials and enhanced the documentation of the platform over time incorporating user questions and feedback we received. However, the further lowering of this entry barrier remains an important future task.

Additionally, we received feature requests from the community. These requests are mainly targeting the user interface. However, one feature request focuses on the sharing of data. At the moment, it is not possible for containers executed inside the platform to share a common directory. Instead, data which has to be shared needs to be sent using the message queues. In the future, we want to make use of a feature of Docker containers which allows them to share a common data container without exposing the local hard drives of the servers to the 3rd party programs that are executed inside the containers of the benchmarks and the systems.

With respect to LEMMING, the main target is to improve its runtime to allow the generation of larger graphs. In addition to that, several extensions are planned. At the moment, the distribution of vertex degrees is always modeled as Poisson distribution. In future releases, LEMMING will learn individual distribution types for the different types of vertices. Thereafter, we plan to use LEMMING in various settings, e.g., to generate large graphs to evaluate the scalability of triple stores.

3.9 Conclusion

This chapter presents three major contributions to the benchmarking of Linked Data systems. First, we present an extension of GERBIL that enables a fair comparison of knowledge extraction systems that use different reference knowledge graphs. At the same time, our extensions reduces the influence of outdated IRIs that occur in many manually created gold standards.

Second, the architecture of the HOBBIT benchmarking platform is presented, which is based on real requirements collected from experts from across the world. The platform is designed to be modular and easy to scale up. HOBBIT is hence the first benchmarking platform that can be used for benchmarking Big Linked Data systems. The platform has already been used in several challenges and was shown to address the requirements of large-scale benchmarking for storage, predictive maintenance, knowledge acquisition, and question answering. These challenges showed clearly that HOBBIT can be used to measure both the scalability and accuracy

of Big Data platforms. As the platform is not limited to a particular step of the Linked Data life cycle and can be configured to use virtually any data generator and task generator, it is well suited for benchmarking any step of the Big Linked Data life cycle. A fully fledged implementation of the platform is available as an open-source solution and has started to attract the developer community. It will also serves as one of the key stones of the Innovative Training Network (ITN) KnowGraphs during the next years.[45] We hence aim to extend it so as to build the reference point for benchmarking Big Linked Data applications.

Third, we present LEMMING, a graph generator for creating graphs that mimic a given, real-world RDF dataset. We propose the usage of graph invariants and a refinement operator that is able to find these invariants based on a given set of graph metrics. Further, we propose six different approaches for the generation of a graph with a given size that abides to the determined graph invariants. Our evaluation shows that LEMMING is able to generate graphs that lead to similar benchmarking results as the real-world graph while a comparable baseline struggled to achieve this for all datasets. The contributions of this chapter will be used in other chapters of this thesis to evaluate Linked Data systems.

[45] `https://knowgraphs.eu/`; last accessed on 03.08.2022.

4 Crawling the Web of Data

The data on the Web are provided through means ranging from SPARQL endpoints over simple dump files to information embedded in Hyper Text Markup Language (HTML) pages. An indispensable step towards automating the usage of this data is the *automated and periodic gathering of information about available open data*. A necessary technical solution towards this end is a *scalable crawler for the Web of Data*. While the need for such a solution is already dire, it will become even more pressing to manage the growing amount of data that will be made available each year into the future. At present, the number of open-source crawlers for the Web of data that can be used for this task is rather small and all come with several limitations.

The efficiency and effectiveness of available open-source crawlers are typically evaluated by crawling the Web for a set amount of time while measuring different performance indicators such as the number of requests performed by the crawler [122, 133]. While this kind of experiment can be performed for a crawler at a given point in time, the experiments are virtually impossible to repeat and thus, their results are hard to compare with results of similar experiments. This is due to several factors, including primarily the fact that the Web is an ever-changing, evolving network of single, partly unreliable nodes. Another influence is the geographical location of the machine on which the crawler is executed. For example, geo-blocking can have an influence on the shape of the crawled network. Executing the same crawler on the same hardware might also lead to different evaluation results when various internet service providers offering different connections and bandwidths are used. In addition, the ground truth is not known in such experiments. Since the content of the complete Web is unknown, it is hard to measure the effectiveness of a crawler, i.e., its ability to retrieve relevant data. Hence, a *benchmark for Data Web crawlers* is needed.

We address both gaps within this chapter. First, we present SQUIRREL [236]—a distributed, open-source crawler for the Web of data.[1] SQUIRREL [236] supports

[¶] Parts of this chapter have been published as conference articles [236, 239]. The author of this thesis is also the main author of these articles. For both publications, the author developed the main ideas, designed, and implemented major parts of the solution, and wrote the majority of the publication.

[1] Our code is available at `https://github.com/dice-group/squirrel` and the documentation at `https://w3id.org/dice-research/squirrel/documentation`. (Last accessed on 04.08.2022.)

a wide range of RDF serializations, decompression algorithms, and formats of structured data. The crawler is designed to use Docker[2] containers to provide a simple build and run architecture [180]. SQUIRREL is built using a modular architecture and is based on the concept of dependency injection [101]. This allows for a further extension of the crawler and adaptation to different use cases.

Second, we propose ORCA [239]—a benchmark for Web Data Crawlers. The basic idea of ORCA is to alleviate the limitations of current benchmarking approaches by 1) generating a synthetic Data Web and 2) comparing the performance of crawlers within this controlled environment. The generation of the synthetic Web is based on statistics gathered from a sample of the real Data Web. The deterministic generation process implemented by our approach ensures that crawlers are benchmarked in a repeatable and comparable way.

Throughout the rest of this chapter, we model a crawler as a program that is able to

1. Download Web resources,
2. Extract information from these resources, and
3. Identify the addresses of other Web resources within the extracted information.

It will use these (potentially previously unknown) addresses to start with step 1 again in an autonomous way. A Data Web crawler is a crawler that extracts RDF triples from Web resources. Note that this definition excludes programs like the LOD Laundromat [30], which download and parse a given list of Web resources without performing the third step.

This chapter has the following main contributions.

1. We present Squirrel—a distributed, open-source crawler for the Web of data.
2. We provide an approach to generate a synthetic Data Web.
3. Based on this generator, we present ORCA—the first extensible FAIR benchmark for Data Web crawlers, which can measure the efficiency and effectiveness of crawlers in a comparable and repeatable way.
4. We present the first direct comparison of SQUIRREL and a state-of-the-art Data Web crawler in a repeatable setup.
5. We show how ORCA can be used to evaluate the politeness of a crawler, i.e., whether it abides by the Robots Exclusion Protocol [151].
6. We evaluate Data Web crawlers on synthetic graphs that mimic real-world graphs by combining ORCA and LEMMING, which we presented in Section 3.5.

[2] `https://www.docker.com/`; last accessed on 04.08.2022.

This chapter is organised as follows. Section 4.1 presents related work while Section 4.2 describes the developed crawler. In Section 4.3, the benchmark for Data Web crawlers is described before it is used in Section 4.4 to evaluate the Data Web crawlers. Section 4.5 discusses the evaluation results as well as limitations of the presented work. Section 4.6 describes two applications of our crawler before 4.7 concludes the chapter.

4.1 Related work

We separate our overview of the related work into two parts. First, we present publications related to crawlers and their evaluations with a focus on Data Web crawlers. Second, we present a brief overview of related work, with statistics regarding the Semantic Web.

4.1.1 Crawlers and their Evaluation

The Mercator Web Crawler [122] is an example of a general Web crawler. The authors describe the major components of a scalable Web crawler and discuss design alternatives. The evaluation of the crawler includes an 8-day run, which was compared to similar runs of the Google and Internet Archive crawlers. As performance metrics, the number of HTTP requests performed in a certain time period, and the download rate (in both documents per second and bytes per second) are used. Additionally, further analysis is undertaken regarding the received HTTP status codes, different content types of the downloaded data, and in which parts of the crawler the most CPU cycles are spent. This publication is an example of a classical crawler evaluation, which comes with the drawbacks explained in the previous Section. Srinivasan et al. [262] present an evaluation framework for comparing topical crawlers, i.e., crawlers that are searching for Web pages of a certain topic. It relies on a given topic hierarchy and the real Web, which makes it susceptible for the aforementioned drawbacks. The BUbiNG crawler [45] was evaluated relying on the real Web as well as a simulation. This simulation was carried out by using a proxy that generated synthetic HTML pages. However, the authors do not give further details about the structure of the simulated Web.

There is only a small number of open-source Data Web crawlers available that can be used to crawl RDF datasets. An open-source Linked Data crawler to crawl data from

the Web is LDSpider[3] [133]. It can make use of several threads in parallel to improve the crawling speed, and offers two crawling strategies. The breadth-first strategy follows a classical breadth-first search approach for which the maximum distance to the seed IRIs can be defined as termination criteria. The load-balancing strategy tries to crawl IRIs in parallel without overloading the servers hosting the data. The crawler implements a static politeness strategy and offers the configuration of the delay that is inserted between two consecutive requests. The crawled data can be stored either in files or can be sent to a SPARQL endpoint. It supports a limited amount of RDF serializations (details can be found in Table 4.1 in Section 4.3.2). In addition, it cannot be deployed in a distributed environment. Another limitation of LDSpider is the missing functionality to crawl SPARQL endpoints and open data portals. A detailed comparison of LDSpider and SQUIRREL can be found in Sections 4.3.2 and 4.4.

A crawler focusing on structured data is presented by Harth et al. [118]. It comprises a 5-step pipeline and converts structured data formats like XHTML or RSS into RDF. The evaluation is based on experiments in which the authors crawl 100 thousand randomly selected IRIs. To the best of our knowledge, the crawler is not available as open source project.

Hogan et al. [127, 128] use a distributed crawler to index resources for the Semantic Web Search Engine. In the evaluation, different configurations of the crawler—different numbers of threads as well as machines on which the crawler has been deployed—are compared, based on the time the crawler needs to crawl a given amount of seed IRIs. To the best of our knowledge, the crawler is not an open-source project.

Beek et al. [30] present the LOD Laundromat—an approach to download, parse, clean, analyze, and republish RDF datasets. The tool relies on a given list of seed URLs and comes with a robust parsing algorithm for various RDF serializations. Fernández et al. [95] use the LOD Laundromat to provide a dump file comprising 650 thousand datasets and more than 28 billion triples.

A Web crawler extended for processing RDF data is the open-source crawler Apache Nutch [146].[4] Table 4.1 in Section 4.3.2 shows the RDF serializations, compressions, and forms of structured data that are supported by the Apache Nutch plugin.[5]

[3] `https://github.com/ldspider/ldspider`; last accessed on 04.08.2022.
[4] `http://nutch.apache.org/`; last accessed on 04.08.2022.
[5] The information has been gathered by an analysis of the plugin's source code.

However, the plugin stems from 2007, relies on an outdated crawler version and failed to work during our tests.[6]

Overall, the open-source crawlers currently available are either not able to process RDF data, are limited in the types of data formats they can process, or are restricted in their scalability.

4.1.2 The Data Web

There are several publications analyzing the Web of data that are relevant for our work, since we use their insights to generate a synthetic Data Web. The Linked Open Data (LOD) Cloud diagram project periodically generates diagrams representing the LOD Cloud and has grown from 12 datasets in 2007 to more than 1200 datasets in 2020 [177]. These datasets are entered manually, require a minimum size, and must be connected to at least one other dataset in the diagram.

Other approaches for analyzing the Data Web are based on the automatic gathering of datasets. LODStats [22, 90] collects statistical data about more than 9 000 RDF datasets gathered from a dataset catalog.[7] In a similar way, Schmachtenberg et al. [246] use the LDSpider crawler [133] to crawl datasets in the Web, starting from a list of 560 000 seed IRIs. These IRIs are gathered from the datahub.io dataset catalog, the billion triple challenge [117], and dataset advertisements on the `public-lod@w3.org` mailing list. Overall, the authors collect more than 900 000 documents describing 8 million resources. These are grouped to 1014 datasets. Only 77 datasets are not crawled, since it is forbidden by the server's `robots.txt` file.[8] When analysing the crawled datasets, Schmachtenberg et al. find that most datasets are only sparsely linked. One large, weakly linked connected component of the graph comprises 71.99% of all datasets. 44% of all datasets have no outgoing links to other datasets and thus have only incoming links, or are completely isolated. At the same time a small number of datasets are highly linked.

Hogan et al. [130] gather and analyze 3.985 million open RDF documents from 778 different domains regarding their conformity to Linked Data best practices. On average, 70.3% of the IRIs within a dataset are dereferenceable with a high standard deviation, i.e., for some datasets, none of its IRIs offer this feature. From the dereferenceable IRIs, 83.6% return triples with local outgoing links, i.e., triples

[6] A brief description of the plugin and its source code can be found at `https://issues.apache.org/jira/browse/NUTCH-460`; last accessed on 04.08.2022.

[7] The dataset catalog is `http://thedatahub.org`.

[8] Such a file can be hosted on a Website to allow or disallow the crawling of certain pages [151]. `https://www.robotstxt.org/`

that have the queried IRI as subject. Only 55% offer back links, i.e., triples that have the queried resource as object. Additionally, the authors find that on average a dataset is linked to 20.4 other datasets. However, it has to be stated that highly linked datasets are preferred during the analysis.

Paulheim et al. [215] compare different methods to identify SPARQL endpoints. They compare the best practice proposed in the Vocabulary of Interlinked Datasets (VoID) specification [11]—i.e., to use the `/.well-known/void` path on a Web server to provide an RDF file with VoID information about datasets hosted on the server—with the usage of a dataset catalog like datahub. Their results show that the proposed usage of a special path for VoID information is not adopted on a large scale. A much larger amount of SPARQL endpoints can be found using dataset catalogs. Schmachtenberg et al. [246] confirm this finding by pointing out that only 14.69% of the crawled datasets provide VoID metadata.

Another application of Semantic Web technologies is the usage of semantic information embedded in Web pages. Bizer et al. [40] analyze 3 billion HTML pages to determine the adoption of the technologies like RDFa [8], Microdata [144], and Microformats [5]. 7.3 billion quads have been extracted from 369 million different pages containing structured data. The JSON-LD serialization [261], which can be used for the same task, is not taken into account by the authors.

4.2 Web of Data Crawler

This section presents our Web of Data crawler SQUIRREL. First, the requirements for the crawler are discussed. After that, an overview of the crawler is given before the two main components—Frontier and Worker—are described in detail.

4.2.1 Requirements

The requirements for our Web of Data crawler were gathered from nine organisations within the scope of the projects LIMBO and OPAL [302, 303].OPAL aimed to create an open data portal by integrating the available open data of different national and international data sources.[9] The goal of LIMBO was to collect available mobility

[9] See `http://web.archive.org/web/20220309232608/http://projekt-opal.de/en/welcome-project-opal/` and `https://www.bmvi.de/SharedDocs/DE/Artikel/DG/mfund-projekte/open-data-portal-germany-opal.html`; last accessed on 04.08.2022.

data of the ministry of transport, link them to open knowledge graphs, and publish them within a data portal.[10]

To deliver a robust, distributed, scalable, and extensible data Web crawler, we pursue the following goals with SQUIRREL:

CR1: The crawler should be designed to provide a distributed and scalable solution on crawling structured and semi-structured data [303].

CR2: The crawler must exhibit "respectful" behaviour when fetching data from servers by following the Robots Exclusion Standard Protocol [151] and using delays between requests [1, 2, 303]. This reduces the chance that the crawler is blocked by a server because of misbehavior, i.e., because the crawler 1) tried to access forbidding resources or 2) sent too many requests in a short amount of time.

CR3: Since not all data is available as structured data, crawlers for the data Web should offer a way to gather semi-structured data [303].

CR4: The project should offer easy addition of further functionality (e.g., novel serializations, other types of data, etc.) through a fully extensible architecture [303].

CR5: The crawler should provide metadata about the crawling process, allowing users to get insights from the crawled data [303].

In the following, we give an overview of the crawler's components, before describing them in more detail.

4.2.2 Overview

SQUIRREL comprises two main components: *Frontier* and *Worker* (**CR1**). To achieve a fully extensible architecture, both components rely on the dependency injection pattern [101], i.e., they comprise several modules that implement the single functionalities of the components. These modules can be injected into the components, facilitating the addition of more functionality (**CR4**). To support the addition of the dependency injection, SQUIRREL is based on the Spring framework.[11] Figure 4.1 illustrates the architecture of SQUIRREL.

When executed, the crawler has exactly one Frontier and a number of Workers, which can operate in parallel (**CR1**). The Frontier is initialized with a list of seed

[10] See `https://www.limbo-project.org/` and `https://www.bmvi.de/SharedDocs/DE/Artikel/DG/mfund-projekte/linked-data-services-for-mobility-limbo.html`; last accessed on 04.08.2022.

[11] `https://spring.io/`; last accessed on 04.08.2022.

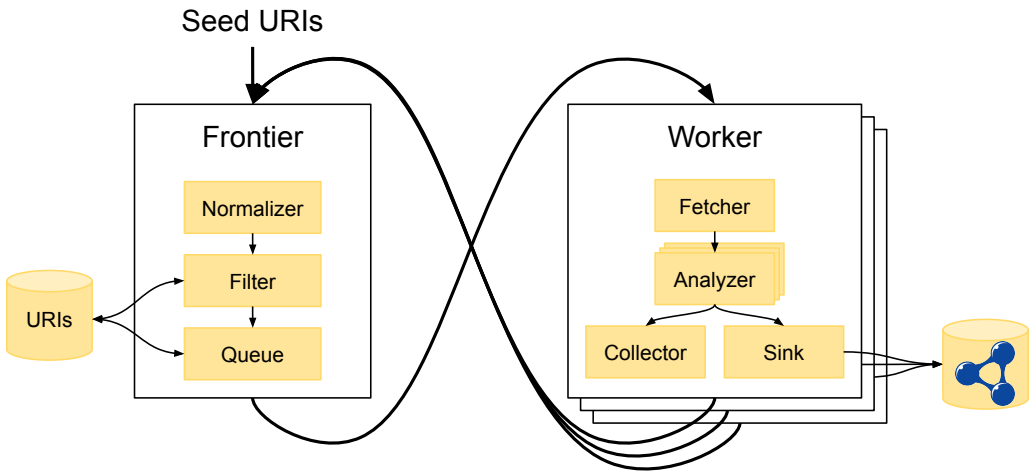


Figure 4.1.: SQUIRREL Core Achitecture.

IRIs. It normalizes and filters the IRIs, which includes a check of whether the IRIs have been seen before. Thereafter, the IRIs are added to an internal queue. Once the Frontier receives a request from a Worker, it gives a set of IRIs to the Worker. For each given IRI, the Worker fetches the IRI's content, analyzes the received data, collects new IRIs, and forwards the data to its sink. When the Worker processed all IRIs in the given set, it sends the set back to the Frontier together with the newly identified IRIs. The crawler implements the means for a periodic re-evaluation of IRIs known to have been crawled in past iterations.

4.2.3 Frontier

The Frontier has the task of organising the crawling. It keeps track of the IRIs to be crawled, and those that have already been crawled. It comprises three main modules:

1. A normalizer that preprocesses incoming IRIs,
2. A filter that removes already seen IRIs, and
3. A queue used to keep track of the IRIs to be crawled in the future.

Normalizer

The Frontier has to be able to identify IRIs that have already been seen before. To this end, it is necessary to be able to compare IRIs with respect to their equality. Two IRIs could be defined to be equal if they refer to the same resource [86]. However,

this is impractical as the resource would have to be requested by the crawler to decide whether the IRIs are the same. RFC 3987 [86] suggests a rule-based approach to transform a given IRI into a base form.[12] If two IRIs have the same base form, they refer to the same resource. Although this approach cannot fully eliminate false negatives [86] it has three advantages:[13]

1. False positives, i.e., the classification of two IRIs as equal although they refer to different resources, can be excluded [86].
2. The approach can be implemented without requesting and comparing the resources to which the IRIs refer.
3. The comparison of two IRIs that have been reduced to their base form becomes a simple string comparison.

The normalizer module transforms incoming IRIs into their base form. The IRI normalization comprises the following actions:

- Removal of default ports, e.g., port 80 for HTTP [36, 86].
- Removal of percentage-encoding for unreserved characters [36, 86].
- Normalization of the IRI path, e.g., by removing punctuations [36, 86].
- Removal of the IRIs' fragment part [36, 86].
- Alphanumeric sorting of key-value pairs for the IRIs' query parts to ease the string-based comparison.

In addition, the normalizer can be configured to remove session identifiers or similar parts of the IRI that have no influence on the retrieved content. The strings that mark such a part of the IRI have to be defined within the configuration.

Filter

The filter module is mainly responsible for filtering IRIs that have already been processed. To achieve this goal, the module stores all IRIs that have been processed in a persistent way. This ensures that SQUIRREL can be interrupted and restarted later on. Additionally, other filters can be added to narrow the search space of the crawler if necessary, e.g., black or white list filters.

[12] RFC 3986 [36] defines the term "base URI" when defining the handling of relative URIs. It should be noted that this term is not related to the term "base form" as we define it in this work.

[13] A false negative in this scenario are two IRIs that are considered to be not equal although they refer to the same resource.

Queue

The queue module stores the IRIs that should be crawled. It groups and sorts the IRIs, which makes it the main module for implementing crawling strategies. At present, SQUIRREL offers two queue implementations—an IP- and a domain-based first-in-first-out (FIFO) queue. Both work in a similar way by grouping IRIs based on their IP address or their pay-level domain, respectively. The definition of the queue's sorting is a major part of the crawling strategy. The current default implementation sorts the IRI groups according to the FIFO principle. When a Worker requests a new set of IRIs, the next available group is retrieved from the queue and sent to the Worker. Internally, this group is marked as blocked, i.e., it remains in the queue and new IRIs can be added by the Frontier. However, it cannot be sent to a different Worker. As soon as the Worker returns the requested IRIs, the group is unblocked and the crawled IRIs are removed from it. If the group is empty, it is removed from the queue. This implements a load-balancing strategy that aims to crawl the Web efficiently without overloading single IP addresses or pay-level domains.

Like the filter module, the queue module relies on a persistent database. This enables a restart of the Frontier without a loss of its internal states.

4.2.4 Worker

The Worker component performs the crawling based on a given set of IRIs. Crawling a single IRI comprises the following four steps:

1. The IRI content is fetched,
2. The fetched content is analyzed,
3. New IRIs are collected, and
4. The content is stored in a sink.

The modules for these steps are described in the following.

Fetcher

The fetcher module takes a given IRI and downloads its content. Before accessing the given IRI, the crawler follows the Robots Exclusion Standard Protocol [151] and checks the server's `robots.txt` file (**CR2**). If the IRI's resource can be crawled, one of the available fetchers is used to access it. At present, SQUIRREL uses four different fetchers. Two general fetchers cover the HTTP and the FTP protocol, respectively.

Two additional fetchers are used for SPARQL endpoints and CKAN portals, respectively. However, other fetchers can be added by means of the extensible SQUIRREL API if necessary.[14]

The Worker tries to retrieve the content of the IRI by using the fetchers, in the order in which they were defined, until one of them is successful. The fetcher then stores the data on the disk and adds additional information (e.g., the file's MIME type) to the IRI's properties for later usage. Based on the MIME type, the Worker checks whether the file has a compressed or archive file format. In this case, the file is decompressed and extracted for further processing. In its current release, SQUIRREL supports the formats Gzip, Zip, Tar, 7z, and Bzip2.[15]

Analyzer

The task of the Analyzer module is to process the content of the fetched file and extract triples from it. The Worker has a set of Analyzers that are able to handle various types of files. Table 4.1 lists the supported RDF serializations, the compression formats, and the different ways SQUIRREL can extract data from HTML files. It compares the supported formats with the formats supported by Apache Nutch and LDSpider [133]. Each Analyzer offers an `isElegible` method that is called with an IRI and the IRI's properties to determine whether it is capable of analyzing the fetched data. The first Analyzer that returns true receives the file together with a Sink and a Collector, and starts to analyze the data.

The following Analyzers are available in the current implementation of SQUIRREL:

1. The RDF Analyzer handles RDF files and is mainly based on the Apache Jena project.[16] Thus, it supports the following formats: RDF/XML, N-Triples, N3, N-Quads, Turtle, TRIG, JSON-LD, and RDF/JSON.
2. The HDT Analyzer is able to process compressed RDF graphs that are available in the HDT file format [96].
3. The RDFa Analyzer processes HTML and XHTML Documents extracting RDFa data using the Semargl parser.[17]

[14] Details about implementing a new fetcher can be found at `https://dice-group.github.io/squirrel.github.io/tutorials/fetcher.html`; last accessed on 04.08.2022.

[15] Details regarding the compressions can be found at `https://www.gnu.org/software/gzip/`, `https://www.iana.org/assignments/media-types/application/zip`, `https://www.gnu.org/software/tar/manual/html_node/Standard.html`, `https://www.7-zip.org/7z.html`, and `http://sourceware.org/bzip2/`, respectively. (Last accessed on 04.08.2022)

[16] `https://jena.apache.org`; last accessed on 04.08.2022.

[17] `https://github.com/semarglproject/semargl`; last accessed on 04.08.2022.

Table 4.1.: Comparison of RDF serializations, compressions, methods to extract data from HTML and other methods to access data supported by Apache Nutch (including the RDF plugin), LDSpider and SQUIRREL. Additionally, the table lists our benchmark ORCA explained in 4.3. (✓) marks serializations in ORCA that are supported by the benchmark but never demanded from a benchmarked crawler. X marks serializations that are listed as processible by a crawler but were not working during our evaluation (Section 4.4).

		Apache Nutch	LDSpider	SQUIRREL	ORCA
RDF serializations	RDF/XML	✓	✓	✓	✓
	RDF/JSON	–	–	✓	(✓)
	Turtle	✓	✓	✓	✓
	N-Triples	✓	X	✓	✓
	N-Quads	–	✓	✓	(✓)
	Notation 3	✓	✓	✓	✓
	JSON-LD	–	✓	✓	(✓)
	TriG	–	–	✓	(✓)
	TriX	–	–	✓	(✓)
	HDT	–	–	✓	–
Compressions	ZIP	✓	–	✓	✓
	Gzip	✓	–	✓	✓
	bzip2	–	–	✓	✓
	7zip	–	–	✓	–
	Tar	–	–	✓	–
HTML	RDFa	✓	✓	✓	✓
	Microdata	✓	✓	✓	–
	Microformat	✓	✓	✓	–
	HTML (scraping)	–	–	✓	–
	SPARQL	–	–	✓	✓
	CKAN	–	–	✓	✓

4. The scraping Analyzer uses the Jsoup framework for parsing HTML pages and relies on user-defined rules to extract triples from the parsed page.[18] This enables the user to use SQUIRREL to gather not only structured but also semi-structured data from the Web (**CR3**).
5. The CKAN Analyzer is used for the JSON line files generated by the CKAN Fetcher when interacting with the API of a CKAN portal. The Analyzer trans-

[18] `https://jsoup.org/`; last accessed on 04.08.2022.

forms the information about datasets in the CKAN portal into RDF triples using the DCAT ontology [173].

6. The Any23-based Analyzer processes HTML pages, searching for Microdata or Microformat embedded within the page.[19]
7. In contrast to the other Fetchers, the SPARQL-based Fetcher directly performs an analysis of the retrieved triples.

New Analyzers can be implemented if the default API does not match the user's needs.[20]

Collector

The Collector module collects all IRIs from the RDF data. SQUIRREL offers an SQL-based collector that makes use of a database to store all collected IRIs. It ensures the scalability of this module for processing large data dumps. For testing purposes, a simple in-memory collector is provided. As soon as the Worker has finished crawling the given set of IRIs, it sends all collected IRIs to the Frontier and cleans up the collector.

Sink

The Sink has the task to store the crawled data. Currently, a user can choose from three different sinks that are implemented. First, a file-based sink is available. This sink stores given triples in files using the Turtle serialization for RDF [29]. These files can be further compressed using GZip. The second sink is an extension of the file-based sink and stores triples in the compressed HDT format [96]. It should be noted that both sinks separate the crawled data by creating one file for each IRI that is crawled. An additional file is used to store metadata from the crawling process. Both sinks have the disadvantage that each Worker has a local directory in which the data is stored. The third sink uses SPARQL update queries to insert the data in a SPARQL store. This store can be used by several Workers in parallel. For each crawled IRI, a graph is created. Additionally, a metadata graph is used to store the metadata generated by the Workers. New sinks can be added by making use of the extensible API.[21]

[19] `https://any23.apache.org/`; last accessed on 04.08.2022.

[20] Details about implementing a new Analyzer can be found at `https://dice-group.github.io/squirrel.github.io/tutorials/analyzer.html`; last accessed on 04.08.2022.

[21] Details about implementing a new sink can be found at `https://dice-group.github.io/squirrel.github.io/tutorials/sink.html`; last accessed on 04.08.2022.

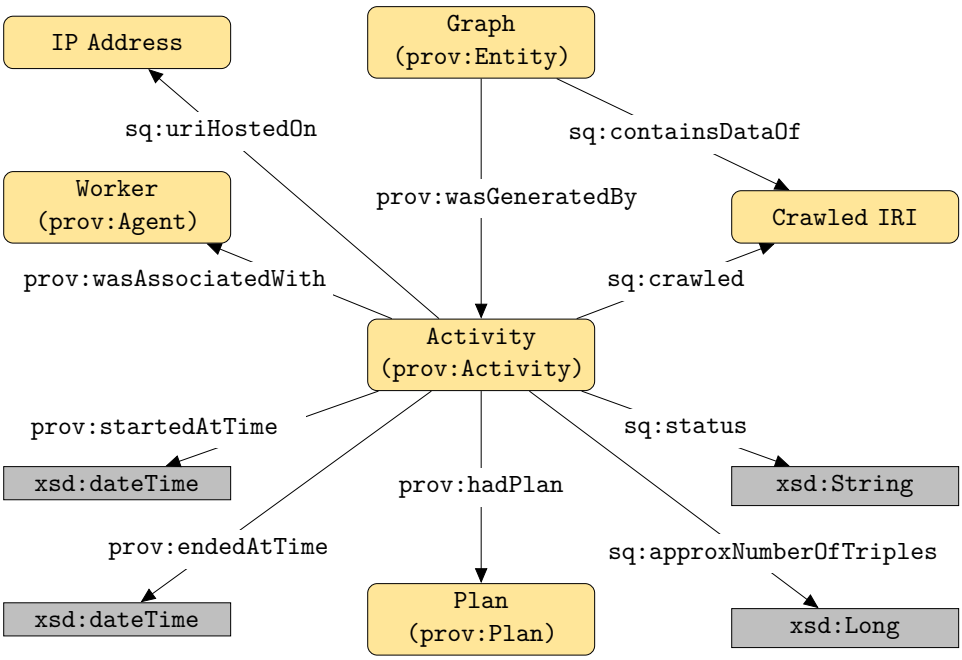


Figure 4.2.: The metadata stored by SQUIRREL for each `Activity` including the extensions of PROV-O.

Activity

The Workers of SQUIRREL document the crawling process by writing metadata to a metadata graph (**CR5**). This metadata mainly relies on PROV-O [161] and has been extended where necessary. Figure 4.2 gives an overview of the generated metadata. The crawling of a single IRI is modelled as an activity. Such an activity comes with data like the start and end time, the approximate number of triples received, and a status line indicating whether the crawling was successful. The result graph (or the result file in case of a file-based sink) is an entity generated by the activity. Both the result graph and the activity are connected to the IRI that has been crawled.

4.3 Crawling Benchmark

In this section, the benchmark for data Web crawlers is presented. First, preliminaries are introduced before the approach of the benchmark is presented. After that, the implementation details of the benchmark are described.

Table 4.2.: Different types of dataset gathered using URLs from LODStats.

	Dump file		SPARQL	HTML	Others
	Uncompressed	Compressed			
URLs	8536	260	593	10	660
Domains	332	138	354	9	402

4.3.1 Preliminaries

Crawlable Graph

Let $\mathbb{G} = (V, E)$ be a directed graph. $V = \{v_1, v_2, \ldots\}$ is the set of nodes of the graph. $E \subseteq V^2$ is the set of edges of $\mathbb{G}$. Given an edge $e = (v_i, v_j)$, we call v_i the source node and v_j the target node of e. Let $\mathfrak{S} \subseteq V$ be a set of seed nodes. We call a graph $\mathbb{G}$ *crawlable w.r.t.* $\mathfrak{S}$ iff all nodes $v \in V$ can be reached from nodes $v_k \in \mathfrak{S}$ in a finite number of steps by traversing the edges of the graph following their direction. It could follow that each directed graph can be made crawlable by defining $\mathfrak{S} = V$. However, following our definition of a Linked Data Crawler we are not interested in such a trivial case. A special case of crawlable graphs are graphs that are crawlable w.r.t. a singleton $\mathfrak{S} = \{v_\epsilon\}$. We call v_ϵ an *entrance node* of such graphs.

Data Web Analysis

The Data Web comprises servers of varying complexity. The types of nodes in this portion of the Web include simple file servers offering their data as dump files, Web servers able to dereference single RDF IRIs or to serve HTML Web pages with embedded structured data, and SPARQL endpoints that are able to handle complex queries.

We analyze the LODStats [22, 90] dump from 2016 to collect the statistics necessary for our benchmark by searching triples with the `dcat:downloadURL` property.[22] These triples show which URLs have been used to download the datasets. Based on the structure of the URL, we count the different types of sources. For example, if a URL ends on `.bz2` it is counted as a compressed dump file, while URLs ending with `/sparql` are counted as SPARQL endpoint.[23] Table 4.2 shows the counts of the single URLs and domains per node type. It can be seen that the majority of datasets

[22] The complete IRI of the property is `http://www.w3.org/ns/dcat#downloadURL`.

[23] The regular expressions used for that can be found at `https://github.com/dice-group/orca/blob/master/orca.tools/README.md`; last accessed on 04.08.2022.

are provided as an uncompressed dump file. 5.9% of the datasets are provided as SPARQL endpoint. From the 6.6% that are typed as Others, many are referring to single resources, e.g., FOAF profiles.[24] The servers hosting these single resources can be seen as dereferencing nodes. Note that these numbers only give a single view on the Data Web. Different approaches to access the Web might lead to completely different views. This explains the small number of IRIs that can be classified as HTML pages, since LODStats collects statistics about RDF datasets, excluding RDFa and other formats embedded in HTML.

Another important part of the Data Web are data catalogs. Although the related work does not mention them as part of the Data Web itself, they fulfill a crucial role by providing pointers to existing RDF datasets, which makes the Web much more connected. This can be seen in the fact that the list of RDF datasets analyzed by Auer et al. [22], Ermilov et al. [90], and Schmachtenberg et al. [246] relied at least partly on data from dataset catalogs. Another argument is given by Paulheim et al. [215], who show that using a data catalog to identify SPARQL endpoints works better than relying on the best practice proposed in the VoID specification [11]. Hence, we define the different types of nodes in the synthetic Data Web that is to be generated and used for the benchmark:

1. **Dump file node.** This node comprises an HTTP server offering RDF data as a single dump file. In its current implementation, ORCA randomly chooses one of the following RDF serializations: RDF/XML, Notation 3, N-Triples, or Turtle. Additionally, the file might be compressed with one of three available compression algorithms—ZIP, Gzip or bzip2.
2. **Dereferencing node.** This node comprises an HTTP server and answers requests to single RDF resources by sending all triples of its RDF graph that have the requested resource as subject. The server offers all serializations supported by Apache Jena. When a request is received, the serialization is chosen based on the HTTP `Accept` header sent by the crawler. The complete list of serializations supported by ORCA can be seen in Table 4.1.
3. **SPARQL endpoint.** This node offers an API, which can be used to query the RDF data using SPARQL via HTTP.[25]
4. **RDFa.** This node offers HTML Web pages via HTTP. The Web pages contain structured data that is embedded in the HTML. We choose RDFa as an example format of such type of data. Adding other types of embedded structured data (Microdata, Microformats, and JSON-LD) is left as future work. In its current

[24] FOAF is the 'Friend of a Friend' vocabulary. `https://web.archive.org/web/20220701160413/https://xmlns.com/foaf/spec/`; last accessed on 04.08.2022.

[25] In its current implementation, ORCA uses Virtuoso instances for this type of node.

version, the RDFa node relies on RDFa 1.0 and RDFa 1.1 test cases for HTML and XHTML of an existing RDFa test suite.[26]

5. **CKAN.** CKAN is a dataset catalog containing meta data about datasets.[27] It offers human-readable HTML pages and an API that can be used to query the catalog content.

Robots Exclusion Protocol

The Robots Exclusion Protocol allows the definition of rules for bots like crawlers [151]. The draft of the standard defines two rules—`allow` and `disallow`. They allow or disallow access to a certain path on a domain, respectively. The rules are defined in a `robots.txt` file, which is typically hosted directly under the domain in which the rules have been defined. Although additional rules are not covered by the standard, the standard allows the addition of lines. Some domain owners and crawlers make use of a `Crawl-delay` instruction to define how much delay a crawler should have between its requests to this single Web server.[28]

4.3.2 Approach

The main idea behind ORCA is to ensure the comparable evaluation of crawlers by creating a local, synthetic Data Web. The benchmarked crawler is initialized with a set of seed nodes of this synthetic cloud and asked to crawl the complete cloud. Since the cloud is generated, the benchmark knows exactly which triples are expected to be crawled and can measure the completeness of the crawl and the speed of the crawler. Since the cloud generation is deterministic, a previously used cloud can be recreated for benchmarking another crawler, ensuring that evaluation results are comparable if the experiments are executed on the same hardware. In the following, we describe the cloud generation in detail. An overview of the implementation and its details is given in Section 4.3.3.

Since the synthetically generated Data Web will be used to benchmark a Data Web crawler, we generate it as a crawlable graph w.r.t. a set of seed nodes $\mathfrak{S}$ as defined in Section 4.3.1. The generation of the synthetic Web can be separated into three steps:

1. Generating the single nodes of the cloud (V_ν),

[26] `http://rdfa.info/test-suite/`; last accessed on 04.08.2022.
[27] `https://ckan.org/`; last accessed on 04.08.2022.
[28] Examples are Bing [2] and Yandex [1].

Table 4.3.: Connectivity matrix $\mathcal{K}$ used for the experiments.

from \ to	Deref.	Dump file	SPARQL	CKAN	RDFa
Deref.	1	1	1	1	1
Dump file	1	1	1	1	1
SPARQL	1	1	1	1	1
CKAN	0	1	1	1	1
RDFa	1	1	1	1	1

2. Generating the node graph, i.e., the edges between the nodes, and
3. Generating the RDF data contained in the single nodes.

Node Generation

The set of nodes V_ν is generated by virtue of types selected from the list of available types in Section 4.3.1. The number of nodes in the synthetic Web (ν) and the distribution of node types are user-defined parameters of the benchmark. The node generation process makes sure that at least one node is created for each type with an amount > 0 in the configuration. Formally, let $\Upsilon = \{\mathrm{u}_1, \mathrm{u}_2, \ldots\}$ be the set of node types and $\Upsilon_\mathrm{g} \subseteq \Upsilon$ be the set of node types to be generated. To ensure that every type occurs at least once, the generation of the first $|\Upsilon_\mathrm{g}|$ nodes of V_ν is deterministic and ensures every type in Υ_g is generated. The remaining types are assigned using a seeded random model based on the user-defined distribution until $|V_\nu| = \nu$.

Node Graph Generation

In the real-world Data Web, connections (i.e., edges) between instances of certain node types are unlikely. For example, an open data portal is very likely to point to dump files, SPARQL endpoints or even other open data portals. However, it is very unlikely that it points to a single RDF resource, i.e., to a server which dereferences the IRI of the resource. To model this distribution, we introduce a connectivity matrix. Let $\mathcal{K}$ be a $|\Upsilon| \times |\Upsilon|$ matrix. $\mathfrak{k}_{ij} = 1$ means that edges from nodes of type u_i to nodes of type u_j are allowed. Otherwise, $\mathfrak{k}_{ij} = 0$. An example of such a connectivity matrix is given in Table 4.3 and will be used throughout this chapter. For the node types used in the current implementation of ORCA, all connections are allowed, except the example mentioned above.

The algorithm that generates the node graph takes the matrix $\mathcal{K}$, the previously created list of typed nodes, and the user-configured average node degree as input.

It starts with the first $|\Upsilon_g|$ nodes of V_ν and creates connections between them. For these initial nodes, all connections allowed in $\mathcal{K}$ are created. This initial graph is extended step-wise by adding the other nodes from V_ν. In each step, the next node from the list is added to the graph. The outgoing edges of the new node are added using a weighted sampling over the nodes that are permissible from the new node according to $\mathcal{K}$. Since the Web is known to be a scale-free network, the weights are the in-degrees of the nodes following the Barabási-Albert model for scale-free networks [9]. In the same way, a similar number of connections to the new node are generated.

Algorithm 4.1: Generation of the set of seeds $\mathfrak{S}$

Input : V, E
Output : $\mathfrak{S}$

```
1  S, V_m ← {}
2  for v_i ∈ V do
3      if inDegree(v_i) == 0 then
4          S ← S ∪ {v_i}
5          V_m ← markNodes(v_i, V_m, E)
6  V_u ← V \ V_m
7  while |V_u| > 0 do
8      v_i ← pop(V_u)
9      S ← S ∪ {v_i}
10     V_m ← markNodes(v_i, V_m, E)
11     V_u ← V_u \ V_m
```

After generating the node graph, a set of seed nodes $\mathfrak{S}$ has to be generated to make the graph crawlable as described in Section 4.3.1. This search is equivalent to the set cover problem [142]. Hence, searching for the smallest set of seed nodes would be NP-hard. Thus, we use a greedy solution (see Algorithm 4.1), which takes V and E of the generated node graph as input. We start by defining all nodes as unmarked nodes and the set of marked nodes V_m as empty (line 1). First, the algorithm searches for all nodes that have no incoming edge (line 3) since these nodes have to be seed nodes. For each of this node, the method `markNodes` is called. This method is shown in Algorithm 4.2 and implements a breadth-first search which starts at the given node and adds all connected nodes to the set of marked nodes. When all of the nodes without incoming edges have been processed, the number of unmarked nodes is checked. As long as unmarked nodes are left, the first unmarked node is added to $\mathfrak{S}$ and used to update the set of marked nodes (lines 7 – 11).[29] The

[29] The `pop` method returns and removes the first element from the given set.

algorithm terminates when all nodes are marked, i.e., reachable from the generated set of seed nodes $\mathfrak{S}$.

Algorithm 4.2: Breadth-first search to update the set of marked nodes V_m starting from the given node v.

Input : v, V_m, E
Output : V_m

```
1 V_Q ← {v}
2 V_m ← V_m ∪ {v}
3 while |V_Q| > 0 do
4     v_j ← pop(V_Q)
5     for e_i ∈ E do
6         if (source(e_i) == v_j) && (target(e_i) ∉ V_m) then
7             v_k ← target(e_i)
8             V_m ← V_m ∪ {v_k}
9             V_Q ← V_Q ∪ {v_k}
```

RDF Data Generation

The benchmark can work with any RDF data generator (see Section 3.1.2). An approach for mimicking real-world graphs is described in Section 3.5. However, ORCA comes with a simple generator that ensures the crawlability of the generated graph. The generator can be configured with three parameters:

1. The average number of triples per graph (τ),
2. The distribution of the sizes of the single graphs, and
3. The average degree of the RDF resources ($\bar{\mathrm{d}}$) in the graph.

In its current version, ORCA offers a simple approach that statically assigns the given average size to every RDF graph. However, this can be changed to use any other distribution. Let $\mathcal{G} = (V, T)$ be an RDF graph as defined in Definition 2.7. Further, let the resources of $\mathcal{G}$ comprise two sets $R_\mathcal{G} = R_i \cup R_e$. R_i is the set of internal IRI resources of the graph. These resources are defined in detail in $\mathcal{G}$ and belong to the graph's thematic domain. R_e is the set of external IRI resources. These resources are from a different thematic domain and are described in more detail in another graph. It can be followed that $R_i \cap R_e = \varnothing$. The generator focuses on creating nodes that are important for the crawling process. Consequently, it does not make use of any literals or blank nodes and the generated graph solely comprises IRI resources,

i.e., $V = R_{\mathcal{G}}$. It follows, that the triples T' of this specific graph can be defined as follows:

$$T' = \{(s, p, o) | s \in R_i \wedge p \in P \wedge o \in (R_i \cup R_e)\} \quad . \tag{4.1}$$

T' can be separated into two subsets $T' = T'_i \cup T'_e$. The set of graph-internal triples T'_i comprises triples with objects $o \in R_i$. In contrast, the set of outgoing triples T'_e (a.k.a. link set) contains only triples with external resources as objects ($o \in R_e$).

Like the node graph, each created RDF graph has to be crawlable w.r.t. a set of resources. For the RDF graphs, we implemented an algorithm based on the Barabási-Albert model for scale-free networks [9]. The implemented algorithm guarantees that all resources within the generated RDF graph can be reached from the first resource it generates. As defined in Section 4.3.1, this resource can be used later on as entrance node by all other RDF graph generators, which have to generate links to this graph.

Let τ be the RDF graph size that has been determined based on the chosen parameters. Based on the previously created node graph, the number of outgoing edges $\tau_e = |T'_e|$ as well as their objects, i.e., the set of external IRI resources R_e, are known. Algorithm 4.3 takes $\tau_i = \tau - \tau_e$ together with the average degree $\bar{\mathrm{d}}$ and a generated set of properties P as input to generate an initial version of graph $\mathcal{G}$. First, the first resource of the graph is created by generating its IRI and adding it to the set of internal nodes (lines 2 and 3). The loop (lines 4–13) adds new IRI resources to the graph until the number of necessary triples has been reached. For each new resource r_j, an IRI is generated (line 5) and an initial degree of the new resource d_j is drawn from a uniform distribution in the range $[1, 2\bar{\mathrm{d}}]$ (line 6). The d_j resources the newly created resource r_j will be connected to are sampled from the previously created resources based on their degree, i.e., the higher the degree of a resource, the higher the probability that it will be chosen for a new connection (line 6). The result of this step is the set R_j with $|R_j| = \mathrm{d}_j$.[30] For each of these resources, a direction of the newly added triple is chosen. Since the graph needs to be crawlable, the algorithm chooses the first triple to be pointing to the newly resourced node, i.e., the new resource is the object of the triple. This ensures that all resources can be reached, starting from the first resource of the graph. For every other triple, the decision is based on a Bernoulli distribution with a probability of $\frac{0.5\mathrm{d}_r - 1}{\mathrm{d}_r - 1}$ being a triple that has the new node as an object. This takes into account that the first triple is always added as incoming edge to the newly added node. Hence, the overall probability of an incoming edge, as well as for an outgoing edge, is 0.5 (line 9). Based on the

[30] In case there are less resources than the sampled d_j in the graph all available resources are used.

Algorithm 4.3: Initial RDF graph generation

Input : $\tau_i, P, \bar{\mathrm{d}}$
Output : $\mathcal{G}$

1 $T_i' \leftarrow \{\}$
2 $r_1 \leftarrow$ `generateResource(`0`)`
3 $R_i \leftarrow \{r_1\}$
4 **while** $|T_i'| < \tau_i$ **do**
5 $\quad r_j \leftarrow$ `generateResource(`$|R_i|$`)`
6 $\quad \mathrm{d}_j \leftarrow$`drawDegree(`$\bar{\mathrm{d}}$`)`
7 $\quad R_j \leftarrow$ `drawFromDegreeDist(`d_j, R_i, T_i'`)`
8 $\quad$ **for** $r_k \in R_j$ **do**
9 $\quad\quad$ **if** (`degree(`r_j`)`$== 0$)||(`bernoulli(`$\frac{0.5\mathrm{d}_j - 1}{\mathrm{d}_j - 1}$`)`) **then**
10 $\quad\quad\quad T_i' \leftarrow T_i' \cup \{$`generateTriple(`$r_k$,`draw(`$P$`)`,$r_j$`)`$\}$
11 $\quad\quad$ **else**
12 $\quad\quad\quad T_i' \leftarrow T_i' \cup \{$`generateTriple(`$r_j$,`draw(`$P$`)`,$r_k$`)`$\}$
13 $\quad R_i \leftarrow R_i \cup \{r_j\}$
14 $\mathcal{G} \leftarrow \{R_i, P, \varnothing, T_i'\}$

Table 4.4.: Templates of resource IRIs to refer to an external resource and its dependency on the external node type. H = host name; F = file format; N = resource ID.

Node type	IRI template
Dereferencing	http://H/dataset-0/resource-N
Dump file	http://H/dumpFile$F$#dataset-0-resource-N
RDFa	http://H/dataset-0/resource-0
SPARQL	http://H:8890/sparql
CKAN	http://H:5000/

chosen direction, the new triple is created with a property that is randomly drawn from the property set P (lines 10 and 12).

After the initial version of the RDF graph is generated, the outgoing edges of T_e' are created. For each link to another dataset, a triple is generated by drawing a node from the graph as subject, drawing a property from P as predicate and the given external node as object. Both—T_e' and R_e—are added to $\mathcal{G}$ to finish the RDF graph generation.

IRI Generation

Every resource of the generated RDF graphs needs to have an IRI. To make sure that a crawler can use the IRIs during the crawling process, the IRIs of the resources are generated depending on the type of node hosting the RDF dataset. The different IRI templates are available in Table 4.4. All IRIs contain the host name (marked with H in Table 4.4). At the moment, the dump file and the dereferencing node have only one single dataset. Therefore, the IRI templates of these node types contain the string "dataset-0". A numeric ID is attached (marked with N) to make each resource IRI unique. Additionally, the dump file node IRIs contain the file extension representing the format (marked with F). This comprises the RDF serialization and the compression (if a compression has been used). The RDFa node has only one generated HTML page that refers to the single RDFa tests. The IRIs of the single tests use the file structure of the used test suite. If a resource of the SPARQL node is used in another generated RDF graph (i.e., to create a link to the SPARQL node), the URL of the SPARQL API is used instead of a resource IRI. The resources that are stored within the SPARQL endpoint use the IRI template of the dereferencing node. In a similar way, the links to the CKAN nodes are created by pointing to the CKAN's Web interface without any additional information.

4.3.3 Implementation

Overview

ORCA is a benchmark built upon the HOBBIT benchmarking platform [237] described in Chapter 3.[31] This FAIR benchmarking platform allows Big Linked Data systems to be benchmarked in a distributed environment. It relies on the Docker container technology to encapsulate the single components of the benchmark and the benchmarked system.

We adapted the suggested design of a benchmark described in Section 3.4.5 to implement ORCA. The benchmark comprises a benchmark controller, data generators, an evaluation module, a triple store, and several nodes that form the synthetic Data Web. The benchmark controller is the central control unit of the benchmark. It is created by the HOBBIT platform, receives the configuration defined by the user and manages the other containers that are part of the benchmark. Figure 4.3 gives an overview of the benchmark components, the data flow and the single steps of the

[31] `https://github.com/hobbit-project/platform`; last accessed on 04.08.2022.

workflow. The workflow itself can be separated into 4 phases—creation, generation, crawling and evaluation.[32] When the benchmark is started, the benchmark controller creates the other containers of the benchmark.[33] During this creation phase, the benchmark controller chooses the types of nodes that will be part of the synthetic Data Web, based on the parameters configured by the user. The Docker images of the chosen node types are started together with an RDF data generator container for each node that will create the data for the node. Additionally, a node data generator, a triple store, and the evaluation store are started. The node data generator will generate the node graph. The triple store serves as a sink for the benchmarked Linked Data crawler during the crawling phase while the evaluation module will evaluate the crawled data during the evaluation phase.

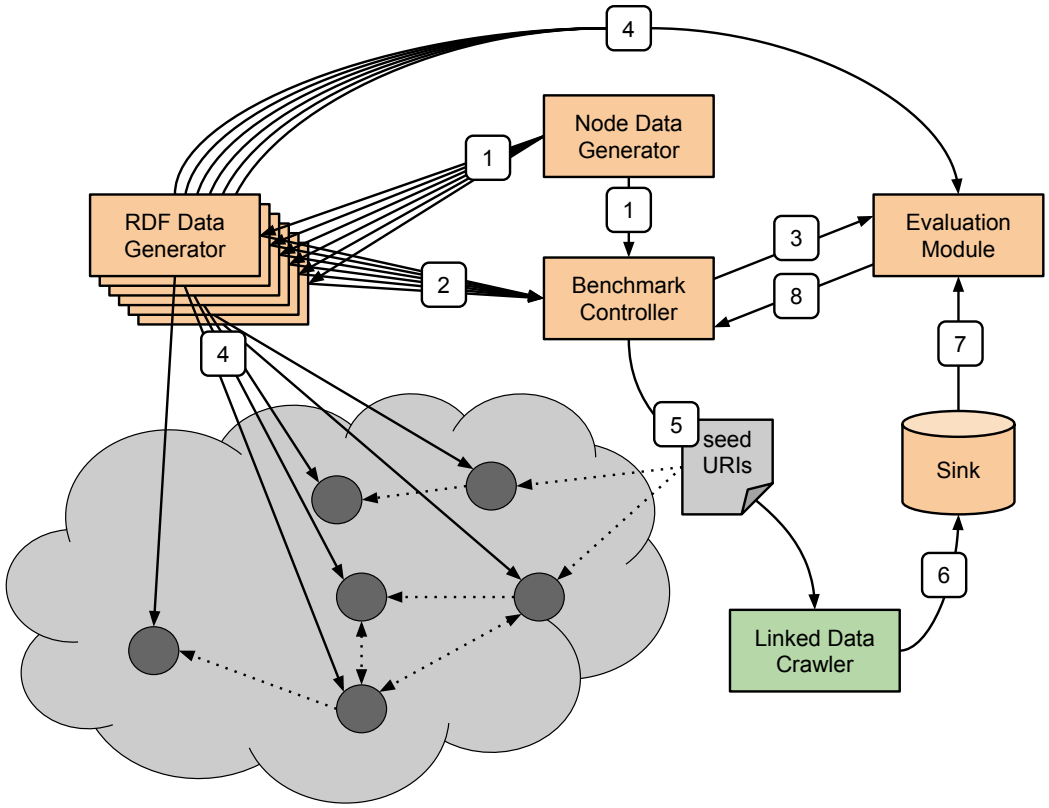


Figure 4.3.: Overview of the Benchmark components and the flow of data. Orange: Benchmark components; Grey: Synthetic Data Web generated by the benchmark; Dark blue: The benchmarked crawler; Solid arrows: Flow of data; Dotted arrows: Links between RDF datasets; Numbers indicate the order of steps.

[32]The phases can be mapped to the three phases described in Section 3.4.5. The creation and generation phases form the initialization phase while the crawling phase equates the benchmarking phase.

[33]The benchmarked crawler is created by the HOBBIT platform as described in Section 3.4.5.

After the initial creation, the graph generation phase is started. This phase can be separated into two steps—initial generation and linking. During the first step, each RDF data generator creates an RDF graph for its Web node. In most cases, we use the algorithm described in Section 9. For data portal and RDFa nodes, the generation process differs. The portal nodes solely use the information to which other nodes they have to be linked to, i.e., each linked node is inserted as a dataset into the data portal node's database. The RDFa node relies on an already existing test suite.[34] It generates an HTML page that refers to the single tests and to all other connected nodes using RDFa. The node data generator creates the node graph as described in Section 4.3.2. After this initial generation step, the node graph is sent to the benchmark controller and all RDF data generators (Step 1 in Figure 4.3). This provides the RDF data generators with the information to which other nodes their RDF graph should be linked. Subsequently, the RDF data generators send their metadata to each other and the benchmark controller (Step 2).[35] This provides the data generators with the necessary data to create links to the entrance nodes of other RDF datasets during the linking step. Additionally, the benchmark controller forwards the collected metadata to the evaluation module and the nodes in the cloud (Step 3).[36] At the end of the generation phase, the generated RDF graphs are forwarded to the single nodes and the evaluation module (Step 4). The generation phase ends as soon as all nodes have signalled to the benchmark controller that they have processed the received data.

After the generation phase is finished and the HOBBIT platform signals that the crawler has initialized itself, the benchmark controller submits the seed IRIs to the crawler (Step 5). This starts the crawling process in which the crawler must download RDF data from the nodes, process it to extract new, unseen IRIs, and forward the data to its sink (Step 6) before it crawls the collected, unseen IRIs. When the crawler finishes its crawling—i.e., all given IRIs and all IRIs found in the crawled RDF data have been crawled—the crawler terminates and the crawling phase ends.

During the evaluation phase, the evaluation module measures the recall of the crawler by checking whether the RDF graphs generated by the data generators can be found in the sink (Step 7). For smaller RDF graphs, all triples are checked. However, ORCA offers the option to sample triples from the generated RDF graphs to reduce the number of SPARQL queries and, hence, the runtime of the evaluation. The result of this evaluation is sent to the benchmark controller, which adds further

[34] `http://rdfa.info/test-suite/`; last accessed on 04.08.2022.

[35] The submissions from each data generator to each other data generator have been omitted in the figure to keep it clean.

[36] The submission to the cloud nodes has been omitted in the figure to keep it clean.

data and results of the benchmarking process (Step 8). This can include data that has been gathered from the single nodes of the cloud, e.g., access times. After this, the final results are forwarded to the HOBBIT platform.

Benchmark Features

The benchmark offers several additional features. The overview already shows that the design of the benchmark is kept scalable regarding the size of the synthetic Data Web. It is also expandable with respect to the types of nodes that can be part of the cloud. Additionally, the benchmark can be used to check single features of a Data Web crawler. As an example, the current implementation checks whether a crawler follows the rules defined in a Web server's `robots.txt` file. This check covers two features. First, the benchmark can check whether a crawler accesses an IRI even though it has been listed as disallowed [151]. If this feature is configured by the user, the nodes add additional triples to their graph. These triples refer to resources listed as disallowed in the node's `robots.txt` file. Second, the user can define a `Crawl-delay` value, which will be added to the file. Although the `robots.txt` standard [151] does not include this command, it is used by several crawlers to define how much delay a crawler should have between its request to this single Web server [1, 2]. During the crawling process, the nodes keep track of the delays between requests and whether the crawler requests one of the disallowed resources. Both are forwarded to the benchmark controller during the evaluation phase. Another feature of ORCA is the decoupling of the synthetic cloud from the Semantic Web. By using a DNS server within the benchmark, the crawler is prevented from crawling the "real" Web even if IRIs might refer to it (e.g., when using real-world data instead of generated data).

Parameters

The benchmark offers several parameters to adapt it to various scenarios.

- **Number of nodes** (ν)**:** The number of nodes in the synthetic graph.
- **Average node degree:** The average degree of the nodes in the generated graph.
- **RDF dataset size** (τ)**:** Average number of triples of the generated RDF graphs.
- **Average resource degree** ($\bar{\mathrm{d}}$)**:** The average degree of the resources in the RDF graphs.

- **Node type amounts:** For each node type, the user can define the proportion of nodes that should have this type.
- **Dump file serializations:** For each available dump file serialization, a boolean flag can be set.
- **Dump file compression ratio:** Proportion of dump files that are compressed.
- **Average ratio of disallowed resources:** Proportion of resources that are generated within a node and marked as disallowed for crawling.
- **Average crawl delay:** The crawl delay of the node's `robots.txt` file.
- **Seed:** A seed value for initializing random number generators is used to ensure the repeatability of experiments.

Key Performance Indicators

The benchmark measures the effectiveness and efficiency of Data Web crawlers. For this purpose, it provides the following KPIs.

- **Recall:** The recall of the crawler is calculated based on the triples of the generated RDF graphs. To this end, the evaluation module counts the true positives—i.e., the number of expected triples found in the crawler's sink. We define a crawler's recall as the number of true positives divided by the number of checked triples. Note that using recall does not punish a crawler for additional triples that have been added to the sink, e.g., provenance data of the crawled datasets. The benchmark offers the recall in three forms—1) per node, 2) as micro average, and 3) as macro average. Additionally, the benchmark reports the number of true positives as well as the number of checked triples.
- **Runtime:** the benchmark measures the time it takes from starting the crawling process to termination by sending the seed IRIs to the crawler. A shorter runtime indicates greater efficiency if the recall is the same.
- **Requested disallowed resources:** the number of forbidden resources crawled by the crawler, divided by the number of all resources forbidden by the `robots.txt` file. The number of disallowed resources that exist in the synthetic Data Web are also reported.
- **Crawl delay fulfilment:** this KPI is used to check whether a crawler respects the `Crawl-delay` instruction in the `robots.txt` file. We define this KPI as the average measured delay between the requests received by a single node divided by the delay defined in the `robots.txt` file. If the measure is below 1.0 the crawler does not strictly follow the delay instruction.
  The KPI is calculated per node. The single per-node values are summarized as average, minimum, and maximum.

- **Consumed hardware resources:** the evaluation module measures the RAM and CPU consumption of the benchmarked crawler. The measured values are available as diagram over time or as summary.[37]
- **Number of evaluated triples:** the exact number of triples that have been used to calculate the recall.
- **Number of disallowed resources:** the exact number of resources that have been marked as disallowed.
- **Triples over time:** the evaluation module keeps track of the number of triples in the sink over time and reports the numbers.
- **Cloud graph visualisation:** the benchmark generates a visualization of the generated synthetic Linked Data Web as depicted in Figure 4.4.[38] It is backed by a graph that contains meta data of the single nodes and per-node results.

4.4 Evaluation

For evaluating the approaches presented within this chapter, we use four experiments. We start by evaluating Data Web crawlers including SQUIRREL using our benchmark ORCA. The first experiment uses all node types available in ORCA to generate a realistic synthetic Data Web. This experiment mainly focuses on the recall of the benchmarked crawlers. The second experiment uses a simpler Web to measure the crawler's efficiency. The third experiment checks whether the crawlers abide by the Robots Exclusion Protocol [151]. The fourth experiment uses generated graphs of LEMMING within ORCA to benchmark data Web crawlers with RDF graphs that mimic real-world graphs.

For all experiments, the online instance of HOBBIT is used. It is deployed on a cluster with 3 servers that are solely used by the benchmark and 3 servers that are available for the system. Each of the servers has 16 cores with Hyperthreading and 256 GB RAM.[39]

[37] i.e., sum of CPU time, average RAM consumption, and maximum RAM consumption.

[38] The figure has been taken from a small example experiment that is available at `https://master.project-hobbit.eu/experiments/1617032707587`; last accessed on 05.08.2022.

[39] The details of the hardware setup that underlies the HOBBIT platform can be found at `https://hobbit-project.github.io/master#hardware-of-the-cluster`; last accessed on 05.08.2022.

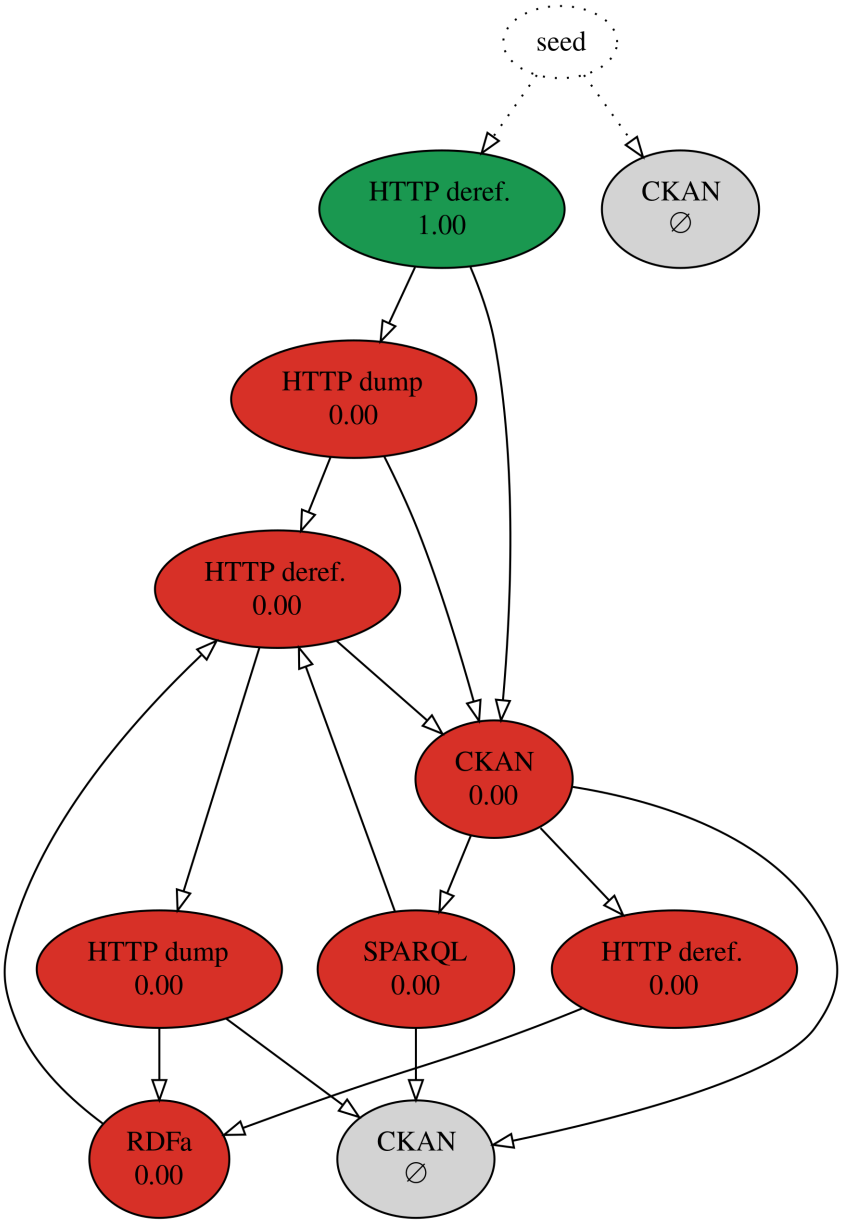


Figure 4.4.: Example cloud graph visualization of an experiment with LDspider. The dotted "seed" node points to the nodes in the graph that have been given as seed to the crawler. The remaining nodes are colored based on the recall the benchmarked crawler achieved. All expected triples of the green node have been crawled while the red nodes have a recall of 0.0. The two gray CKAN nodes do not have any outgoing link. Hence, they do not have any data that could be crawled.

4.4.1 Benchmarked Crawlers

To the best of our knowledge, SQUIRREL and LDSpider are the only working open-source Data Web crawlers available. Other crawlers are either not available as open-source project or Web crawlers without the ability to process RDF data.

For our experiments, we implemented an adapter for SQUIRREL. SQUIRREL (W1), (W3), (W9), and (W18) are instances of the crawler using 1, 3, 9 or 18 worker instances, respectively.[40]

LDSpider [133] can be configured to use different numbers of Threads and different crawling strategies. For our experiments, we dockerized LDSpider and implemented a system adapter to make it compatible with the HOBBIT platform. We created several LDSpider instances with different configurations. LDSpider (T1), (T8), (T16), and (T32) use the breadth-first strategy and 1, 8, 16 or 32 threads, respectively. During our first experiments, we encountered issues with LDSpiders' SPARQL client, which was not storing the crawled data in the provided triple store. To achieve a fair comparison of the crawlers, we extended our system adapter to implement our own SPARQL client, used LDSpider's file sink to get the output of the crawling process, and sent file triples to the benchmark sink. These instances of LDSpider are marked with the addition "FS". Additionally, we configured the LDSpider instance (T32,FS,LBS), which makes use of the load-balancing strategy to compare the two strategies offered by the crawler.

4.4.2 Data Web Crawling

The first experiment simulates a real-world Data Web and focuses on the effectiveness of the crawlers. To this end, we measure the amount of correct triples they retrieve by calculating the micro recall Re_{mic}. To define the distribution of node types, we analyze the download URLs of the LODStats [22, 90] dump from 2016. Based on this analysis, the generated cloud comprises 100 nodes with 40% dump file nodes, 30% SPARQL nodes, 21% dereferencing nodes and 5% CKAN nodes. 4% are RDFa nodes to represent the 1 billion RDFa triples identified by Bizer et al. [40] in comparison to the 28 billion RDF triples gathered from the Semantic Web by Fernández et al. [95]. Following Hogan et al. [130], the average degree of each node is set to 20. For each node, the RDF graph generation creates 1000 triples with an average degree of 9

[40] Since the HOBBIT cluster assigns 3 servers for the benchmarked crawler, we use multiples of 3 for the number of workers.

triples per resource.[41] Based on our LODStats analysis, 30% of the dump file nodes use one of the available compression algorithms for the dump file. The usage of `robots.txt` files is disabled.

Since LDSpider does not support the crawling of SPARQL endpoints, data catalogs like CKAN, or compressed dump files, we expect LDSpider to achieve a lower Recall than SQUIRREL. The results of the experiment are listed in Table 4.5.[42]

4.4.3 Efficiency Evaluation

The second experiment focuses on the efficiency of the crawler implementations. For this purpose, a synthetic Web comprising 200 dereferencing nodes is used since they offer to negotiate the RDF serialization for transferring the data. This ensures that all crawlers can crawl the complete Web. The other parameters remain as before.

For LDSpider, we use only the FS instances and configure its politeness to not insert any delay between two requests. We expect both crawlers to be able to crawl the complete cloud and that crawler instances with more threads or workers will crawl faster. The results of the experiment are listed in Table 4.5.[43]

4.4.4 Robots Exclusion Protocol Check

In the third experiment, we evaluate whether the crawlers follow the rules defined in a server's `robots.txt` file. To this end, we configure ORCA to generate a smaller Web comprising 25 dereferencing nodes. Each of the nodes copies 10% of its RDF resources and marks the copies disallowed for crawling using the `disallow` instruction in its `robots.txt` file. Additionally, we define a delay of 10 seconds between two consecutive requests using the `Crawl-delay` instruction in the same

[41] The dataset size is roughly the average document size from Hogan et al. [130] (after excluding an outlier). The average degree is derived from the statistics of DBpedia [21, 162], Freebase [46], OpenCyc [164], Wikidata [298], and Yago [174] from Färber et al. [92].

[42] The detailed results can be seen at `https://w3id.org/hobbit/experiments#1584544940477,1584544956511,1584544971478,1585403645660,1584545072279,1585230107697,1584962226404,1584962243223,1585574894994,1585574924888,1585532668155,1585574716469`; last accessed on 05.08.2022.

[43] While repeating the experiments, the measures turned out to be stable with standard deviations of ∼2% for the RAM, ∼5% for runtime, and CPU time. The detailed results can be found at `https://w3id.org/hobbit/experiments#1608306725630,1608306734009,1609758770866,1604685169129,1608052657254,1609763993514,1609779299724,1609792299538,1608038447931`; last accessed on 05.08.2022.

Table 4.5.: Results of the Data Web crawling and efficiency experiments.

Crawler	Data Web		Efficiency			
	Re_{mic}	**Runtime** (in s)	Re_{mic}	**Runtime** (in s)	**CPU** (in s)	**RAM** (in GB)
LDSpider (T8)	0.00	67	–	–	–	–
LDSpider (T16)	0.00	73	–	–	–	–
LDSpider (T32)	0.00	74	–	–	–	–
LDSpider (T1,FS)	0.31	1 798	1.00	1 847	627.2	1.8
LDSpider (T8,FS)	0.30	1 792	1.00	1 717	658.9	5.2
LDSpider (T16,FS)	0.31	1 858	1.00	1 753	1 677.1	1.6
LDSpider (T32,FS)	0.31	1 847	1.00	1 754	1 959.1	4.0
LDSpider (T32,FS,LBS)	0.03	66	0.01	56	–	–
SQUIRREL (W1)	0.98	6 663	1.00	12 051	1 096.7	3.5
SQUIRREL (W3)	0.98	2 686	1.00	4 096	992.0	7.7
SQUIRREL (W9)	0.98	1 412	1.00	1 500	652.0	16.6
SQUIRREL (W18)	0.97	1 551	1.00	893	424.0	24.0

Table 4.6.: Results for a Data Web with `robots.txt` files including disallow and crawl-delay rules. CDF = Crawl delay fulfilment; RDR = Requested disallowed resources.

Crawler	CDF			RDR	Runtime
	Min	Max	Avg		(in s)
LDSpider (T32,FS)	0.052	0.122	0.089	0.0	224
LDSpider (T32,FS,LBS)	0.002	0.007	0.004	0.0	43
SQUIRREL (W18)	0.697	0.704	0.699	0.0	2384

file. The average node degree of the nodes is configured as 5 while the average resource degree is set to 6. Table 4.6 shows the results of this experiment.[44]

4.4.5 Evaluation with Lemming Graphs

For the last experiment, we integrate graphs generated by LEMMING into the ORCA benchmark. To this end, we generate 10 graphs. We use the Semantic Web dog food dataset described in Section 3.6.3 as input for LEMMING and generate graphs with 45 398 vertices using the BCS-BIS mode and 50000 optimization steps. We use these graphs to generate a synthetic data Web. The ORCA benchmark is configured to use the pre-generated datasets and extend them with external triples as explained in Section 9. These triples are necessary to connect the 10 graphs with each other and create a crawlable Web. This Web comprises 4 dereferencing nodes and 6 dump file

[44]The detailed results can be seen at `https://w3id.org/hobbit/experiments#1575626666061, 1575592492658, 1575592510594`; last accessed on 05.08.2022.

Table 4.7.: Results of the fourth experiments measured on a synthetic Web comprising RDF graphs generated by LEMMING. Std. dev. = standard deviation.

Crawler	Re_{mic}	**Runtime** (in s)		**RAM** (in GB)	
		Average	Std. dev.	Average	Std. dev.
LDSpider (T1,FS)	1.0	10 295	95	2.7	0.1
LDSpider (T8,FS)	1.0	11 525	41	3.2	0.9
LDSpider (T16,FS)	1.0	12 177	67	3.9	1.0
LDSpider (T32,FS)	1.0	13 748	190	2.7	0.1
SQUIRREL (W1)	1.0	30 248	2 274	9.8	0.5
SQUIRREL (W3)	1.0	12 391	396	15.4	0.2
SQUIRREL (W9)	1.0	6 619	91	35.6	0.6
SQUIRREL (W18)	1.0	4 845	205	80.0	1.4

nodes. We executed each experiment three times and report the average results and the standard deviations in Table 4.7.[45]

4.5 Discussion

The experiment results give several insights. As expected, none of the instances of LDSpider were able to crawl the complete synthetic Linked Data Web during the first experiment. Apart from the expected reasons previously mentioned (i.e., the missing support for SPARQL, CKAN nodes and compressed dump files), we encountered two additional issues. First, as mentioned in Section 4.4.1, the SPARQL client of LDSpider did not store all the crawled triples in the provided triple store. This leads to the different recall values of the LDSpider instances with and without the "FS" extension. Second, although we tried several content handler modules and configurations, LDSpider did not crawl dump files provided as N-Triples. In comparison, the SQUIRREL instances crawl the complete cloud, except for some triples of RDFa and CKAN nodes.

The second experiment reveals that overall, LDSpider is more resource-efficient than SQUIRREL. In nearly all cases, LDSpider crawls the Web faster and uses less

[45] The detailed results can be found at `https://w3id.org/hobbit/experiments#1636536652973`, `1636708996546`, `1636998810648`, `https://w3id.org/hobbit/experiments#1636452031325`, `1636709034488`, `1636998838887`, `https://w3id.org/hobbit/experiments#1636452042712`, `1636709009010`, `1636998818625`, `https://w3id.org/hobbit/experiments#1636452058515`, `1636709024625`, `1636998827415`, `https://w3id.org/hobbit/experiments#1636566308274`, `1636709047446`, `1637163281570`, `https://w3id.org/hobbit/experiments#1636536685922`, `1636709056292`, `1637163292990`, `https://w3id.org/hobbit/experiments#1636649328186`, `1636998790922`, `1637163321762`, and `https://w3id.org/hobbit/experiments#1636659475867`, `1636998800768`, `1637252026055`; last accessed on 05.08.2022.

resources than the SQUIRREL instances. Only with 9 or more workers SQUIRREL is able to crawl faster. For the size of the graph, the number of threads used by LDSpider do not seem to play a major role when employing the breadth-first strategy. It could be assumed that the synthetic Web, with 200 nodes, provides only rare situations in which several nodes are crawled by LDSpider in parallel. However, this assumption can be refuted since SQUIRREL achieves lower runtimes. Therefore, the load-balancing strategy of SQUIRREL seems to allow faster crawling of the Web than the breadth-first strategy of LDSpider. However, the LDSpider (T32,FS,LBS) instance implementing a similar load-balancing strategy aborts the crawling process very early in all three experiments. Therefore, a clearer comparison of both strategies is not possible.

The third experiment shows that both crawlers follow the Robots Exclusion Protocol as both did not request disallowed resources. However, SQUIRREL seems to insert delays between it's requests following the `Crawl-delay` instruction—although it reaches on average only 69.9% of the delay the server asked for—while LDSpider does not take this instruction into account and solely relies on its static politeness strategy with a configurable default delay of 500ms.

The fourth experiments shows that both benchmarked crawlers are able to handle larger datasets that are close to real-world datasets. In addition to that, it confirms the findings of the second experiment. Again, SQUIRREL instances with a large amount of workers are faster than the LDSpider instances. However, they consume much more memory (RAM). In addition, the results suggest that although the experiments are complex and take several hours, the measured values are stable across several runs leading to comparably low standard deviations. This finding underlines that ORCA supports the reproducibility of experiment results.

4.6 Application

The knowledge graphs available on the Web have been growing over recent years both in number and size [95, 177]. This development has been accelerated by governments publishing public sector data on the Web [90].[46] SQUIRREL is used within several research projects, of which two are of national importance in Germany and are both tackling the increased amount of data provided in the public sector. The OPAL project created an integrated portal for open data by integrating datasets from

[46]Examples include the European Union at `https://ec.europa.eu/digital-single-market/en/open-data` and the German Federal Ministry of Transport and Digital Infrastructure with data at `https://mobilithek.info/`; last accessed on 31.07.2022.

Table 4.8.: Crawling statistics of the OPAL project.

	Datasets	Triples	Run time	Type
mCLOUD.de	1 394	19 038	25min	HTML
govdata.de	34 057	138 669	4h	CKAN
europeandataportal.eu	1 008 379	13 404 005	36h	SPARQL
OpenDataMonitor.eu	104 361	464 961	7h	CKAN

several data sources from all over Europe.[47] The project focused on public sector data and mainly on the portals mCLOUD.de, govdata.de, and europeandataportal.eu. In addition, several sources found on OpenDataMonitor.eu were integrated. SQUIRREL was used to regularly gather information about available datasets from these portals. Table 4.8 lists the number of datasets that were extracted from the portals, the time the crawler needed to gather them, and the way the crawler accessed data. It should be noted that the run times include the delays SQUIRREL inserts between single requests to ensure that the single portals are not stressed. The portals evidently used different ways to offer their data. Two of them were CKAN portals, while mCLOUD.de had to be scraped using SQUIRREL's HTML scraper. Only europeandataportal.eu offered a SPARQL endpoint to access the dataset's metadata. The data integrated by OPAL were to be written back into the mCLOUD.de portal and cater for the needs of private and public organisations requiring mobility data. Users of this data range from large logistic companies needing to plan transport of goods, to single persons mapping their movement with pollen concentration.

Another project that made use of SQUIRREL to collect data from the Web was LIMBO.[48] Its aim was to unify and refine mobility data of the German Federal Ministry of Transport and Digital Infrastructure. The refined data was made available to the general public to create the basis for new, innovative applications. To this end, SQUIRREL was used to collect this and related data from different sources.

4.7 Conclusion

This chapter presented SQUIRREL, a scalable, distributed and extendable crawler for the Data Web, which provides support for several different protocols and data serializations. Other open-source crawlers currently available are either not able to process RDF data, are limited in the types of data formats they can process, or

[47] http://web.archive.org/web/20220309232608/http://projekt-opal.de/en/welcome-project-opal/; last accessed on 04.08.2022.

[48] https://www.limbo-project.org/; last accessed on 04.08.2022.

are restricted in their scalability. SQUIRREL addresses these drawbacks by providing an extensible architecture adaptable to supporting any format of choice. Moreover, the framework was implemented for simple deployment both locally and at a large scale.

In Addition, we presented ORCA—the first extensible FAIR benchmark for Data Web crawlers, which measures the efficiency and effectiveness of crawlers in a comparable and repeatable way. Using ORCA, we compared SQUIRREL with LDSpider in a repeatable setup. We showed that ORCA revealed strengths and limitations of both crawlers. SQUIRREL was able to crawl data from different sources (HTTP, SPARQL, and CKAN) and compression formats (zip, gzip, bz2), leading to a higher recall than LDSpider. LDSpider was more resource efficient throughout the experiments. Additionally, we showed that ORCA can be used to evaluate the politeness of a crawler, i.e., whether it abides by the Robots Exclusion Protocol.

Both approaches will be extended in various ways in future work. For SQUIRREL, it's efficiency is the main focus for future development and improvement. Future versions of ORCA will include HTML pages with Microdata, Microformat or JSON-LD. A similar extension will be the addition of further compression algorithms to the dump nodes (e.g., `tar`), as well as the HDT serialization [96]. The generation step will be further improved by adding literals and blank nodes to the generated RDF datasets and altering the dataset sizes. A simulation of network errors will round up the next version of the benchmark.

5 A Topic Model for the Data Web

With crawlers like the one described in the previous chapter, we are able to access a large amount of data of the Semantic Web. However, with the size of the available data comes the need to process this data in a scalable way. The large number of datasets that are available online and their sheer size make it costly or even infeasible to handle each of these datasets manually without support of proper tools. One major issue that arises is that even the identification of datasets that are of interest for a particular task may become challenging since experts are able to read RDF data but may not have the time to look into hundreds of thousands of datasets. Hence, *we need to be able to characterize RDF datasets so that users can easily find datasets of interest*.

A similar problem is already known from the processing of large amounts of human-readable documents. Most users might be able to read all books within a library. However, they may not have the time to do so just to identify the books that they are interested in. Although there are search engines that allow the indexing of documents, users would have to know the right keywords to find documents that they are interested in [123]. Hence, "search engines are not the perfect tool to explore the unknown in document collections" [123]. Instead, topic modeling algorithms can be used to infer latent topics in a given document collection. These topics can be used to structure the document collection and enable users to focus on subsets of the collection.

In this chapter, we apply topic modeling to a large set of RDF datasets. Our approach allows users to explore datasets already available on the Web. To this end, we tackle the challenge of transforming the RDF datasets into a form that allows the application of a topic modeling inference algorithm.

A second challenge arises from the inferred topic model itself. The unsupervised topic modeling inference algorithm *gives no guarantees on the interpreteability of its output*. Topics that are generated by a topic modeling inference algorithm might

¶ Parts of this chapter have been published as conference article [227]. The author of this thesis is also the main author of the published article and developed the idea, designed and implemented the solution, and wrote the majority of the publication.

be hard to understand by a user. Thus, they might not be of any use for the task at hand. Since a manual check of generated topics leads to a large effort, an automatic approach for the evaluation of generated topics is needed. We develop the idea of topic coherence presented by Newman et al. [199] further and present a common framework for topic coherence measures dubbed PALMETTO [227]. We use the framework to evaluate 555 660 measures and identify a new topic coherence measure that shows a better performance than previous state-of-the-art measures.

Based on the solutions for both challenges, we present LODCAT—an approach to support the exploration of the Data Web based on human-interpretable topics. Our evaluation shows that this approach can be applied to hundreds of thousands of RDF datasets. The results of a questionnaire suggest that humans generally agree with the topics that our approach assigned to a sample of example datasets.

The following section describes related work before Section 5.2 describes the single steps of our approach dubbed LODCAT. Section 5.3 describes the PALMETTO framework and the evaluation of topic coherence measures. The best topic coherence measure is used within LODCAT during its evaluation, which is described in Section 5.4. Section 5.5 concludes this chapter.

5.1 Related Work

We split the related work into two parts. First, we look at dataset portals and their related work. After that, we list work related to the evaluation of topics.

5.1.1 RDF Dataset Search

In recent years, the Web became a growing source for valuable datasets. While it was originally created to share documents and link these with each other, people started to use the Web for many other activities. This includes the sharing of data [22, 90, 220].

A survey of data search engines was recently published by Chapman et al. [65]. They divide these search engines into four categories. The first category are database search engines. They are used with structured queries that are executed against a database back end. The authors emphasize that vertical search engines that try to access the deep Web are part of this category. In this context, they define the deep Web as the data that cannot be accessed directly by calling a Web page but can be

accessed by using a Web form or some similar functionality. The second set of search engines are information retrieval engines. These are integrated into data portals like CKAN[1] and offer a keyword-based search on the metadata of datasets. This metadata can include tags defined by the creator of the dataset. The third category are entity-centric search engines. The query of such an engine comprises entities of interest and the search engine derives additional information about these entities. The last category is named tabular search. A user of such a search engine tries to extend or manipulate one or more existing tables by executing search queries.

The second category of Chapman et al. [65] represents the most common approach to tackle the search for datasets on the Web. Several open data portals exist that offer a list of datasets and a search on the dataset's metadata. Examples are the aforementioned CKAN, kaggle[2] or open government portals like the european data portal[3]. The Google dataset search presented by Brickley et al. [53] works in a similar way. It collects metadata of datasets from various portals and offers a search on this collected metadata. However, in contrast to a data portal, the Google crawler collects the data from different sources. Internally, it is based on the established RDF vocabularies DCAT [173] and Schema.org.[4] Data providers can embed the metadata of their datasets into their Web page using microdata [144], JSON-LD [261] or RDFa [8]. The Google crawler extracts this information during its Web crawl and forwards it to the Google dataset search engine. Our approach differs to these approaches as we focus on RDF datasets and rely on the dataset itself instead of only using metadata. In addition, we do not rely on a keyword search or user created tags but automatically generated topics that are assigned to the datasets.

Singhal et al. [255] present DataGopher—a dataset search that is optimized for research datasets. The authors tackle the problem that most datasets do not have a long textual description and, hence, cannot be found easily in a classic keyword-based search. They make use of a search engine for scientific articles to retrieve articles that mention a dataset. These articles are further processed to generate a context for the dataset. When a user executes a keyword-based search the extracted context of the datasets is taken into account in addition to classical features like title or description.

Devaraju et al. [81] propose a personalized recommendation of datasets based on user behavior. Their approach combines a similarity measure that is based on datasets' metadata and their co-occurrence, i.e., how many users downloaded both

[1] `https://ckan.org/`; last accessed on 05.08.2022.
[2] `https://www.kaggle.com/datasets`; last accessed on 05.08.2022.
[3] `https://data.europa.eu/en`; last accessed on 05.08.2022.
[4] `https://schema.org/`; last accessed on 05.08.2022.

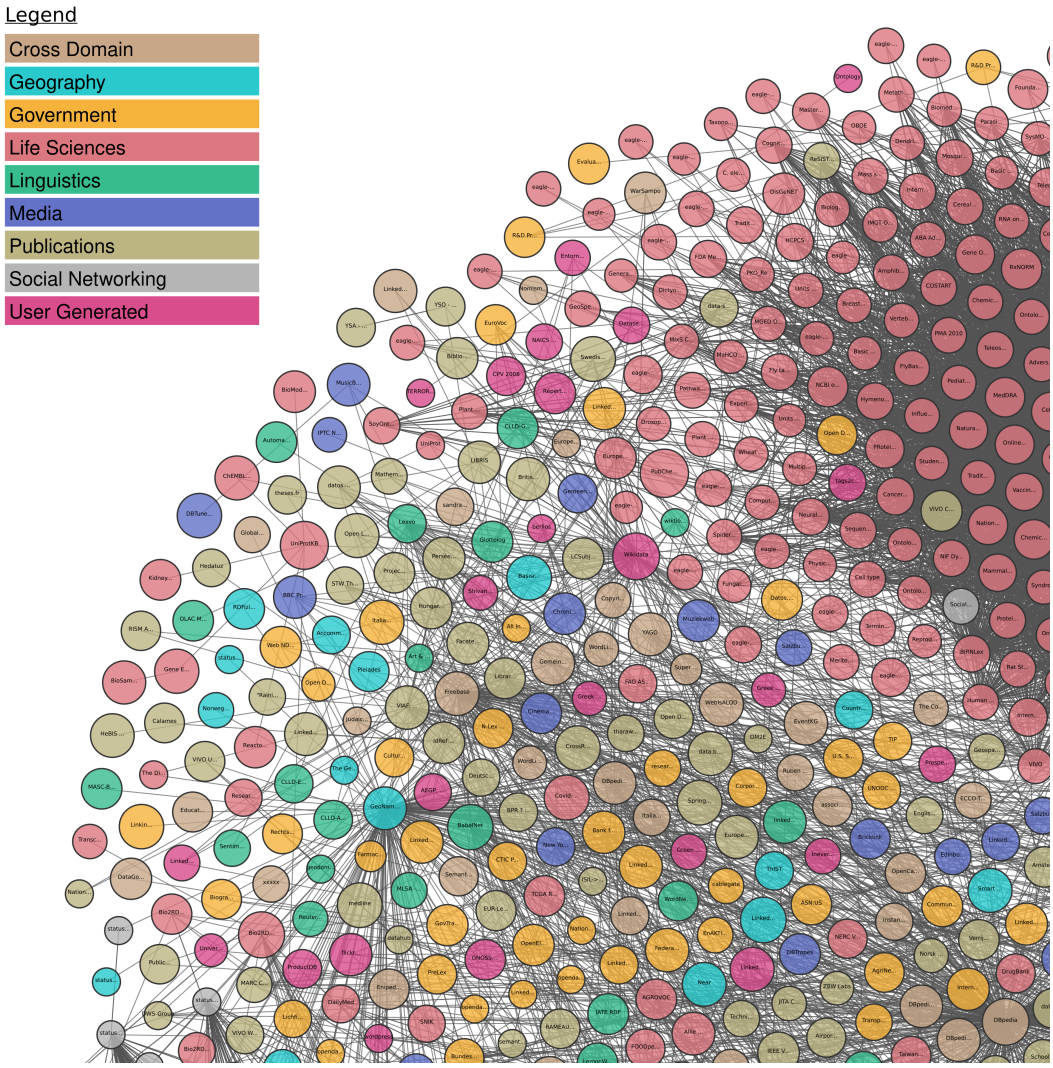


Figure 5.1.: A part of the Linked Open Data cloud diagram [177]. Each circle represents an RDF dataset. The black lines indicate links between two datasets.

datasets. The metadata as well as the access to datasets is gathered from an open dataset portal.

A related project is the Linked Open Data cloud project [177]. It maintains an overview of central datasets of the Semantic Web—dubbed Linked Open Data cloud. Figure 5.1 depicts a part of the diagram generated based on the project data. Although the project gives a nice first idea of a part of the Semantic Web it comes with several drawbacks. First and foremost, the whole data is curated manually. Dataset creators have to submit the metadata of their datasets through a form. This manual curation will become infeasible as soon as we try to do the same for the whole Semantic Web. With the form, the data creator assigns the dataset to one

of 9 pre-defined categories that are listed in the upper left corner of Figure 5.1. This manual classification of the dataset is another drawback. In comparison, our approach LODCat handles datasets automatically and assigns a mixture of topics, which allows a more precise representation of the datasets.

Wagner et al. [299] propose an approach to enable users to query across multiple, separated structured data sources. A user starts by defining an information need as SPARQL query or keyword search based on a certain dataset. The system suggests the addition of further datasets based on the similarity of datasets. This similarity is calculated with an entity-centric approach. Entities are extracted from the single datasets together with all their triples. Based on the triples, a similarity with entities of other datasets is calculated. Based on the pair-wise entity similarities, entity clusters are created which are used to characterize the datasets. Finally, given a dataset, a score for each other dataset can be calculated based on 1) the amount of new but similar entities the second dataset has and 2) the amount of new properties the second dataset offers for these entities. The authors also provide a user interface with a keyword search to enable non-experts to interact with their system. With this search, users can identify possible entities of interest. Starting from an entity, they can further include entities of other sources into their search. However, our approach is not based on keywords and not centered on entities. Hence, we offer a different perspective on the datasets.

Vandenbussche et al. [295] present a Web search for RDF vocabularies.[5] However, it is designed as a classic keyword-based search on the metadata and single elements of the vocabulary combined with manually curated tags. Hence, the user has to match exactly the right term to be able to identify a potentially interesting vocabulary.

Several approaches exist to explore RDF datasets. Tzitzikas et al. [278] define a theoretical framework for these explorative search engines and compare several approaches. However, all these approaches focus on exploring a single RDF dataset while our goal is to explore a set of RDF datasets. Thus, in our use case, the user starts on a higher level with much more data. However, this does not exclude that our approach could be combined with existing explorative searches for RDF datasets. With such a combination, a user could start to choose one or a small number of RDF datasets using LODCat and further explore these datasets using existing tools.

Kunze et al. [155] propose an explorative search engine for a set of RDF datasets. This engine is mainly based on filters that work similar to a faceted search. For ranking, the authors use a similarity function that comprises different aspects. One of these aspects is called *topical aspect* and is based on the RDF vocabularies that are

[5] `https://lov.linkeddata.es/dataset/lov`; last accessed on 05.08.2022.

used inside the different datasets. The more vocabularies are similar, the higher is the topical similarity.

LODAtlas [217] combines several features of the previously mentioned systems into a single user interface. The user can run complex key-word-based queries or do an explorative search based on several dataset features like RDF vocabularies, RDF classes, user defined tags and others. In addition, the user can create plots, e.g., to compare several selected datasets with respect to their size or their links to each other.

Kopsachilis et al. [149] propose GeoLOD—a dataset catalog that focuses on geographical RDF datasets. It extends the metadata of datasets with geo-spatial data that is extracted from the dataset. This enables the user to perform search queries via maps and geo-spatial features. Our approach analyses the given RDF dataset as well but is not limited to geo-spatial data.

Sleeman et al. [256] proposed an approach to use topic modeling with RDF data. While their work has a similar basis it differs in many ways since it aims at other use cases. Their approach generates a single document for every entity described in a dataset while our approach creates a single document for every RDF dataset. Thus, their documents are based on a different set of triples and on different textual data gathered from the dataset.

5.1.2 Topic Evaluation

The evaluation of topic models can be carried out on two levels: 1) the complete topic model can be evaluated or 2) the single topics can be checked independently from each other. The classic approach for evaluating a topic model is to calculate the likelihood of held-out data [41, 63]. The higher the likelihood of the data, the higher is the quality of the model. However, for evaluating single topics, topic modeling experts relied on looking at the most important words of a topic. Each topic is a distribution over all words. We sort the words in descending order based on their probability within the topic's distribution to retrieve the topic's *top words*. Based on these most important words, a human expert decides whether a topic has a good or bad quality. However, this manual process to check whether topics are coherent is very expensive. Additionally, Chang et al. [63] showed that measures like the likelihood of held-out data are not or sometimes even negatively correlated with human ratings. Thus, the evaluation of topic models needs an additional measure that can rate topics automatically with respect to human understandability and interpretability [63].

Alsumait et al. [15] propose to check whether a topic distribution has features that identify it as a bad topic. To this end, they define that a topic is bad if its distribution is:

1. Similar to an even distribution,
2. Similar to the corpus' background distribution, or
3. Likely to be a mixture of several topics instead of a single topic.

Another approach that also focuses on the topic distributions is proposed by Mimno et al. [184] and is called posterior prediction checks. The main idea is to sample observations from the generated topic model and check whether the sampled data differs from the original corpus. For each word in a topic, the difference between its predicted occurrence and its real occurrence can be measured to achieve an evaluation of the single topics. However, both aforementioned approaches are difficult to link with manually generated gold standards.

Newman et al. [199] propose to focus on the topic's top words. An example of such a set is {*game, sport, ball, team*}, which we will use for the purpose of illustration throughout the section. In the paper, the authors begin by collecting human ratings (good, neutral or bad) for sets of top words of generated topics. Several automatic topic ranking methods that measure topic coherence are evaluated by comparison to these human ratings. The evaluated topic coherence measures take the set of n top words of a topic and sum a confirmation measure over all word pairs. A confirmation measure depends on a single pair of top words. Several confirmation measures are evaluated by the authors. The coherence based on PMI gives largest correlations with human ratings during their evaluation. Let $\overline{W} = \{\mathrm{w}_1, \ldots, \mathrm{w}_n\}$ be the ordered set of n top words of a given topic, i.e., they are the word types with the n highest probabilities in the topic's word distribution ϕ. The set is ordered in descending order by the word type's probabilities in the topic's word distribution ϕ starting with w_1 as the word type with the highest probability. The *UCI coherence* proposed by Newman et al. [199] is calculated by:[6]

$$\mathcal{C}_{\mathrm{UCI}}(\overline{W}) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{PMI}_{\epsilon}(\mathrm{w}_i, \mathrm{w}_j)\,. \tag{5.1}$$

Formally, probabilities are estimated based on word co-occurrence counts [199]. Those counts are derived from documents that are constructed by a sliding window that moves over articles of the Wikipedia, which is used as external reference corpus.

[6] We adapt the equation and make use of PMI_{ϵ} as defined in Equation 2.30 to avoid the logarithm of zero. Note that Newman et al. [199] do not only mention the arithmetic mean but also the median as technique to summarize the values of the pairwise comparisons.

Each window position defines such a document. Let $\overline{W}_{\text{ex}}$ be the set of top words of the example topic described above. For this example topic, the coherence proposed by Newman et al. [199] would be calculated as follows:

$$\begin{aligned} \mathcal{C}_{\text{UCI}}(\overline{W}_{\text{ex}}) = \frac{1}{6}(&\text{PMI}_\epsilon(\textit{game}, \textit{sport}) + \text{PMI}_\epsilon(\textit{game}, \textit{ball}) \\ &+ \text{PMI}_\epsilon(\textit{game}, \textit{team}) + \text{PMI}_\epsilon(\textit{sport}, \textit{ball}) \\ &+ \text{PMI}_\epsilon(\textit{sport}, \textit{team}) + \text{PMI}_\epsilon(\textit{ball}, \textit{team}))\,. \end{aligned} \tag{5.2}$$

Mimno et al. [185] propose to use the smoothed conditional probability as asymmetrical confirmation measure between top word pairs. The summation of *UMass coherence* accounts for the ordering among the top words of a topic. The coherence can be calculated as follows:[7]

$$\mathcal{C}_{\text{UMass}}(\overline{W}) = \frac{2}{n(n-1)} \sum_{i=2}^{n} \sum_{j=1}^{i-1} \log \frac{\mathbb{P}(\mathrm{w}_i, \mathrm{w}_j) + \epsilon}{\mathbb{P}(\mathrm{w}_j)}\,. \tag{5.3}$$

The word probabilities are estimated based on document frequencies of the original documents used for learning the topics [185]. The calculation for our example would be:

$$\begin{aligned} \mathcal{C}_{\text{UMass}}(\overline{W}_{\text{ex}}) = \frac{1}{6}\Bigg(&\log\left(\frac{\mathbb{P}(\textit{sport}, \textit{game}) + \epsilon}{\mathbb{P}(\textit{game})}\right) + \log\left(\frac{\mathbb{P}(\textit{ball}, \textit{game}) + \epsilon}{\mathbb{P}(\textit{game})}\right) \\ &+ \log\left(\frac{\mathbb{P}(\textit{ball}, \textit{sport}) + \epsilon}{\mathbb{P}(\textit{sport})}\right) + \log\left(\frac{\mathbb{P}(\textit{team}, \textit{game}) + \epsilon}{\mathbb{P}(\textit{game})}\right) \\ &+ \log\left(\frac{\mathbb{P}(\textit{team}, \textit{sport}) + \epsilon}{\mathbb{P}(\textit{sport})}\right) + \log\left(\frac{\mathbb{P}(\textit{team}, \textit{ball}) + \epsilon}{\mathbb{P}(\textit{ball})}\right)\Bigg)\,. \end{aligned} \tag{5.4}$$

Stevens et al. [265] found that both—UCI and UMass coherence—perform better if the parameter ϵ is chosen to be rather small instead of $\epsilon = 1$ as in respective original publications.

Aletras et al. [10] introduce topic coherence based on context vectors for every top word. A context vector of a word type w is created using word co-occurrence counts determined using context windows that contain all words located ± 5 tokens around

[7] Note that ideally, one would want to compute $\log(\mathbb{P}(\mathrm{w}_i, \mathrm{w}_j)\,/\mathbb{P}(\mathrm{w}_j))$. However, this can lead to numerical errors when $\mathbb{P}(\mathrm{w}_i, \mathrm{w}_j) = 0$. ϵ is hence set to be a small positive value, which ensures that our computations are sound.
Note that Mimno et al. [185] originally define the coherence as the sum of the pairwise comparisons while we define it as the arithmetic mean. We do that since the sum and arithmetic mean are equivalent aggregation techniques with respect to the topic coherence task. Both techniques will lead to the same order of rated topics with the same relative distance to each other as long as a constant number of top words is used.

the occurrences of the word type w in the reference corpus. Largest correlation to human topic coherence ratings were found when defining the elements of these vectors as NPMI [49] of the word pairs. Additionally, they showed that restricting the word co-occurrences to those words that are part of the same topic performs best (*top word space*). Thus, the j-th element of the context vector $\vec{\mathrm{v}}_i$ of word type w_i has the following value:

$$\mathrm{v}_{ij} = \mathrm{NPMI}_\epsilon(\mathrm{w}_i, \mathrm{w}_j)^\kappa = \left(\frac{\log \frac{\mathbb{P}(\mathrm{w}_i, \mathrm{w}_j)+\epsilon}{\mathbb{P}(\mathrm{w}_i)\mathbb{P}(\mathrm{w}_j)}}{-\log(\mathbb{P}(\mathrm{w}_i, \mathrm{w}_j)+\epsilon)} \right)^\kappa . \tag{5.5}$$

κ is a weight parameter. An increase of κ gives higher NPMI values more weight. For our example topic, the vector of its top word *game* would be calculated as:

$$\begin{aligned} \vec{\mathrm{v}}_{game} = \{&\mathrm{NPMI}_\epsilon(game, game)^\kappa, \mathrm{NPMI}_\epsilon(game, sport)^\kappa, \\ &\mathrm{NPMI}_\epsilon(game, ball)^\kappa, \mathrm{NPMI}_\epsilon(game, team)^\kappa\} . \end{aligned} \tag{5.6}$$

Confirmation measures between pairs of context vectors are vector similarities like cosine, Dice or Jaccard [10] that are averaged over all pairs of a topic's top words as suggested by Newman et al. [199]. The cosine coherence for the example top words would be calculated as:

$$\begin{aligned} \mathcal{C}_{\cos}(\overline{W}_{\mathrm{ex}}) = \frac{1}{6}(&\cos(\vec{\mathrm{v}}_{game}, \vec{\mathrm{v}}_{sport}) + \cos(\vec{\mathrm{v}}_{game}, \vec{\mathrm{v}}_{ball}) \\ &+ \cos(\vec{\mathrm{v}}_{game}, \vec{\mathrm{v}}_{team}) + \cos(\vec{\mathrm{v}}_{sport}, \vec{\mathrm{v}}_{ball}) \\ &+ \cos(\vec{\mathrm{v}}_{sport}, \vec{\mathrm{v}}_{team}) + \cos(\vec{\mathrm{v}}_{ball}, \vec{\mathrm{v}}_{team})) . \end{aligned} \tag{5.7}$$

Alternatively, topic coherence is computed as average similarity between top word context vectors and their centroid $\vec{\mathrm{v}}_c$ [10]:

$$\vec{\mathrm{v}}_c = \vec{\mathrm{v}}_{game} + \vec{\mathrm{v}}_{sport} + \vec{\mathrm{v}}_{ball} + \vec{\mathrm{v}}_{team} , \tag{5.8}$$

$$\begin{aligned} \mathcal{C}_{\mathrm{cen}}(\overline{W}_{\mathrm{ex}}) = \frac{1}{4}(&\cos(\vec{\mathrm{v}}_{game}, \vec{\mathrm{v}}_c) + \cos(\vec{\mathrm{v}}_{sport}, \vec{\mathrm{v}}_c) \\ &+ \cos(\vec{\mathrm{v}}_{ball}, \vec{\mathrm{v}}_c) + \cos(\vec{\mathrm{v}}_{team}, \vec{\mathrm{v}}_c)) . \end{aligned} \tag{5.9}$$

Additionally, Aletras et al. [10] show that the UCI coherence $\mathcal{C}_{\mathrm{UCI}}$ performs better if the PMI is replaced by the NPMI. We name the latter coherence $\mathcal{C}_{\mathrm{NPMI}}$.

Lau et al. [159] structure the topic evaluation in two different tasks—word intrusion and observed coherence. In the first task, an intruder word has to be identified among the top words of a topic. For the second task, topics have to be rated regarding their coherence, while ratings are compared to human ratings. Both

tasks can be carried out for single topics or the whole topic model. Lau et al. [159] confirm that $\mathcal{C}_{\text{NPMI}}$ performs better than the original $\mathcal{C}_{\text{UCI}}$.

Theoretical work on coherence of sets of statements in a broader sense are reviewed by Douven et al. [83]. We follow their notation but adapt the presentation of measures to word coherence. Shogenji's [254] and Olsson's [207] coherences are defined as:

$$\mathcal{C}_S(\overline{W}) = \frac{\mathbb{P}\left(\mathrm{w}_1, \ldots, \mathrm{w}_n\right)}{\prod_{i=1}^{n} \mathbb{P}\left(\mathrm{w}_i\right)}\,, \tag{5.10}$$

$$\mathcal{C}_O(\overline{W}) = \frac{\mathbb{P}\left(\mathrm{w}_1, \ldots, \mathrm{w}_n\right)}{\mathbb{P}\left(\mathrm{w}_1 \vee \ldots \vee \mathrm{w}_n\right)}\,. \tag{5.11}$$

The usage of these coherences for our example is straight forward:

$$\mathcal{C}_S(\overline{W}_{\text{ex}}) = \frac{\mathbb{P}(\mathit{game}, \mathit{sport}, \mathit{ball}, \mathit{team})}{\mathbb{P}(\mathit{game})\mathbb{P}(\mathit{sport})\mathbb{P}(\mathit{ball})\mathbb{P}(\mathit{team})}\,, \tag{5.12}$$

$$\mathcal{C}_O(\overline{W}_{\text{ex}}) = \frac{\mathbb{P}(\mathit{game}, \mathit{sport}, \mathit{ball}, \mathit{team})}{\mathbb{P}(\mathit{game} \vee \mathit{sport} \vee \mathit{ball} \vee \mathit{team})}\,. \tag{5.13}$$

Fitelson [99] evaluates a single word in the context of all subsets that can be constructed from the remaining words. The set of all subsets without the word type w_i is denoted by $\mathbb{W}_i$ and defined as:

$$\mathbb{W}_i = \left\{W_k | W_k \subseteq \overline{W} \setminus \{\mathrm{w}_i\} \wedge W_k \neq \emptyset\right\}\,. \tag{5.14}$$

Fitelson's coherence is defined by comparing the probability of the i-th word with every single set in $\mathbb{W}_i$:

$$\mathcal{C}_F(\overline{W}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{2^{n-1}-1} m_f(\mathrm{w}_i, \mathbb{W}_{i,j})}{n(2^{n-1}-1)}\,. \tag{5.15}$$

The measure used for the comparison is:

$$m_f(\mathrm{w}_i, \mathbb{W}_{i,j}) = \frac{\mathbb{P}(\mathrm{w}_i|\mathbb{W}_{i,j}) - \mathbb{P}(\mathrm{w}_i|\neg\mathbb{W}_{i,j})}{\mathbb{P}(\mathrm{w}_i|\mathbb{W}_{i,j}) + \mathbb{P}(\mathrm{w}_i|\neg\mathbb{W}_{i,j})}\,. \tag{5.16}$$

Note that this approach takes relationships between word sets into account and goes beyond averaging confirmations between word pairs.[8]

[8] In Section 5.3.1, we will give an example for S^{one}_{any} which is the equivalent to the $(\mathrm{w}_i, \mathbb{W}_{i,j})$ pairs of Fitelson's coherence.

Douven et al. [83] take the idea to go beyond word pairs further by creating pairs of word subsets $S_i = (W', W'')$. These pairs of subsets are used to test whether the existence of the subset W'' supports the occurrence of the subset W'. This support is measured using several confirmation measures and has been adapted to the evaluation of topics by Rosner et al. [233]. The authors found that using larger subsets W' and W'' can lead to better performing topic coherence measures.

While several coherence measures have been proposed, there is no work that creates a unifying framework for their definition. Additionally, a comparative evaluation of all the coherences is missing. We provide both in Section 5.3.

5.2 LODCat

Figure 5.2 shows an overview of our proposed approach LODCat. It relies on a large reference text corpus (e.g., the English Wikipedia) as a source of general knowledge and uses topic modeling to assign human-readable topical labels to the single RDF datasets. First, we use the reference corpus to generate topic models. After that, each generated model is evaluated and the best model is chosen for further processing. In parallel, the RDF datasets are transformed into a textual representation (i.e., documents). Based on the chosen topic model, a topic distribution is assigned to each of the generated documents. In addition, a label is generated for each of the topics. These labels are used to make the complex topic distributions human-readable. At the end, each RDF dataset has a set of topics that are dominant for that dataset and that are described by their labels. This data is used to provide a faceted search, which helps a data provider to find datasets related to their own data. The single steps of our approach are described in more detail in the following.

5.2.1 Topic Inference

The given reference corpus is pre-processed by converting it into plain text. The plain-text documents are further preprocessed using a tokenizer and a lemmatizer [175]. The tokenizer separates a given text into single tokens. The lemmatizer transforms all tokens into a basic word form called lemma [141]. This helps to reduce the number of different word types in the corpus, e.g., by transforming plural forms of

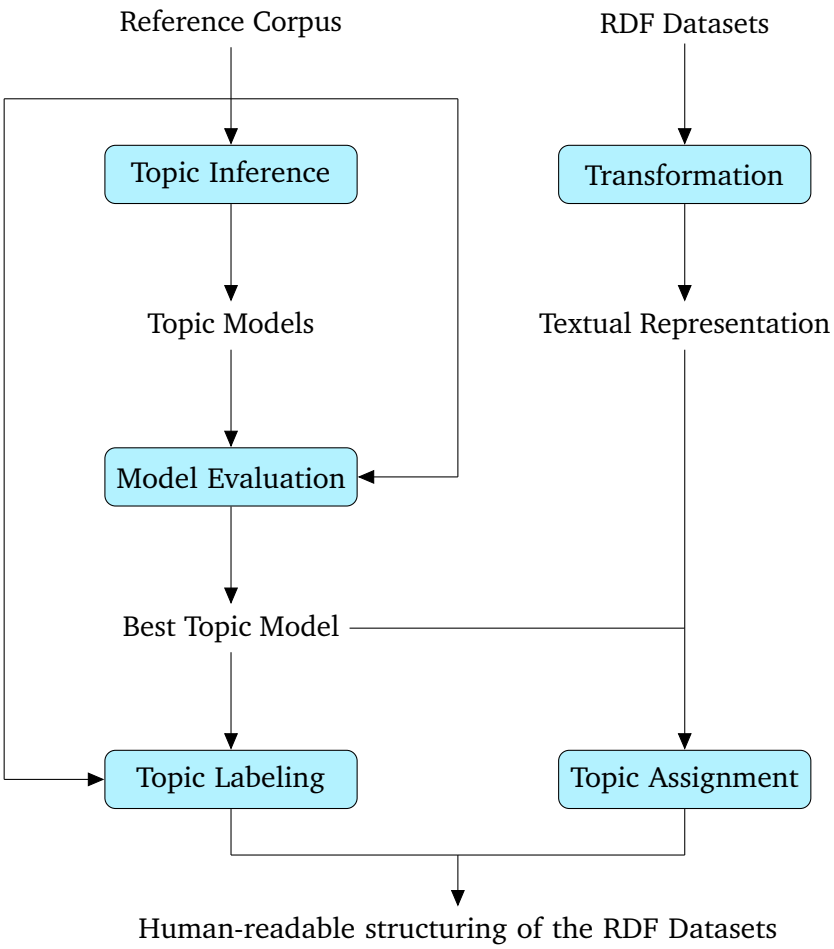


Figure 5.2.: Overview of the workflow of LODCAT.

nouns into their singular form. After that, we remove stop words from the corpus. These include common stop words but also RDF-related terms like “subject”.[9]

After the preprocessing, an LDA inference algorithm is applied. We expect the reference corpus to be large and, hence, use the Variational Bayesian inference proposed by Hoffman et al. [126] and described in Section 2.2.2. Since the best number of topics ϱ is unknown, we generate models with different ϱ values.

5.2.2 Model Evaluation

From the set of generated topic models, the best model has to be chosen. Several strategies can be employed to achieve this goal. A classic way to evaluate a topic model is to predict the probability of existing documents. One such measure is the perplexity, which is calculated on held-out data [43, 126]. Another approach proposed by Griffiths et al. [110] would be to calculate $\mathbb{P}(D|\Phi)$ as explained in Section 2.2.3. However, both approaches focus on the statistical correctness of the model but not on the human readability and interpretability by users. Chang et al. [63] showed that some classic statistical measures are even negatively correlated with human interpretability. Hence, several topic coherence measures have been proposed [10, 159, 185, 199, 225, 233, 265]. These measures are used to evaluate a single topic. Section 5.3 defines a framework for topic coherence measures and presents the results of a detailed comparison of the available coherence measures and more than 500 thousand other measures. In the following, we will use the measure that performed best in the experiments reported in the section aforementioned. For each generated topic model, we calculate the coherence values of the single topics and the average topic coherence of the complete model.

Note that this step does not only select the best model, but also assigns a coherence value to each topic of the selected model. These coherence values are interesting for the presentation of the topics since they enable us to exclude topics that have a low coherence value and, hence, are not easy to interpret by users.

[9] The used stop word list can be found at `https://github.com/MichaelRoeder/topicmodeling/blob/master/topicmodeling.lang/src/main/resources/english.stopwords` and `https://github.com/dice-group/lodcat/blob/develop/lodcat.model/src/main/resources/stopwords.txt`; last accessed on 05.08.2022.

5.2.3 Topic Labeling

After identifying the best topic model, LODCAT creates labels for each of the model's topics to summarize the meaning of the topics for the users. The task of generating such labels—dubbed topic labeling—is a known field of research and LODCAT can use any of the available labeling methods [152, 157, 158]. For our current implementation, we chose the Neural Embedding Topic Labelling (NETL) approach of Bathia et al. [37] since 1) their evaluation shows that NETL outperformed the approach of Lau et al. [158] and 2) the approach is available as an open-source project.[10] NETL comprises two steps. First, label candidates are generated for a given topic. Second, the candidates are ranked according to a trained model.

The candidate generation is mainly based on the reference corpus. Bathia et al. [37] propose the English Wikipedia and use it as input to generate two embedding models using the doc2vec [160] and word2vec [181] algorithms. The first can represent a natural language phrase as a vector in an embedding space while the latter does the same for single words. The NETL algorithm extracts the document titles from the same reference corpus and rates them according to their similarity to the top words of the given topic. Let $\overline{W}$ be the set of top words of the given topic as defined in Section 5.1.2 and W_L the topic label candidate. Further, let $\mathfrak{e}^w_{w2v}$ and $\mathfrak{e}^w_{d2v}$ be the embedding functions that embed a given word into a word2vec and doc2vec model, respectively.[11] Let $\mathfrak{e}^d_{d2v}$ be the embedding function that embeds a given phrase into a doc2vec model and let $\text{sim}_{\cos}$ be the cosine similarity between two vectors. The algorithm calculates the overall similarity sim_{lt} between a label candidate and a set of top words as sum of the cosine similarities between the embeddings of the label candidate and the top words in doc2vec and word2vec, respectively. The similarity is defined as follows:

$$\text{sim}_{lt}(W_L, \overline{W}) = \frac{1}{|\overline{W}|} \sum_{\text{w} \in \overline{W}} \Big(\text{sim}_{\cos}\left(\mathfrak{e}^d_{d2v}(W_L), \mathfrak{e}^w_{d2v}(\text{w})\right) + \text{sim}_{\cos}\left(\mathfrak{e}^w_{w2v}(W_L), \mathfrak{e}^w_{w2v}(\text{w})\right)\Big) . \tag{5.17}$$

In the second step, NETL reranks the best 19 label candidates for a topic using a supervised regression algorithm. This reranking is based on the following features [37]:

[10] `https://github.com/sb1992/NETL-Automatic-Topic-Labelling-`; last accessed on 05.08.2022.

[11] It is possible to use other embedding algorithms for the representation of words and documents. However, we stick to the embedding algorithms suggested by Bathia et al. [37].

- The letter trigram similarity of $\overline{W}$ and W_L [152],
- The page rank [208] of the document from which the label candidate was taken,
- The number of words in the label [158], and
- The number of overlapping words of $\overline{W}$ and W_L [158].

Bathia et al. [37] train a support vector regression model to rank the label candidates based on a dataset comprising 228 topics with human-rated labels. In our current implementation, we use the pre-generated embedding models, the page rank scores, and the pre-trained classification model of NETL [37].

After this step, each topic has a human-readable representation. This representation comprises the label derived with NETL and the topic's 10 top words, which are used as additional description of the topic.

5.2.4 RDF Dataset Transformation

The given RDF datasets are transformed into a textual representation that can be used in combination with the generated topic model. This step relies on the IRIs that occur in the datasets. We determine the frequency of each IRI in the dataset (either as subject, object or predicate of a triple). IRIs of well-known namespaces that do not have any topical value like `rdf`, `rdfs`, and `owl` are filtered out. After that, the labels of each IRI are retrieved. The label retrieval is based on the list of IRIs identified as label-defining properties by Ell et al. [87]. If there are no labels available, the namespace of the IRI is removed and the remaining part is used as label. If this generated label is written in camel case or contains symbols like underscores, it is split into multiple words. If an IRI has a description, i.e., a triple with the `rdf:comment` property, it is treated as additional label. The derived labels are further preprocessed using a tokenizer and a lemmatizer [175] as described in 5.2.1. We create a list of word types that are used for the label(s) of each IRI. For each word type w, we determine a frequency count $\mathrm{f_w}$ by summing up all counts of all IRIs, which have this word type in their list. However, we do not use the counts directly for generating a document since some IRIs may occur hundreds of thousand times. Their words would dominate the generated document and marginalize the influence of other words. In addition, large count values would lead to very long documents that could create problems with respect to the memory consumption and the runtime of the inference. To reduce the influence of words with very high

f values we determine the frequency of word type w for the document of the i-th dataset as follows:

$$\psi_{i,\mathrm{w}} = \mathrm{r}(\log_2(\mathrm{f}_\mathrm{w}) + 1)\,, \tag{5.18}$$

where r is the rounding function which returns the closest integer value preferring the higher value in case of a tie [3].[12] The result of this step is a bag of words representation of one document for each RDF dataset.

5.2.5 Topic Assignment

The last step of our approach is the assignment of topics to the documents that represent the RDF datasets. For each document, a topic inference is executed, similar to the inference described in Section 2.2.2. However, this inference is limited to the documents generated in the previous step, i.e., for each document, only the "E" step of the inference is executed. Hence, only the document's γ and ξ are optimized while the topic model's χ is treated as a constant.[13] The inference provides us with a distribution over the topics for each document. This distribution is further used together with the topic labels and top words as a human-readable representation of the RDF dataset that is represented by this document.

5.3 Topic Evaluation

The evaluation of a topic with respect to its human understandability and interpretability is crucial for topic-modeling-based approaches like LODCat. Since our approach relies on presenting topics to the user, it is important that the topics are helpful for the user and can be understood easily. Since classic topic modeling algorithms do not give any guarantees with respect to the quality of their topics, Chang et al. [63] argue that an automatic measure for the coherence of a given word set is needed.

The automatic calculation of the coherence of a set is researched in other areas as well. Generally, a set of statements or facts is said to be coherent if the statements in the set support each other [50]. Thus, a coherent fact set can be interpreted in a context that covers all or most of the facts. An example of a coherent fact set is {"the game is a team sport", "the game is played with a ball", "the game demands great

[12] The transformation of counts into occurrences in a synthetic document is similar to the logarithmic variant of Tapioca described in Section 6.2.2.

[13] The meanings of γ, ξ, and χ are explained in Section 2.2.2.

physical effort"}. A long-standing open question is how to quantify the coherence of a fact set [50]. Approaches proposed in scientific philosophy have formalized measures as functions of joint and marginal probabilities associated with the facts. Bovens et al. [50] discuss many examples that lead to a demanding set of complex properties such a measure needs to fulfill. An example is the non-monotonic behavior of coherence in case of growing fact sets. The coherence of the two statements "the animal is a bird" and "the animal cannot fly" can be increased by adding the fact "the animal is a penguin". The non-monotonicity becomes apparent when the coherence is lowered again by adding non-related facts [50]. The discussion of coherence measures in that community deals mainly with schemes that estimate the hanging and fitting together of the individual facts of a larger set. Examples of such schemes are 1) to compare each fact against the rest of all other fact, 2) compare all pairs against each other, and 3) compare disjointed subsets of facts against each other. Such theoretical work on coherence from scientific philosophy—see Douven et al. [83] for an overview—has potential to be adapted for the coherence of word sets.

The seminal work of Newman et al. [199] proposes automatic coherence measures that rate topics regarding to their understandability. The proposed measures reduces a topic to its top words. This important restriction will apply to all analyses presented in this section. Furthermore, Newman et al. [199] restrict coherence to be always based on comparing word pairs. Our analyses will go beyond this limitation.

Evaluations proposed by Newman et al. [199] are based on human-generated topic rankings and showed that measures based on word co-occurrence statistics estimated on Wikipedia outperform measures based on WordNet [182] and similar semantic resources. Subsequent empirical works on topic coherence proposed a number of measures based on word statistics that differ in several details [159, 185, 265]: definition, normalization, and aggregation of word statistics and reference corpus. In addition, a method based on word context vectors has been proposed by Aletras et al. [10].

Looking at the two lines of research on coherence—scientific philosophy and topic modelling—we note that the contributions are mainly complementary. While the former proposes a good number of schemes for comparing facts or words, the latter proposes useful methods for estimating word probabilities and normalizing numeric comparisons. However, a systematic, empirical evaluation of the methods of both worlds and their yet unexplored combinations is still missing.

Human topic rankings serve as the gold standard for coherence evaluation. However, they are expensive to produce. There are three publicly available sources of such

rankings at the time of writing: 1) Chang et al. [63] that have been prepared by Lau et al. [159] for topic coherence evaluation, 2) Aletras et al. [10], and 3) Rosner et al. [233]. A systematic, empirical evaluation should take all these sources into account. For this reason, we choose the concept of a framework providing an objective platform for comparing the different approaches. Following our research agenda, this can lead to completely new insights into the behavior of different algorithms with regard to the available benchmarks. Hence, it will be possible to finally evaluate the reasons for specific behavior of topic coherences on a comparable basis.

Our contributions within this section are the following:

1. We propose a unifying framework that spans a configuration space of topic coherence definitions. The signature of this space will be explained in the Section 5.3.1.
2. We exhaustively search this space for the coherence definition with the best overall correlation with respect to all available human topic ranking data. This search empirically evaluates published coherence measures as well as unpublished ones based on combinations of known approaches.
3. Our results reveal a coherence measure based on a new combination of known approaches that approximates human ratings better than the state of the art.[14]

5.3.1 Framework of Coherence Measures

Our new unifying framework represents a coherence measure as composition of parts that can be freely combined. Hence, existing measures as well as yet unexplored measures can be constructed. The parts are grouped into dimensions that span the configuration space of coherence measures. Each dimension is characterized by a set of exchangeable components. The first dimension is the kind of segmentation that is used to divide a word set into smaller pieces. These pieces are compared against each other, e.g., segmentation into word pairs. The set of different kinds of segmentation is $\mathcal{S}$. The second dimension is the confirmation measure that scores the agreement of a given pair, e.g., the NPMI of two words. The set of confirmation measures is $\mathfrak{M}$. Confirmation measures use word probabilities that can be computed in different ways, which forms the third dimension of the configuration space. The set of methods to estimate word probabilities is $\mathcal{P}$. Last, the methods of how to

[14] Data and tools for replicating our coherence calculations are available at `https://github.com/dice-group/palmetto`.

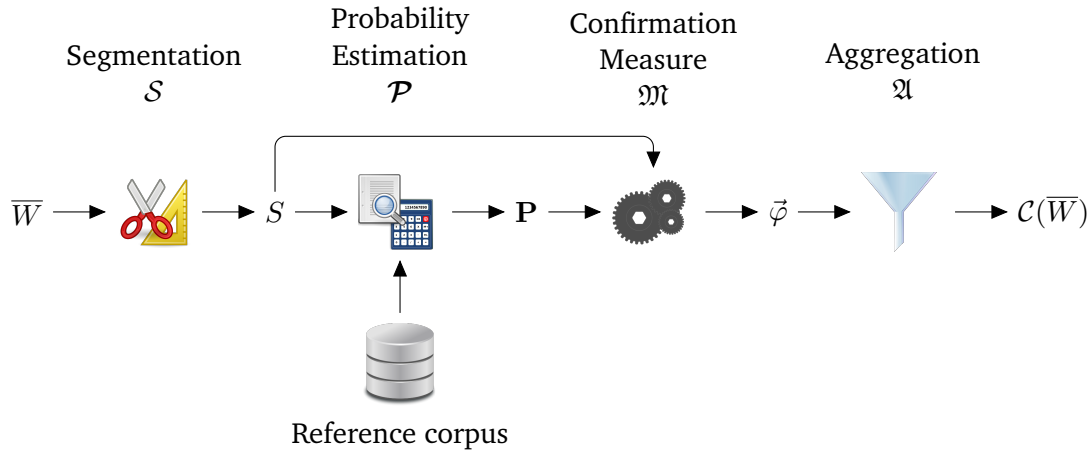


Figure 5.3.: Overview over the unifying coherence framework—its four parts and their intermediate results.

aggregate scalar values computed by the confirmation measure forms the fourth dimension. The set of aggregation functions is $\mathfrak{A}$.

The workflow of our framework as shown in figure 5.3 comprises four steps. First, the word set $\overline{W}$ is segmented into a set of pairs of word subsets S. Second, word probabilities $\mathbf{P}$ are computed based on a given reference corpus. Both, the set of word subsets S as well as the computed probabilities $\mathbf{P}$ are consumed by the confirmation measure to calculate the agreements φ of elements of S. Last, those values are aggregated to a single coherence value $\mathcal{C}(\overline{W})$.

In summary, the framework defines a space of configurations that is the cross product of the four sets $\mathscr{C} = \mathcal{S} \times \mathcal{P} \times \mathfrak{M} \times \mathfrak{A}$. In the following subsections, these four dimensions are explained in more detail.

Segmentation of word subsets

Following Douven et al. [83], coherence of a word set measures the degree that a subset is supported by another subset. The result of the segmentation of a given set of top words $\overline{W}$ is a set of pairs of subsets of $\overline{W}$. The definition of a subset pair consists of two parts that are differently used by the following confirmation measures. The first part of a pair is the subset for which the support by the second part of the pair is determined.

Definition 5.1 (Segmentation)**.** *A segmentation S_*^* of a given word set $\overline{W}$ is a set of pairs of subsets:*

$$S_*^* = \left\{ (W', W'') \middle| W' \subseteq \overline{W} \wedge W'' \subseteq \overline{W} \right\}. \tag{5.19}$$

The notation of the segmentation contains two $*$ symbols which are replaced by two labels. The upper label expresses the definition of W' while the lower label defines W''. We use S_i to denote a single pair of word subsets, i.e., $S_i = (W', W'')$ and $S_i \in S_*^*$.

Most proposed coherence measures for topic evaluation compare pairs of single words, e.g., C_{UCI} defined in Section 5.1.2. Every single word is paired with every other single word. Those segementations are called *one-one* and are defined as follows:

$$S_{one}^{one} = \left\{ (W', W'') \middle| W' = \{\mathrm{w}_i\} \wedge W'' = \{\mathrm{w}_j\} \wedge \mathrm{w}_i \in \overline{W} \wedge \mathrm{w}_j \in \overline{W} \wedge i \neq j \right\}. \tag{5.20}$$

Mimno et al. [185] propose to take the order of words within the set of top words into account. The following two segmentations are variations of S_{one}^{one} and compare a word only to the preceding or succeeding words, respectively:

$$S_{pre}^{one} = \left\{ (W', W'') \middle| W' = \{\mathrm{w}_i\} \wedge W'' = \{\mathrm{w}_j\} \wedge \mathrm{w}_i \in \overline{W} \wedge \mathrm{w}_j \in \overline{W} \wedge i > j \right\}, \tag{5.21}$$

$$S_{suc}^{one} = \left\{ (W', W'') \middle| W' = \{\mathrm{w}_i\} \wedge W'' = \{\mathrm{w}_j\} \wedge \mathrm{w}_i \in \overline{W} \wedge \mathrm{w}_j \in \overline{W} \wedge i < j \right\}. \tag{5.22}$$

Douven et al. [83] proposed several other segmentations that have been adapted to topic evaluation by Rosner et al. [233]. These definitions allow one or both subsets to contain more than one single word:

$$S_{all}^{one} = \left\{ (W', W'') \middle| W' = \{\mathrm{w}_i\} \wedge \mathrm{w}_i \in \overline{W} \wedge W'' = \overline{W} \setminus \{\mathrm{w}_i\} \right\}, \tag{5.23}$$

$$S_{any}^{one} = \left\{ (W', W'') \middle| W' = \{\mathrm{w}_i\} \wedge \mathrm{w}_i \in \overline{W} \wedge W'' \subseteq \overline{W} \setminus \{\mathrm{w}_i\} \wedge W'' \neq \emptyset \right\}, \tag{5.24}$$

$$S_{any}^{any} = \left\{ (W', W'') \middle| W' \subset \overline{W} \wedge W'' \subset \overline{W} \wedge W' \cap W'' = \emptyset \wedge W' \neq \emptyset \wedge W'' \neq \emptyset \right\}. \tag{5.25}$$

S^{one}_{all} compares every single word to all other words of the word set. S^{one}_{any} extends S^{one}_{all} by using every subset as condition. S^{any}_{any} is another extension that compares every subset with every other disjoint subset. Figure 5.4 shows the different sets of subset pairs produced by applying the different segmentations to an example word set.

The approach of Aletras et al. [10] compares words to the complete word set $\overline{W}$ using word context vectors. Therefore, we define another segmentation

$$S^{one}_{set} = \left\{ (W', \overline{W}) \middle| W' = \{\mathrm{w}_i\} \wedge \mathrm{w}_i \in \overline{W} \right\}. \tag{5.26}$$

Note that this segmentation does not obey the requirement $W' \cap W'' = \emptyset$ stated by Douven et al. [83].

Further, we define S^{all}_{one} and S^{set}_{set} as follows:

$$S^{all}_{one} = \left\{ (W', W'') \middle| W' = \overline{W} \setminus \{\mathrm{w}_i\} \wedge W'' = \{\mathrm{w}_i\} \wedge \mathrm{w}_i \in \overline{W} \right\}, \tag{5.27}$$

$$S^{set}_{set} = \left\{ (\overline{W}, \overline{W}) \right\}. \tag{5.28}$$

These special segmentation schemes are used to represent Shogenji's and Olsson's coherence measures within our framework.

Probability Estimation

The method of probability estimation defines the way how the probabilities are derived from the underlying data source. *Boolean document* ($\mathcal{P}_{bd}$) estimates the probability of a single word as the number of documents in which the word occurs divided by the total number of documents. In the same way, the joint probability of two words is estimated by the number of documents containing both words divided by the total number of documents. This estimation method is called boolean as the number of occurrences of words in a single document as well as distances between the occurrences are not considered. The UMass coherence is based on an equivalent kind of estimation [185]. Text documents with some formatting allow simple variations, namely the *boolean paragraph* ($\mathcal{P}_{bp}$) and *boolean sentence* ($\mathcal{P}_{bs}$). These estimation methods are similar to *boolean document* except instead of documents paragraphs or sentences are used respectively.

The *boolean sliding window* ($\mathcal{P}_{sw(*)}$) determines word counts using a sliding window. The window moves over the documents one word token per step. Each step defines a new virtual document by copying the window content. *Boolean document* is applied

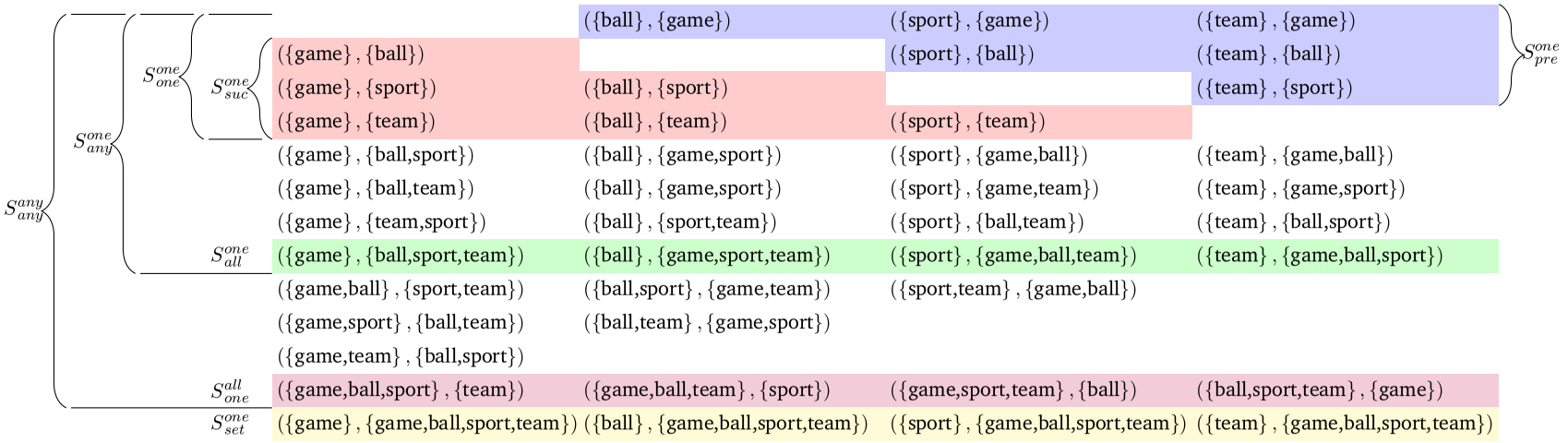


Figure 5.4.: $S^{any}_{any}, S^{one}_{all}, S^{one}_{any}, S^{one}_{one}, S^{one}_{pre}, S^{one}_{set}$, and S^{one}_{suc} segmentations of the word set {game, ball, sport, team} and their hierarchy.

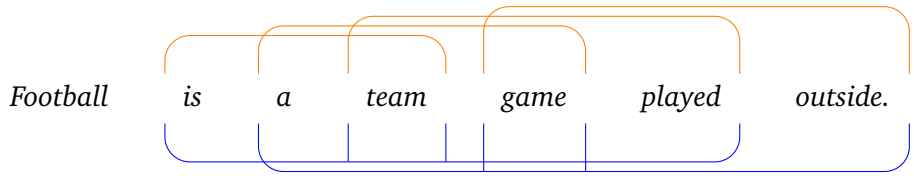


Figure 5.5.: An example document with sliding windows of $\mathcal{P}_{sw(3)}$ (orange) and context windows of $\mathcal{P}_{cw(2)}$ (blue) for the searched words *team* and *game*. Note that only windows relevant for the searched words are shown. That means that sliding windows should contain at least one of the searched words and context windows are centered on one of these words.

to these virtual documents to compute word probabilities. Note that *boolean sliding window* captures the proximity between word tokens to some degree. The window size defines how many words are located within the window and is added to the name, e.g., $\mathcal{P}_{sw(10)}$ for a sliding window of size 10.

The *boolean context window* ($\mathcal{P}_{cw(*)}$) used by Aletras et al. [10] is always centered on a searched word. All words within the window range to the left and the right of the word are counted as cooccurrence. However, it suffers the problem that it can create counts for larger subsets that are higher than the counts of smaller sets that are subsets of the larger sets. This is counter-intuitive as the occurrence of larger subsets is expected to be lower. We add the window size to the name, e.g., $\mathcal{P}_{cw(5)}$ for a context window that covers ± 5 words.

Figure 5.5 shows an example document with the sliding windows of size 3 and the context windows with size ± 2 that are used for the search of the cooccurrence of the words *team* and *game*. The sliding window approach would count 3 occurrences of each of the single words and 2 cooccurrences, since they occur only in 2 windows together. The context windows would count 2 occurrences of the single words and 2 cooccurrences. To transform the counts into a probability, the window-based approaches would divide the counts by the maximum number of windows. If we assume that this is the only document in our reference corpus, there are 5 sliding windows of size 3 and 3 context windows of size ± 2. $\mathcal{P}_{bd}$, $\mathcal{P}_{bp}$, and $\mathcal{P}_{bs}$ would count this example as 1 occurrence of both terms and 1 cooccurrence. The difference between these three probability estimations would become only visible for larger examples that comprise several paragraphs.

Confirmation Measure

Definition 5.2 (Confirmation measure)**.** *A confirmation measure takes a single pair $S_i = (W', W'')$ of word subsets as well as the corresponding probabilities to compute how strong the conditioning word set W'' supports W'.*

The calculation can be carried out either directly [83, 185, 199] or indirectly [10].

Direct confirmation measures. Measures to directly compute the confirmation of a single pair S_i of words or word subsets are:

$$m_c(S_i) = \frac{\mathbb{P}(W', W'')}{\mathbb{P}(W'')}\,, \tag{5.29}$$

$$m_{lc}(S_i) = \log \frac{\mathbb{P}(W', W'') + \epsilon}{\mathbb{P}(W'')}\,, \tag{5.30}$$

$$m_d(S_i) = \mathbb{P}(W'|W'') - \mathbb{P}(W')\,, \tag{5.31}$$

$$m_r(S_i) = \frac{\mathbb{P}(W', W'')}{\mathbb{P}(W')\mathbb{P}(W'')}\,, \tag{5.32}$$

$$m_{lr}(S_i) = \log \frac{\mathbb{P}(W', W'') + \epsilon}{\mathbb{P}(W')\mathbb{P}(W'')}\,, \tag{5.33}$$

$$m_{nlr}(S_i) = \frac{m_{lr}(S_i)}{-\log\left(\mathbb{P}(W', W'') + \epsilon\right)}\,, \tag{5.34}$$

$$m_l(S_i) = \frac{\mathbb{P}(W'|W'')}{\mathbb{P}(W'|\neg W'') + \epsilon}\,, \tag{5.35}$$

$$m_{ll}(S_i) = \log \frac{\mathbb{P}(W'|W'') + \epsilon}{\mathbb{P}(W'|\neg W'') + \epsilon}\,, \tag{5.36}$$

$$m_{\mathbb{P}}(S_i) = \mathbb{P}(W', W'')\,, \tag{5.37}$$

$$m_{\text{Jac}}(S_i) = \frac{\mathbb{P}(W', W'')}{\mathbb{P}(W' \vee W'')}\,, \tag{5.38}$$

$$m_{l\text{Jac}}(S_i) = \log \frac{\mathbb{P}(W', W'') + \epsilon}{\mathbb{P}(W' \vee W'')}\,. \tag{5.39}$$

Douven et al. [83] call the confirmation measures m_d, m_r, and m_l difference, ratio, and likelihood measure, respectively. There, log likelihood (m_{ll}) and log ratio measure (m_{lr}) are also defined—the latter is the PMI, the central element of the UCI coherence. The normalized log ratio measure (m_{nlr}) is the NPMI. The log conditional probability measure (m_{lc}) is equivalent to the calculation used by the UMass coherence [185]. The joint probability ($m_{\mathbb{P}}$) of the two sets is proposed by Aletras et al. [10]. The last two confirmation measures are the Jaccard and log Jaccard measures. A small constant ϵ is added to prevent the logarithm of

zero, where necessary. Following Stevens et al. [265], we set it to a small value ($\epsilon = 10^{-12}$).[15] Olsson's and Fitelson's coherences as well as a logarithmic variant of Shogenji's coherence (Equations 5.11, 5.16 and 5.10) are denoted by m_o, m_f and m_{ls}, respectively.

Indirect confirmation measures. Instead of directly computing the confirmation of $S_i = (W', W'')$, an indirect computation of confirmation assumes that given some word of $\overline{W}$, direct confirmations of words in W' are close to direct confirmations of words in W'' with respect to this given word. Thus, indirect confirmations compute the similarity of words in W' and W'' with respect to direct confirmations to all words of the set of top words. Their advantage can be explained with an example. Assume word w_i semantically supports word w_j but they do not appear frequently together in the reference corpus and have therefore a low joint probability. Thus, their direct confirmation would be low as well. However, the confirmations of these words correlate with respect to many other words in $\overline{W}$. These two words could be competing brands of cars, which semantically support each other. However, both brands are seldom mentioned together in documents in the reference corpus. But their confirmations to other words like "road" or "speed" do strongly correlate. This would be reflected by an indirect confirmation measure. Thus, indirect confirmation measures may capture semantic support that direct measures would miss.

This idea can be formalized by representing the word sets W' and W'' as vectors of length n. Such vectors—dubbed context vectors—can be computed with respect to any direct confirmation measure m. In case W' and W'' consist of single words, the vector elements are just the direct confirmations as suggested by Aletras et al. [10]. For the case that a vector for a set of words is needed, we define the vector elements as the sum of the direct confirmations of the single words. Based on a direct confirmation measure m and a weight parameter κ the context vector for a subset of top words, e.g., W', is calculated as follows:

$$\vec{\mathrm{v}}_{m,\kappa,W'} = \left\{ \sum_{\mathrm{w}_i \in W'} m(\mathrm{w}_i, \mathrm{w}_j)^\kappa \right\}_{\substack{j=1,\ldots,n \\ \mathrm{w}_j \in \overline{W}}} . \tag{5.40}$$

Following Aletras et al. [10], the vector elements can be non-linearly distorted. Let $\vec{\mathrm{v}}'$ and $\vec{\mathrm{v}}''$ be the context vectors of the two word sets W' and W'', respectively. For the word sets of a pair $S_i = (W', W'')$, the indirect confirmation is computed as

[15] Additionally to the measures discussed by Stevens et al. [265], we use ϵ for m_l and m_{ll} as well to prevent a division by 0.

vector similarity. Following Aletras et al. [10], we equip our framework with the vector similarities cosine, Dice, and Jaccard, which are defined as follows:

$$\text{sim}_{\text{cos}}(\vec{\text{v}}', \vec{\text{v}}'') = \frac{\sum_{i=1}^{n} \text{v}_i' \text{v}_i''}{\|\vec{\text{v}}'\|_2 \|\vec{\text{v}}''\|_2}\,, \tag{5.41}$$

$$\text{sim}_{\text{Dice}}(\vec{\text{v}}', \vec{\text{v}}'') = \frac{\sum_{i=1}^{n} 2\texttt{min}(\text{v}_i', \text{v}_i'')}{\sum_{i=1}^{n} \text{v}_i' + \text{v}_i''}\,, \tag{5.42}$$

$$\text{sim}_{\text{Jac}}(\vec{\text{v}}', \vec{\text{v}}'') = \frac{\sum_{i=1}^{n} \texttt{min}(\text{v}_i', \text{v}_i'')}{\sum_{i=1}^{n} \max(\text{v}_i', \text{v}_i'')}\,. \tag{5.43}$$

Thus, given a similarity measure sim, a direct confirmation measure m and a value for κ, an indirect confirmation measure $\tilde{m}$ is defined as

$$\tilde{m}_{\text{sim},m,\kappa}(W', W'') = \text{sim}\left(\vec{\text{v}}_{m,\kappa,W'}, \vec{\text{v}}_{m,\kappa,W''}\right)\,. \tag{5.44}$$

Aggregation

Finally, all confirmations $\vec{\varphi} = \{\varphi_1, \ldots, \varphi_{|S|}\}$ of all subset pairs S_i are aggregated to a single coherence score $\mathcal{C}(\overline{W})$. The arithmetic mean ($a_a$) and median ($a_m$) have been used in the literature [199]. Additionally, we evaluate the geometric mean (a_g), harmonic mean (a_h), quadratic mean (a_q), minimum (a_n), and maximum (a_x).

Representation of existing measures

In the following, we will show how to describe all coherence measures from Section 5.1.2 as instances within our framework. The UCI, UMass, and NPMI coherences are defined as follows:

$$\mathcal{C}_{\text{UCI}} = \left(\mathcal{P}_{sw(10)}, S^{one}_{one}, m_{lr}, a_a\right)\,, \tag{5.45}$$

$$\mathcal{C}_{\text{UMass}} = \left(\mathcal{P}_{bd}, S^{one}_{pre}, m_{lc}, a_a\right)\,, \tag{5.46}$$

$$\mathcal{C}_{\text{NPMI}} = \left(\mathcal{P}_{sw(10)}, S^{one}_{one}, m_{nlr}, a_a\right)\,. \tag{5.47}$$

The coherences defined by Douven et al. [83] and adapted by Rosner et al. [233] are written as follows:

$$\mathcal{C}_{\text{one-all}} = \left(\mathcal{P}_{bd}, S^{one}_{all}, m_d, a_a\right)\,, \tag{5.48}$$

$$\mathcal{C}_{\text{one-any}} = \left(\mathcal{P}_{bd}, S^{one}_{any}, m_d, a_a\right)\,, \tag{5.49}$$

$$\mathcal{C}_{\text{any-any}} = \left(\mathcal{P}_{bd}, S^{any}_{any}, m_d, a_a\right)\,. \tag{5.50}$$

Shogenji's [254], Olsson's [207], and Fitelson's [99] coherences do not define how the probabilities are computed. Therefore, these measure definitions can be combined with every method of probability estimation. We indicate this with the wildcard symbol $*$ in their definitions:

$$\mathcal{C}_S = \left(*, S^{all}_{one}, m_{ls}, a_a\right) , \tag{5.51}$$

$$\mathcal{C}_O = \left(*, S^{set}_{set}, m_o, *\right) , \tag{5.52}$$

$$\mathcal{C}_F = \left(*, S^{one}_{any}, m_f, a_a\right) . \tag{5.53}$$

All of the defined aggregation functions can be used for $\mathcal{C}_O$ since the S^{set}_{set} segmentation leads to the creation of a single pair of word sets.

Using the context-window-based probability estimation $\mathcal{P}_{cw}$ described in Section 5.1.2, we are able to formulate the context-vector-based coherences defined by Aletras et al. [10] within our framework:

$$\mathcal{C}_{\text{cos}} = \left(\mathcal{P}_{cw(5)}, S^{one}_{one}, \tilde{m}_{\text{cos},m_{nlr},\kappa}, a_a\right) , \tag{5.54}$$

$$\mathcal{C}_{\text{Dice}} = \left(\mathcal{P}_{cw(5)}, S^{one}_{one}, \tilde{m}_{\text{Dice},m_{nlr},\kappa}, a_a\right) , \tag{5.55}$$

$$\mathcal{C}_{\text{Jac}} = \left(\mathcal{P}_{cw(5)}, S^{one}_{one}, \tilde{m}_{\text{Jac},m_{nlr},\kappa}, a_a\right) , \tag{5.56}$$

$$\mathcal{C}_{\text{cen}} = \left(\mathcal{P}_{cw(5)}, S^{one}_{set}, \tilde{m}_{\text{cos},m_{nlr},\kappa}, a_a\right) . \tag{5.57}$$

We showed that the framework can cover all existing topic coherence measures. However, it also allows the definition of new coherence measures that combine the ideas of existing measures.

5.3.2 Evaluation Setup

The evaluation follows a common scheme that has already been used in the related work [10, 159, 199, 233]. Coherence measures are computed for topics given as word sets that have been rated by humans with respect to understandability. Each measure produces a ranking of the topics that is compared to the ranking induced by human ratings. Following Lau et al. [159], both rankings are correlated using the Pearson correlation. Thus, good quality of a coherence measure is indicated by a high correlation with human ratings.

In the literature, other evaluation methods have been used as well, e.g., humans were asked to classify word sets using different given error types [185]. However,

since the necessary data is not freely available, we cannot use such methods for our evaluation.

A dataset used for our evaluation comprises a corpus, topics, and human ratings. Topics are computed using the corpus and are given by word sets consisting of the topics top words. Human ratings for topics had been created by presenting these word sets to human raters. Topics are rated regarding interpretability and understandability using three categories—good, neutral or bad [199]. The generation of such a dataset is expensive due to the necessary manual work to create human topic ratings. Several datasets have been published [10, 63, 159, 233]. Additionally, the creation of such a dataset is separated from the topic model used to compute the topics, since the humans rate just plain word sets without any information about the topic model [10, 63, 159, 224, 233]. This opens the possibility to reuse them for evaluation.

The *20NG* dataset contains the 20 Newsgroups corpus that consists of Usenet messages of 20 different groups.[16] The *Genomics* corpus comprises scientific articles of 49 MEDLINE journals and is part of the TREC-Genomics Track.[17] Aletras et al. [10] published 100 rated topics for both datasets, each represented by a set of 10 top words. Further, they published 100 rated topics that have been computed using 47 229 New York Times articles (*NYT*). Unfortunately, this last corpus is not available to us.

Chang et al. [63] used two corpora, one comprising New York Times articles (*RTL-NYT*) and the other is a Wikipedia subset (*RTL-Wiki*). A number of 900 topics were created for each of these corpora. Lau et al. [159] published human ratings for these topics. Human raters evaluated word subsets of size five randomly selected from the top words of each topic. We aggregated the ratings for each word set. Word sets with less than three ratings or words with encoding errors are removed.[18] The RTL-Wiki corpus is published in a bag-of-words format that is unsuitable for paragraph-, sentence- or window-based probability estimations. Therefore, we have retrieved the articles in version of May 2009 from Wikipedia history records. Not all of the original 10 000 articles were available anymore. Therefore, the recreated corpus comprises only 7 838 documents.

[16] `http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html`; last accessed on 05.08.2022.

[17] `http://web.archive.org/web/20141020171050/http://ir.ohsu.edu/genomics/`; last accessed on 05.08.2022

[18] The RTL-Wiki dataset contains 23 word sets with 6 words and more than 3 ratings that were removed as well to ensure comparability of ratings.

Table 5.1.: Datasets used for the evaluation.

Name	20NG	Genomics	NYT	RTL-NYT	RTL-Wiki	Movie
Topics	100	100	100	1095	1096	100
Top words	10	10	10	5	5	5
Documents	19 952	29 833	—	—	7 838	108 952
Paragraphs	155 429	2 678 088	—	—	319 859	2 136 811
Sentences	341 583	9 744 966	—	—	1 035 265	6 583 202
Tokens	2 785 319	114 065 923	—	—	13 679 052	86 256 415
Vocabulary	109 610	1 640 456	—	—	591 957	1 625 124

Rosner et al. [233] published the *Movie* corpus—a Wikipedia subset—and 100 rated topics. Topics are given as sets of five top words. Like the RTL-Wiki corpus this corpus was recreated and has only 108 952 of the original 125 411 documents.

Table 5.1 shows an overview of the statistics of the different datasets used for evaluation. Word counts and probabilities necessary to calculate the coherence values are derived from the English Wikipedia. In case the corpus, which was used as training data for topic learning, is available, we compute coherence measures a second time using counts derived from that corpus. All corpora as well as the complete Wikipedia used as reference corpus are preprocessed using lemmatization and stop word removal. Additionally, we removed portal and category articles, redirection and disambiguation pages as well as articles about single years.

During our evaluation, we test a wide range of different parameter settings. We use the values $\{10, 20, \ldots, 300\}$ and $\{5, 10, \ldots, 150\}$ for the sliding and the context window size, respectively. The parameter κ varied in $\{1, 2, 3\}$. Overall, our evaluation comprises a total of 555 660 different coherences and parameterizations.

5.3.3 Results and Discussion

Table 5.2 shows the best performing coherence measures with respect to the different datasets. The largest correlations for all datasets (except for the Movie dataset) were reached, when the coherence measures relied on probabilities derived from the Wikipedia instead of the corpus used for topic learning. We focus our discussion on these calculations.

Following Demsar [78], a direct comparison of achieved correlation values on datasets from different domains is not suggested. Hence, we compare the different approaches using ranks. For each of the datasets, we sort all 555 660 coherence

Table 5.2.: Coherence measures with strongest correlations with human ratings. The upper part shows results for using the original corpus and the lower part those when using the Wikipedia to derive probabilities, respectively.

			$\mathcal{C}_{V2}$	$\mathcal{C}_{P}$	$\mathcal{C}_{\text{UMass}}$	$\mathcal{C}_{\text{one-any}}$	$\mathcal{C}_{\text{UCI}}$	$\mathcal{C}_{\text{NPMI}}$	$\mathcal{C}_{\text{cos}}$
Coherences		$\mathcal{S}$	S^{one}_{all}	S^{one}_{pre}	S^{one}_{pre}	S^{one}_{any}	S^{one}_{one}	S^{one}_{one}	S^{one}_{one}
		$\boldsymbol{\mathcal{P}}$	$\mathcal{P}_{sw(110)}$	$\mathcal{P}_{sw(70)}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{sw(10)}$	$\mathcal{P}_{sw(10)}$	$\mathcal{P}_{cw(5)}$
		$\mathfrak{M}$	$\tilde{m}_{\text{cos},m_{nlr},1}$	m_f	m_{lc}	m_d	m_{lr}	m_{nlr}	$\tilde{m}_{\text{cos},m_{nlr},1}$
		$\mathfrak{A}$	a_a	a_a	a_a	a_a	a_a	a_a	a_a
Or. corpus		20NG	0.665	**0.756**	0.395	0.563	0.312	0.486	0.563
		Genomics	**0.671**	0.652	0.514	0.549	0.624	0.630	0.632
		RTL-Wiki	**0.627**	0.615	0.272	0.545	0.527	0.573	0.542
		Movie	0.548	**0.549**	0.093	0.453	0.473	0.438	0.431
Wikipedia	$n = 10$	20NG	**0.832**	0.825	0.555	0.822	0.747	0.809	0.790
	$n = 10$	Genomics	**0.726**	0.721	0.461	0.452	0.602	0.671	0.640
	$n = 10$	NYT	**0.820**	0.757	0.519	0.612	0.751	0.798	0.733
	$n = 5$	RTL-NYT	**0.736**	0.720	0.099	0.438	0.544	0.659	0.630
	$n = 5$	RTL-Wiki	**0.684**	0.645	0.336	0.499	0.548	0.609	0.579
	$n = 5$	Movie	**0.542**	0.533	0.143	0.454	0.447	0.452	0.465
		Average rank	**2 685.0**	4 677.2	170 129.2	50 099.5	32 093.4	16 442.0	20 019.8
		Standard dev.	4 287.6	5 304.1	120 432.8	34 769.6	22 373.6	20 132.6	14 970.6

measures in descending order based on the correlation that they achieved on that dataset. Based on this order, we assign ranks to the coherence measures.[19] Then, we calculate the average rank and the standard deviation that the measures achieve across the six datasets when the Wikipedia is used as reference corpus. Table 5.2 shows these values in the last two lines. In addition, we use a Wilcoxon signed rank test [305] to compare a chosen set of coherence measures across their performance on all six datasets.[20] Table 5.3 shows the results of the pairwise comparisons. Looking at already proposed coherence measures (five most right columns of Tables 5.2 and 5.3), our results confirm that on average the UCI coherence performs better with NPMI. Among already proposed coherence measures, $\mathcal{C}_{\text{NPMI}}$ shows the best performance and is able to significantly outperform $\mathcal{C}_{\text{UMass}}$ and $\mathcal{C}_{\text{UCI}}$. Slightly lower correlations and, hence, a higher average rank are obtained by $\mathcal{C}_{\text{cos}}$, which is the best performing vector-based coherence of those proposed by Aletras et al. [10] within our experiment. However, the difference to $\mathcal{C}_{\text{NPMI}}$ is not significant. The UMass coherence has lower correlations and the highest average rank in Table 5.2. Especially for smaller word sets, this coherence measure

[19] We make use of shared ranks, i.e., if two or more coherence measures achieve the same correlation value, we assign the average of their ranks to these measures. Coherence measures for which the correlation is not defined (e.g., because a coherence measure assigns the same value to all word sets of the dataset) are put at the end of the ranked list.

[20] We use a Wilcoxon signed rank test with a threshold of 0.05%. As Demsar [78] point out, a coherence measure has to achieve better correlation values on all six datasets to significantly outperform another coherence measure.

Table 5.3.: Results of the pairwise Wilcoxon signed-rank tests. $+$ means that the coherence measure in the row significantly outperforms the measure in the column. $-$ expresses the opposite. 0 means that the differences of both coherences are not significant.

	$\mathcal{C}_{V2}$	$\mathcal{C}_{P}$	$\mathcal{C}_{\text{UMass}}$	$\mathcal{C}_{\text{one-any}}$	$\mathcal{C}_{\text{UCI}}$	$\mathcal{C}_{\text{NPMI}}$	$\mathcal{C}_{\text{cos}}$
$\mathcal{C}_{V2}$		$+$	$+$	$+$	$+$	$+$	$+$
$\mathcal{C}_{P}$	$-$		$+$	$+$	$+$	0	$+$
$\mathcal{C}_{\text{UMass}}$	$-$	$-$		0	$-$	$-$	$-$
$\mathcal{C}_{\text{one-any}}$	$-$	$-$	0		0	0	0
$\mathcal{C}_{\text{UCI}}$	$-$	$-$	$+$	0		$-$	0
$\mathcal{C}_{\text{NPMI}}$	$-$	0	$+$	0	$+$		0
$\mathcal{C}_{\text{cos}}$	$-$	$-$	$+$	0	0	0	

does not seem to achieve good results. Shogenji's (average rank 276 893.5) and Olsson's (267 729.5) coherences (not shown in Table 5.2) have high average ranks and correlations close to zero, while Fitelson's coherence (57 190.7) is comparable to $\mathcal{C}_{\text{one-any}}$ proposed by Rosner et al. [233].[21]

The best performing coherence measure (the leftmost column) is a new combination found by our systematic study of the configuration space of coherence measures. This measure ($\mathcal{C}_{V2}$) combines the indirect cosine measure with the NPMI and the boolean sliding window.[22] This combination has been overlooked so far in the literature and significantly outperforms all previously proposed coherence measures. Also, the best direct coherence measure ($\mathcal{C}_{P}$) found by our study is a new combination. It combines Fitelson's confirmation measure with the boolean sliding window. $\mathcal{C}_{P}$ shows a significantly better performance than all other previously proposed coherence measures except $\mathcal{C}_{\text{NPMI}}$. While $\mathcal{C}_{P}$ achieves a higher correlation than $\mathcal{C}_{\text{NPMI}}$ for five datasets, it has a lower correlation on the NYT dataset.

Among probability estimation methods, the boolean paragraph, boolean sentence, and context window methods perform better than the boolean document (see Table 5.4). The boolean sliding window performs best, but the window size should be larger than the size proposed by Newman et al. [199]. Figure 5.6 shows the average ranks achieved by variants of $\mathcal{C}_{V2}$, $\mathcal{C}_{P}$, $\mathcal{C}_{\text{UCI}}$, and $\mathcal{C}_{\text{NPMI}}$ with different window sizes. It shows that only the $\mathcal{C}_{P}$ coherence achieves a good average rank with a small window ($s = 10$). It reaches its best average ranks with a window size of 40 to 70 word tokens. The ranks of $\mathcal{C}_{V2}$ and $\mathcal{C}_{\text{NPMI}}$ remain on a low level, when the window

[21] More detailed results for the already proposed coherence measures can be found in the appendix in Section A.3.

[22] We name the coherence $\mathcal{C}_{V2}$ since the name C_V (for vector-based coherence) has been used by us in [227]. However, the C_V coherence showed severe performance issues.

Table 5.4.: Best average ranks for the probability estimations, segmentations, and aggregations if they were combined with a direct or indirect confirmation measure (standard deviation in parenthesis).

Name	Direct		Indirect	
Name	Avg. rank	Std. dev.	Avg. rank	Std. dev.
$\mathcal{P}_{bd}$	21 186.7	(20 287.4)	14 736.2	(16 346.6)
$\mathcal{P}_{bp}$	15 974.8	(13 854.7)	9 001.0	(7 555.2)
$\mathcal{P}_{bs}$	11 384.3	(9 043.4)	8 296.8	(13 536.0)
$\mathcal{P}_{sw(70)}$	4 677.2	(5 304.1)	3 150.7	(5 644.1)
$\mathcal{P}_{sw(110)}$	5 275.7	(5 621.6)	2 685.0	(4 287.6)
$\mathcal{P}_{cw(15)}$	16 908.3	(12 029.4)	11 358.7	(17 454.2)
$\mathcal{P}_{cw(20)}$	16 994.3	(12 758.9)	10 488.0	(14 823.2)
S^{all}_{one}	85 393.8	(79 805.3)	2 685.0	(4 287.6)
S^{any}_{any}	26 508.5	(22 316.8)	2 753.2	(3 864.4)
S^{one}_{all}	85 393.8	(79 805.3)	2 685.0	(4 287.6)
S^{one}_{any}	21 186.7	(20 287.4)	3 023.5	(4 757.7)
S^{one}_{one}	4 809.2	(5 383.1)	4 286.0	(4 758.2)
S^{one}_{pre}	4 677.2	(5 304.1)	4 286.0	(4 758.2)
S^{one}_{set}	183 405.1	(85 810.1)	2 778.7	(4 169.6)
S^{one}_{suc}	5 350.5	(5 484.2)	4 286.0	(4 758.2)
S^{set}_{set}	258 283.6	(141 447.3)	306 677.3	(88 002.8)
a_a	4 677.2	(5 304.1)	2 685.0	(4 287.6)
a_g	81 130.2	(36 979.5)	28 430.7	(18 998.7)
a_h	97 568.8	(67 942.5)	36 584.3	(19 231.1)
a_m	12 524.2	(7 955.2)	2 753.2	(3 864.4)
a_n	86 219.2	(34 504.0)	51 933.0	(11 386.2)
a_q	20 378.5	(31 065.3)	5 217.2	(5 894.5)
a_x	83 088.4	(80 401.4)	8 587.5	(9 714.3)

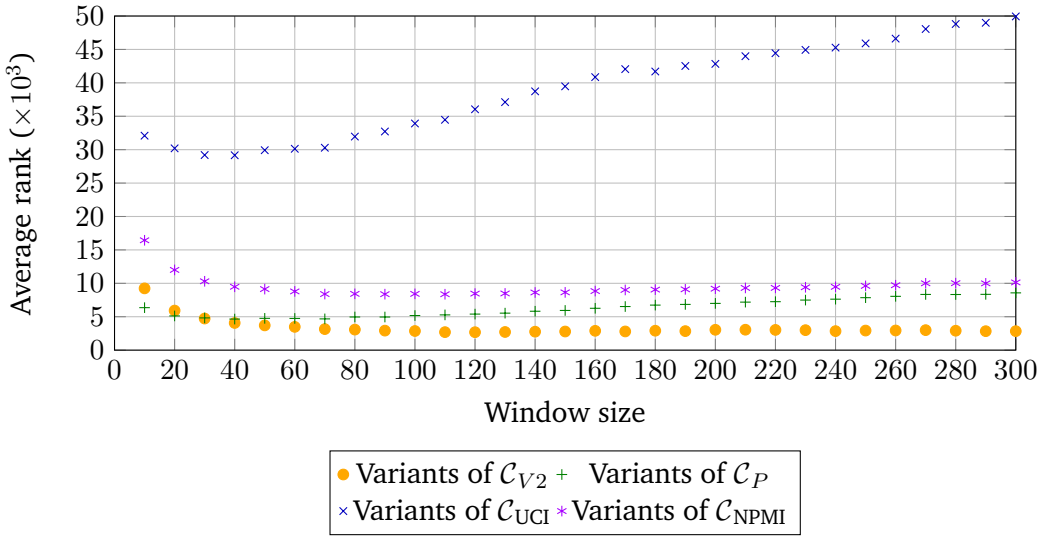


Figure 5.6.: The influence of the sliding window's size on the average ranks of variants of different sliding-window-based coherences (lower values are better).

size is larger than 50. $\mathcal{C}_{\text{UCI}}$ benefits from a larger window size, too, and reaches its best average rank at 40. An explanation for the good performance of the boolean sliding window is that it implicitly represents distances between word tokens within large documents. Further, large documents that are known to have good quality in Wikipedia, are implicitly up weighted because they contain more windows than smaller documents.

Among the segmentation methods, if a direct confirmation measure is used the single-word-based segmentation methods (S^{one}_{one}, S^{one}_{pre}, and S^{one}_{suc}) achieve good ranks, while S^{one}_{set} and S^{set}_{set} have high ranks. This changes when an indirect confirmation measure is used. Nearly all segmentation methods reach a very good rank with indirect confirmation measures. S^{set}_{set} is the only exception with a very high rank.

The arithmetic mean is the aggregation with the best average ranks. Combined with indirect confirmation measures the median achieves a comparable average rank.

Among the direct confirmation measures, m_f achieves the best average rank, followed by m_{nlr} (see Table 5.5). With some distance, m_d and m_{lr} follow. The last three benefit from a combination with an indirect measure, while the rank of m_f drops slightly. In most cases, the indirect measures achieve a better average rank if they use the cosine similarity instead of Dice or Jaccard.

The small differences in correlation of coherences with 1) different window sizes and 2) segmentation methods that are very similar to each other, leads to a large number of coherences having high correlation values that are only slightly lower

Table 5.5.: Best average ranks for the confirmation measures with their standard deviation in parenthesis.

Name	Direct	Indirect		
		$\tilde{m}_{cos}$	$\tilde{m}_{dice}$	$\tilde{m}_{jac}$
m_c	58 288.3 (77 905.9)	48 310.3 (74 366.5)	42 063.2 (59 068.0)	44 833.4 (55 144.0)
m_d	24 998.0 (14 889.2)	14 552.0 (16 444.6)	11 014.0 (11 469.3)	13 397.3 (12 478.7)
m_f	4 650.0 (5 127.1)	5 519.0 (5 558.0)	88 348.8 (44 531.7)	49 917.0 (48 673.2)
m_{Jac}	80 164.7 (36 472.6)	46 320.0 (48 007.4)	41 621.0 (46 860.9)	46 474.8 (44 594.1)
m_l	152 167.7 (91 790.2)	137 029.2 (82 525.2)	142 200.3 (109 696.7)	142 203.5 (109 696.7)
m_{lc}	71 655.0 (58 241.4)	146 413.8 (57 809.7)	226 692.2 (90 988.1)	226 772.0 (90 303.1)
$m_{l\text{Jac}}$	53 171.5 (32 661.9)	224 282.7 (126 293.7)	254 230.5 (110 838.5)	259 565.2 (109 446.3)
m_{ll}	46 740.2 (26 951.3)	55 885.8 (35 941.5)	119 478.0 (98 626.2)	94 395.0 (67 974.3)
m_{lr}	29 162.5 (17 744.9)	10 863.0 (11 944.6)	13 670.0 (8 545.8)	12 763.8 (10 902.0)
m_{ls}	223 416.3 (137 333.9)	55 566.5 (32 509.4)	71 119.2 (36 545.4)	70 952.2 (36 395.9)
m_{nlr}	8 368.2 (9 349.4)	2 685.0 (4 287.6)	10 488.0 (14 823.2)	12 606.9 (13 245.4)
m_o	80 164.7 (36 472.6)	46 320.0 (48 007.4)	41 621.0 (46 860.9)	46 474.8 (44 594.1)
$m_{\mathbb{P}}$	229 723.2 (72 142.1)	61 007.7 (86 917.5)	38 867.4 (41 944.7)	42 623.6 (40 162.3)
m_r	157 698.8 (118 045.0)	48 310.3 (74 366.5)	46 959.7 (42 903.9)	50 603.2 (43 382.9)

Table 5.6.: Coherence measures with the best ranks if one dataset has been left out.

Corpus left out	Coherence $\mathcal{P}$	$\mathcal{S}$	$\mathfrak{M}$	$\mathfrak{A}$	Average rank	Standard deviation
20NG	$(\mathcal{P}_{sw(110)}, S^{any}_{any}, \tilde{m}_{\text{cos},m_{nlr},1}, a_m)$				2 985.6	4 333.3
Genomics	$(\mathcal{P}_{sw(290)}, S^{any}_{any}, \tilde{m}_{\text{cos},m_{nlr},1}, a_m)$				2 188.0	2 566.7
NYT	$(\mathcal{P}_{sw(120)}, S^{one}_{all}, \tilde{m}_{\text{cos},m_{nlr},1}, a_a)$				3 219.7	4 542.1
RTL-NYT	$(\mathcal{P}_{sw(120)}, S^{one}_{all}, \tilde{m}_{\text{cos},m_{nlr},1}, a_a)$				3 191.7	4 558.1
RTL-Wiki	$(\mathcal{P}_{sw(120)}, S^{one}_{all}, \tilde{m}_{\text{cos},m_{nlr},1}, a_a)$				3 193.1	4 557.3
Movie	$(\mathcal{P}_{sw(70)}, S^{one}_{all}, \tilde{m}_{\text{cos},m_{nlr},1}, a_a)$				650.5	2 640.6

Table 5.7.: Coherence runtime results on the NYT dataset and the Wikipedia reference corpus in seconds.

Name	Runtime	Std. dev.
$\mathcal{C}_{V2}$	302.1	2.5
$\mathcal{C}_{P}$	305.2	2.4
$\mathcal{C}_{\text{UMass}}$	6.3	0.1
$\mathcal{C}_{\text{one-any}}$	6.3	0.1
$\mathcal{C}_{\text{UCI}}$	307.9	1.8
$\mathcal{C}_{\text{NPMI}}$	310.2	2.1
$\mathcal{C}_{\text{cos}}$	288.2	1.6

than the best performing coherence $\mathcal{C}_{V2}$. Thus, there are many variants of $\mathcal{C}_{V2}$ that perform well as long as they use a sliding window with a large window size (≥ 20). We confirm this by generating *leave one out averages*, i.e., we calculate the average ranks using only five of the six datasets. Table 5.6 shows that independently from the dataset left out, coherence measures that are very similar to $\mathcal{C}_{V2}$ achieve the best average ranks.

5.3.4 Runtimes

Next to the effectiveness of the coherence measures, we are interested in their efficiency. To this end, we measure the runtimes of all coherence measures.[23] For this experiment, we use the 100 topics of the NYT dataset and the Wikipedia as reference corpus. We use the coherence measures of Table 5.2 to calculate coherence values for all topics and measure the runtime. We repeat this experiment five times with a random order of the coherence measures to reduce the influence of the order on the results. Table 5.7 shows the overall time it takes to calculate the coherence values for all 100 topics for each of the coherence measures.

For the runtime of a coherence measure, the most important component is the probability estimation. The fastest estimation is the boolean document. It needs only 6.2s to retrieve all necessary probability values. The boolean paragraph and the boolean sentence based estimation methods need 15.5s and 62.5s, respectively. Both suffer from the fact that there are much more paragraphs and sentences than single documents. However, they are still faster than the window-based approaches since the reference corpus can be divided into paragraphs or sentences while preprocessing the corpus. In contrast, both window-based estimation methods have the highest

[23] We use a single machine with an Intel Core i5-7300U, 2.60GHz and 16 GB RAM for this experiment. The software was used in a sequential setup, i.e., it did not make use of parallel processing.

Table 5.8.: Different segmentation schemes, the number of subset pairs S_i they contain ($|S|$), and examples of their influence on runtimes of confirmation measures and aggregations when calculating the coherence values for all 100 topics of the NYT dataset. All values in seconds.

Name	$\lvert S\rvert$	$\tilde{m}_{\cos,m_{nlr},1}$		a_a	
		Runtime	Std. dev.	Runtime	Std. dev.
S^{set}_{set}	1	0.041	0.005	0.003	0.000
S^{one}_{all}	n	0.242	0.037	0.004	0.000
S^{one}_{set}	n	0.080	0.006	0.004	0.000
S^{one}_{pre}	$\frac{n(n-1)}{2}$	0.168	0.027	0.008	0.000
S^{one}_{suc}	$\frac{n(n-1)}{2}$	0.166	0.005	0.008	0.000
S^{one}_{one}	$n(n-1)$	0.298	0.074	0.013	0.000
S^{one}_{any}	$n(2^{(n-1)}-1)$	38.979	2.716	0.592	0.001
S^{any}_{any}	$\sum_{i=1}^{n-1}\left(\binom{n}{i}(2^i-1)\right)$	196.660	14.574	6.616	0.038

runtimes. This is caused by the need of retrieving the single positions of the words inside the documents to check whether these words are within the same window. The context-window-based approach needs 258.9s to retrieve all counts with a window size of 5. The sliding window needs 275.5s and 271.7s with a window size of 10 and 110, respectively.[24] We conclude that the number of windows that are processed has an influence on the runtime. The context-window-based approach has to take the lowest number of windows into account, since the number of windows is directly bound to the number of occurrences of the top words within the documents. The sliding window approach has to take a larger number of windows into account if the windows are small. Hence, the approach is slightly faster in our evaluation if the window size is large. Other parameters for the runtime of all probability calculation are 1) the number of topics that have to be evaluated, 2) the number of top words per topic (n), and 3) the size of the reference corpus.

Another important component is the segmentation. While the segmentation of a specific topic is very fast, it controls the number of confirmation values that have to be calculated. Thus, it has an impact on the time needed by the confirmation measure and the aggregation component. We measure the impact on variations of the C_{V2} coherence measure that make use of different segmentations. Table 5.8 shows the number of subset pairs S_i that the different segmentations create and the measured influence of this number on the runtime of confirmation measures

[24] The difference between these runtimes is significant. We used a Wilcoxon signed-rank test with a significance threshold of 0.05.

and aggregations on the NYT dataset. The table shows that the segmentations have an increasing complexity up to an exponential complexity for S^{one}_{any} and S^{any}_{any}. The measured runtimes of the aggregation follow exactly this increasing complexity. However, the measured runtime of the confirmation measure has an intermediate peak for S^{one}_{all} with a higher runtime than S^{one}_{pre} and S^{one}_{suc} although the latter have a higher number of subset pairs. This effect is caused by the need to calculate a single context vector that represents the W'' subset. While S^{one}_{pre} and S^{one}_{suc} lead to W'' subsets with a single element for which a vector is already available, the S^{one}_{all} segmentation leads to W'' subsets of size $n-1$, which is also different for each of the top words. Hence, additional time is consumed to calculate the vector that represents this set. However, the measured runtimes show that in practice, the influence of the majority of segmentations on the overall runtime is low compared to the probability estimation.

5.3.5 Application in LODCat

As described in Section 5.2.2, we infer several topic models with different numbers of topics. To choose the best model, we rely on the best performing topic coherence measures $\mathcal{C}_{V2}$ and $\mathcal{C}_{P}$. We define the human understandability of a topic model as the understandability of its topics. Hence, we can calculate the quality of a model as the average coherence of its topics. In addition, we use these coherence measures to identify low-quality topics that should not be shown to the user since it is unlikely that a user will find them helpful.

5.4 Evaluation of LODCat

We evaluate LODCat in a setup close to a real-world scenario. That means that we start with the English Wikipedia as corpus and process more than 600 thousand RDF datasets using LODCat following the steps described in Section 5.2. The evaluation can be separated into the following three consecutive experiments:

1. The first experiment comprises the generation of the topic model that will be used for the further process. This includes the generation of topic models, the selection of the best model and the application of the topic labeling algorithm.
2. The second experiment uses the best topic model generated within the first experiment and more than 600 thousand RDF datasets as input. It comprises

the transformation of the RDF datasets and the assignment of topics from the topic model.

3. The third experiment finally evaluates the assignment of topics to the RDF datasets based on a user study.

5.4.1 Datasets

For our evaluation, we use two types of data—a reference corpus to generate the topic model and the set of RDF datasets, which should be represented in a human-interpretable way. We use the English Wikipedia as reference corpus.[25] We preprocess the dump file by removing Wikimedia markup from the single articles. After that, we remove redirect articles and handle each remaining article as an own document. Each document is preprocessed as described in Section 5.2.1.[26] From the created set of documents, we derive all word types and count their occurrence. Then, we filter the word types by removing 1) common English terms based on a stop word list, and 2) all word types that occur in more than 50% of the documents or 3) in less than 20 documents.[27] From the remaining word types, we select the 100 000 word types with the highest occurrence counts and remove all other from the documents. After that, we remove empty documents and randomly sample 10% of the remaining documents. Finally, we get a corpus with 619 475 documents and 190 million word tokens.

We gather 623 927 real-world RDF datasets from the LOD Laundromat project [30].[28] These are the datasets that we want to represent in a human-interpretable way. This data has a large overlap with the LOD-a-lot dataset of Fernandez et al. [95] since both rely on cleaned RDF datasets of the LOD Laundromat. However, the number of RDF datasets we use is slightly smaller and the datasets are separated from each other. Using one large dataset like the LOD-a-lot dataset would not fit to our use case. Figure 5.7 shows the size of the RDF datasets. The majority of datasets in the figure have between 100 and 10 000 triples. The largest dataset has 43 million triples. The sum of all triples in these RDF files is 3.7 billion.[29]

[25] We use the dump of the English Wikipedia from September 1st 2021.

[26] In our current implementation, we use the Stanford CoreNLP library [175].

[27] The stop word list can be found at `https://github.com/dice-group/lodcat/blob/develop/lodcat.model/src/main/resources/stopwords.txt`; last accessed on 06.08.2022.

[28] We downloaded the datasets in January 2018.

[29] Note that we do not deduplicate the triples across the datasets.

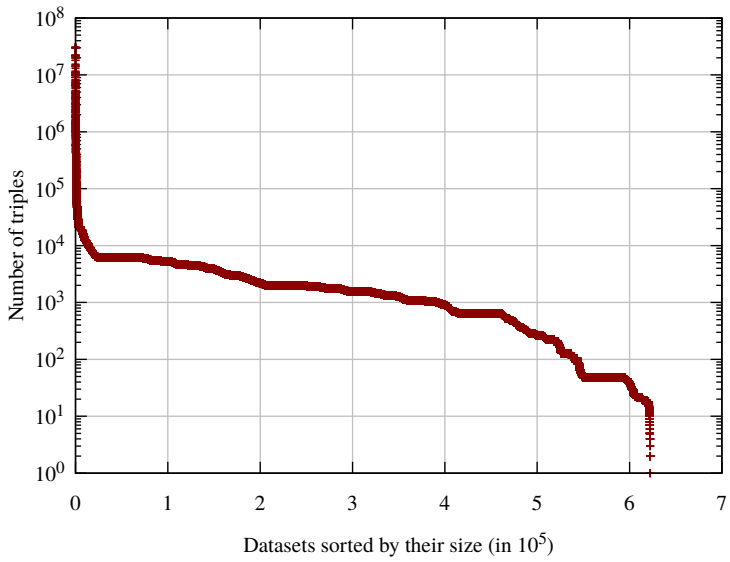


Figure 5.7.: The size of the RDF datasets.

5.4.2 Setup

Experiment I

In the first experiment, we infer the topic models based on the English Wikipedia corpus. As described in Sections 5.2.2 and 5.3.5, we infer several models with different numbers of topics.[30] For this evaluation, we use $\varrho = \{80, 90, 100, 105, 110, 115, 120, 125, 135\}$. We generate three models for each number of topics and configure the inference to use hyper parameter optimization for both hyper parameters. The single models are rated by calculating the average quality of their topics using the $\mathcal{C}_{V2}$ and $\mathcal{C}_P$ topic coherence measures. For each coherence measure, we rank the models based on their average coherence value. We choose the model that achieves the best rank on average for both coherence measures.

We further analyze the chosen topic model in more detail with respect to the quality of the single topics. As described in Section 5.3.5, the topic coherence values can be used to distinguish topics with high and low coherence values. To this end, we define a topic quality threshold. All topics with a minimum $\mathcal{C}_{V2}$ value of 0.125 and a minimum $\mathcal{C}_P$ value of 0.25 are high quality topics while all topics with lower coherence values are low quality topics. In practice, the latter are topics that would not be shown to the user.

[30] We use the Gensim library for the inference [223]. `https://radimrehurek.com/gensim/index.html`; last accessed on 06.08.2022.

In addition, we use the models generated within this experiment to compare our coherence-based results with the two approaches described in Section 2.2.3, which have been proposed by related work to identify a good number of topics.

Experiment II

Based on the best topic model created in the first experiment, we process each of the 623 927 RDF datasets by LODCAT as described in Section 5.2. During this step, we remove datasets that lead to an empty document. The result of LODCAT comprises a document for each RDF dataset and a topic distribution based on the used topic model. After that, we analyze the results by looking at the topics that have been assigned to the documents. Let D_{RDF} be the corpus that is created by LODCAT based on the given RDF datasets and let D_{Wiki} be the corpus that has been used to generate the topic model in the first experiment. Let $\zeta_{i,k}$ be the count of word tokens in the i-th document d_i that have been assigned to the k-th topic as defined in Section 2.2.2. We define the measure $\mathfrak{b}(k, D_{\text{RDF}}, D_{\text{Wiki}})$ to measure the importance of the k-th topic for corpus D_{RDF} in comparison to its importance for corpus D_{Wiki} as follows:

$$\mathfrak{b}(k, D_{\text{RDF}}, D_{\text{Wiki}}) = \frac{\left(\frac{\sum_{i=1}^{|D_{\text{RDF}}|} \zeta_{i,k}}{\sum_{j=1}^{\varrho} \sum_{i=1}^{|D_{\text{RDF}}|} \zeta_{i,j}} \right)}{\left(\frac{\sum_{i=1}^{|D_{\text{Wiki}}|} \zeta_{i,k}}{\sum_{j=1}^{\varrho} \sum_{i=1}^{|D_{\text{Wiki}}|} \zeta_{i,j}} \right)}. \tag{5.58}$$

The importance of a topic is expressed as the number of word tokens that are assigned to this topic while determining the topic distributions for the corpus' documents. Assuming that D_{Wiki}, i.e., the English Wikipedia, represents a broad, general set of topics, this measure can help to identify topics that might be over or underrepresented within the D_{RDF} corpus, i.e., within the RDF datasets.

Experiment III

Finally, we evaluate the assignment of the topics to the datasets. Chang et al. [63] propose the topic intruder experiment to evaluate whether the assignment of topics to a document is good. They determine the top topics of a document (i.e., the topics that have received the highest probabilities for the document) and insert a

randomly chosen topic from the same topic model that is not one of the document's top topics. This randomly chosen topic is called intruder topic. After that, volunteers are given the created list of topics and the document, and are asked to identify the intruder. The more often the intruder is successfully identified, the better is the topic assignment of the topic model. We use the same approach to evaluate whether a topic model can assign meaningful topics to an RDF dataset.

We sample 60 datasets that have more than 100 and less than 10 000 triples. For each of the sampled datasets, we derive the three topics with the highest probability. Based on the dataset content and the quality of their top topics, we choose 10 datasets that 1) have at least two good topics among the top three topics, 2) have a good quality topic as highest ranked topic, 3) have a content that can be understood without accessing further sources, and 4) have not exactly the same top topics as the already chosen datasets. For each chosen dataset, we sample an intruder topic from the set of high quality topics that are not within the top three topics of the dataset.

We create a questionnaire with 10 questions. Each question gives the link to one of the chosen datasets and a list of topics comprising the top topics of the dataset and the intruder topic in a random order. 5 chosen datasets have three good top topics while the other 5 datasets have one top topic with a low coherence value. We remove the topics with the low values. Hence, 5 question comprise 4 topics and the other 5 questions have 3 topics from which a user should choose the intruder topic.[31] For the questionnaire, the topics are represented in the human-readable way described in Section 5.2.3, i.e., with their label and their top words. The participants of the questionnaire are encouraged to look into the RDF dataset. However, they should not include further material. A user may answer all 10 questions. However, all answers are weighted equally even if a user only gave an answer for one or two questions. We send this questionnaire to several mailing lists to encourage experts and experienced users of the Semantic Web to participate.

Following Chang et al. [63], we calculate the topic log odds to measure the agreement between the topic model and the human judgments that we gather with our questionnaire. Let θ_i be the topic distribution of i-th document d_i. Let $\theta_{i,k}$ be the probability of the k-th topic for document d_i. Let $\mathfrak{U}_i = \{\mathrm{u}_{i,1}, \ldots\}$ be the bag of all user answers for document d_i, i.e., the j-th element is the id of the topic that the j-th user has chosen as intruder topic for this document. Let x_i be the id of the real intruder topic for document d_i. Chang et al. [63] define the topic log odds ϑ as the

[31] The 10 questions and their answers can be found in Section A.5 of the appendix.

average difference between the probabilities of the chosen intruder topics compared to the real intruder topic:

$$\vartheta(\theta_i, \mathfrak{U}_i, \mathrm{x}_i) = \frac{1}{|\mathfrak{U}_i|} \sum_{j=1}^{|\mathfrak{U}_i|} \Big(\log(\theta_{i,\mathrm{x}_i}) - \log(\theta_{i,\mathrm{u}_{i,j}}) \Big) \ . \tag{5.59}$$

A perfect agreement between the human participants and the topic model would lead to $\vartheta = 0$. In practice, this is only reached if all participating volunteers always find the correct intruder topic.[32]

5.4.3 Results

Experiment I

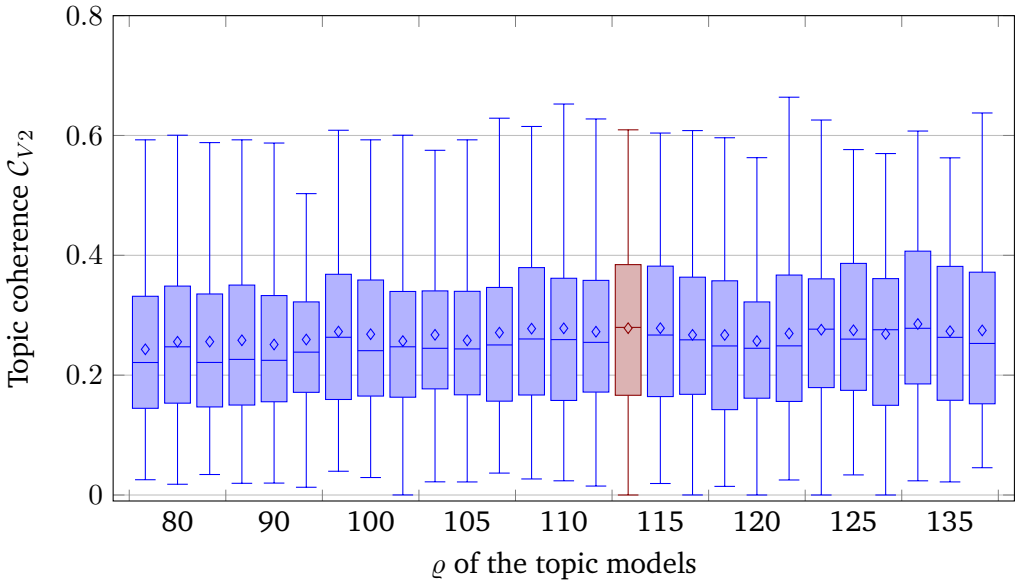


Figure 5.8.: Boxplots for $\mathcal{C}_{V2}$ coherence values of different topic models with different numbers of topics. The marked topic model is the model chosen for further processing. The diamond shaped points mark the arithmetic means.

Figures 5.8 and 5.9 show the topic coherence values of the topic models generated with varying values for ϱ, respectively. We rank the topic models according to their coherence score for $\mathcal{C}_{V2}$ and $\mathcal{C}_P$, respectively, and assign the sum of the ranks to the topic models. We pick the model with the best overall ranking. This model is ranked

[32] In theory, this is not necessary. It is sufficient if the human judges choose topics that have the same probability $\theta_{i,j}$ as the intruder topic. However, this case does not occur often in practice since the number of top topics used for this experiment is low and these topics typically have a higher probability for the document than the intruder topic.

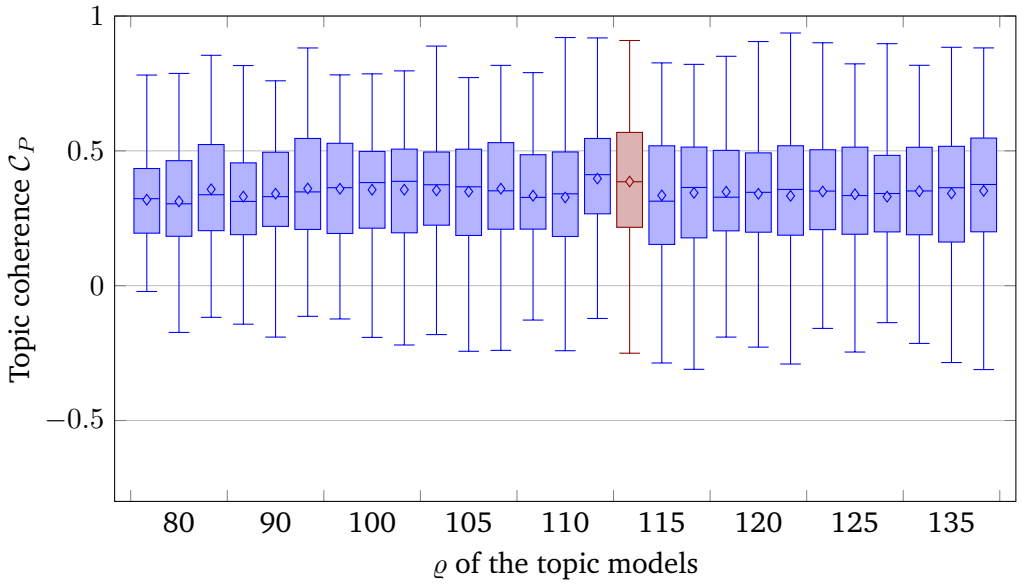


Figure 5.9.: Boxplots for $\mathcal{C}_P$ coherence values of different topic models with different numbers of topics. The marked topic model is the model chosen for further processing. The diamond shaped points mark the arithmetic means.

as second best model by both coherences measures and has $\varrho = 115$. The chosen topic model is marked in Figures 5.8 and 5.9. The latter figure, shows another topic model with $\varrho = 110$ next to the chosen model with a higher average coherence. However, this model is ranked as 11th best model by the $\mathcal{C}_{V2}$ coherence measure. A similar result achieves the model that got the best $\mathcal{C}_{V2}$ coherence value. For the remainder of this section, we will focus our evaluation on the chosen model.

The chosen topic model comprises 115 topics. Figures 5.10 and 5.11 show the coherence values of the topics for both coherence measures. Table 5.9 shows the top words of 10 example topics. These topics have been chosen based on their $\mathcal{C}_{V2}$ coherence values. They represent the topics with the 5 highest and 5 lowest coherence values. While the first 5 topics seem to focus on a single topic the topics with the low coherence scores comprise words that seem to have no strong relation to each other.

We compare our approach to choose the best model with the approaches described in Section 2.2.3. Figure 5.12 shows the $\mathbb{P}(D|\Phi)$ values for the generated models. Griffiths et al. [110] propose to choose the model with the highest value. However, our results suggest that this approach prefers models with many topics. This observation is in line with the results of Wallach et al. [300], who already showed that an optimized asymmetric α hyper parameter is more robust against too many topics. Figure 5.13 shows the values of the $\mathcal{A}$ measure proposed by Arun et al. [19]. The

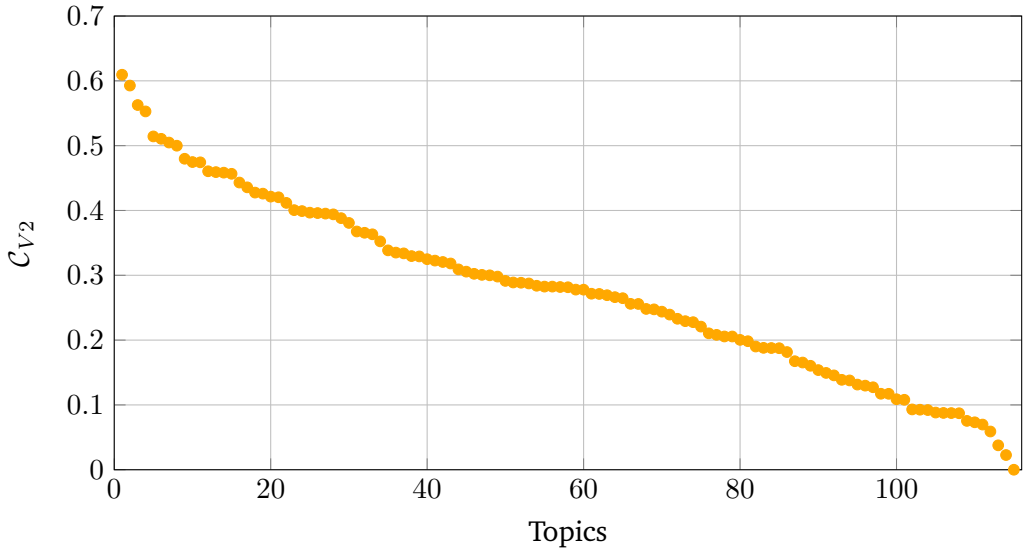


Figure 5.10.: Topics of the best performing model sorted by their $\mathcal{C}_{V2}$ coherence value.

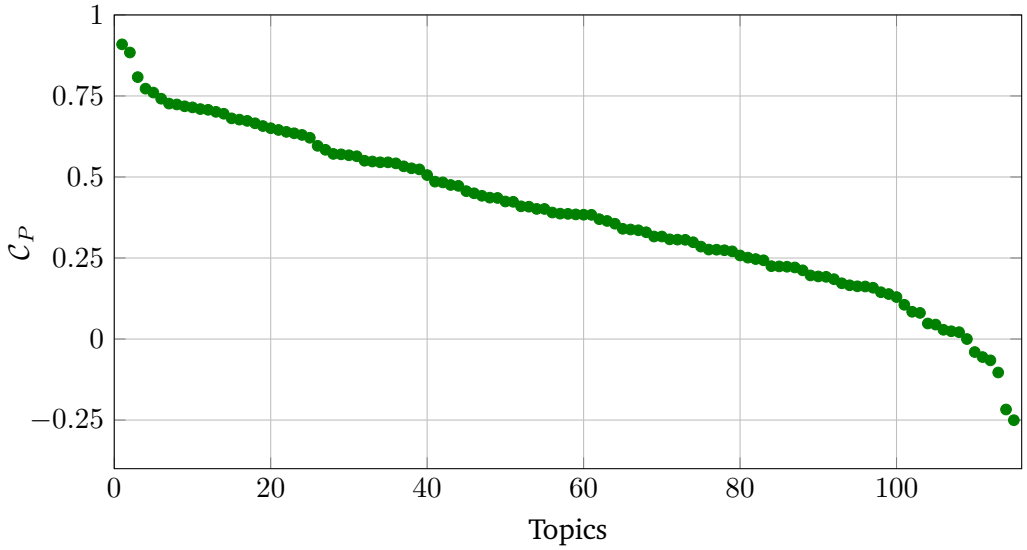


Figure 5.11.: Topics of the best performing model sorted by their $\mathcal{C}_P$ coherence value.

authors predict that with an increasing number of topics the values of $\mathcal{A}$ should decrease until a minimum is reached before the measure's value increases again. The model that has the minimum $\mathcal{A}$ should be chosen. In the figure, we cannot see the described behavior. This could be caused by several factors. Either our grid of ϱ values is too coarse-grained, or the minimum is outside of our search range. A similar experiment with a smaller subset of our reference corpus suggests that the minimum described by Arun et al. [19] is achieved by a model that has much less topics. The detailed results of this experiment can be found in Section A.4

Table 5.9.: The top words of the 5 topics of the best performing topic model with the highest and lowest $\mathcal{C}_{V2}$ values, respectively.

$\mathcal{C}_{V2}$	$\overline{W}$
0.60942	canadian, canada, quebec, ontario, montreal, toronto, ottawa, nova, scotia, alberta
0.59273	album, song, release, band, music, chart, record, single, track, records
0.56269	age, population, household, female, city, male, family, census, average, year
0.55282	chinese, china, singapore, li, wang, shanghai, chen, beijing, hong, zhang
0.51424	league, club, player, football, season, cup, play, goal, team, first
0.06969	rank, time, men, advance, event, final, result, athlete, heat, emperor
0.05900	use, language, word, name, form, one, english, see, greek, two
0.03767	use, system, one, number, two, function, set, space, model, time
0.02269	use, health, may, child, include, provide, would, act, make, public
0.00000	j., a., m., c., r., s., l., e., p., d.

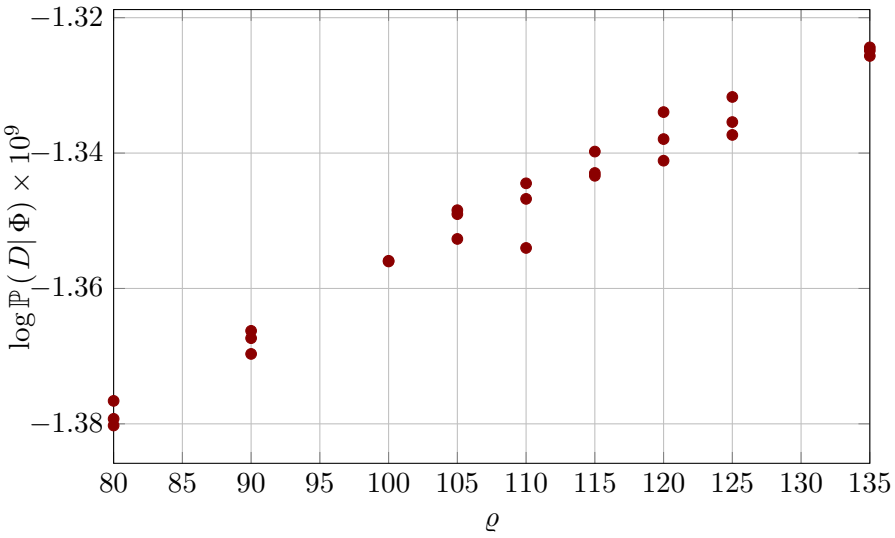


Figure 5.12.: Values of $\log(\mathbb{P}(D|\Phi))$ calculated for the generated models.

of the appendix. We can conclude that both measures for choosing the number of topics do not seem to lead to the same model choice as our coherence-based approach. However, both measures are solely based on the idea of comparing the generated models with an expected, statistical behavior. Griffiths et al. [110] propose to measure whether the model assigns a high probability to the reference corpus that has been used for the inference. Arun et al. [19] base their measure on the interpretation of LDA as a non-negative matrix factorisation and that the created matrices should have the same sum of topic assignments per topic. However, Chang et al. [63] show that achieving a good performance in measures similar to the two aforementioned measures does not have to correlate with high quality topics. Hence, our approach seems to fit better to the goals of LODCat than those suggested by Griffiths et al. [110] or Arun et al. [19].

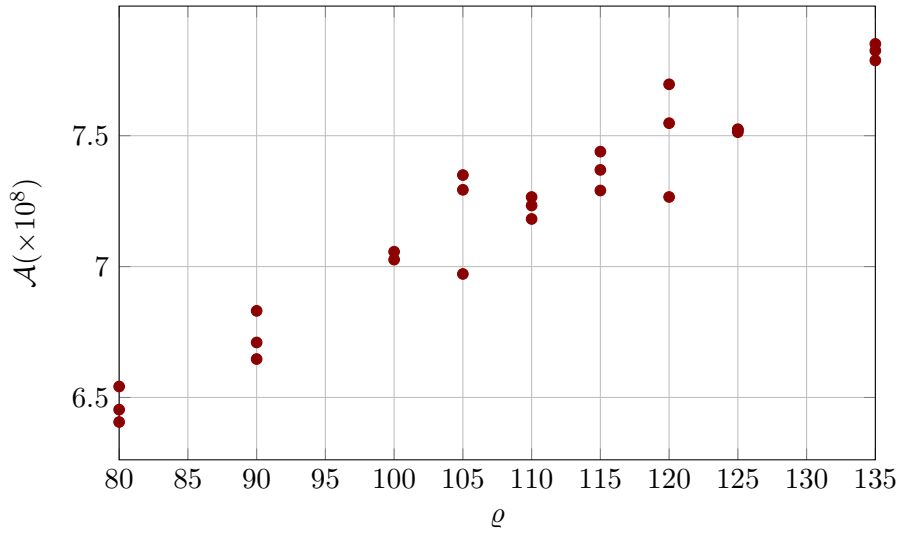


Figure 5.13.: Values of the measure $\mathcal{A}$ calculated for the generated models.

Experiment II

We use the best topic model of the first experiment to process the RDF datasets with LODCat. We received topics for 561 944 datasets. After generating the topic distributions for the documents created from the RDF datasets, we analyze these distributions. For each dataset, we determine its main topic, i.e., the topic with the highest probability for this dataset. Figure 5.14 shows the number of datasets for which the single topics are the main topic. The figure shows that a single topic covers more than 410 thousand datasets. It also shows the prominence of some low-quality topics. Since these topics would not be shown to the user, we repeated this analysis and included only high-quality topics. The results are plotted in Figure 5.15. It shows the same concentration of a high number of datasets on a single topic. We also compare the importance of the topics for the documents generated from the RDF datasets in comparison to their importance for the Wikipedia corpus. Figure 5.16 shows the results of this comparison. We observe that a small number of topics has values above 1.0. The highest point shows that for one of the topics the percentage of tokens that are assigned to it is 17 times higher within the RDF corpus than in the Wikipedia corpus. On the contrary, a large number of topics is underrepresented in the RDF corpus.

Table 5.10 shows the 5 topics that have the highest values in Figure 5.15, i.e., the 5 topics that are most often the top topic of a dataset. We can see that a weather-related topic covers roughly 90% of the datasets to which LODCat could assign topics. The next biggest topics are transportation- and car-related topics and each of

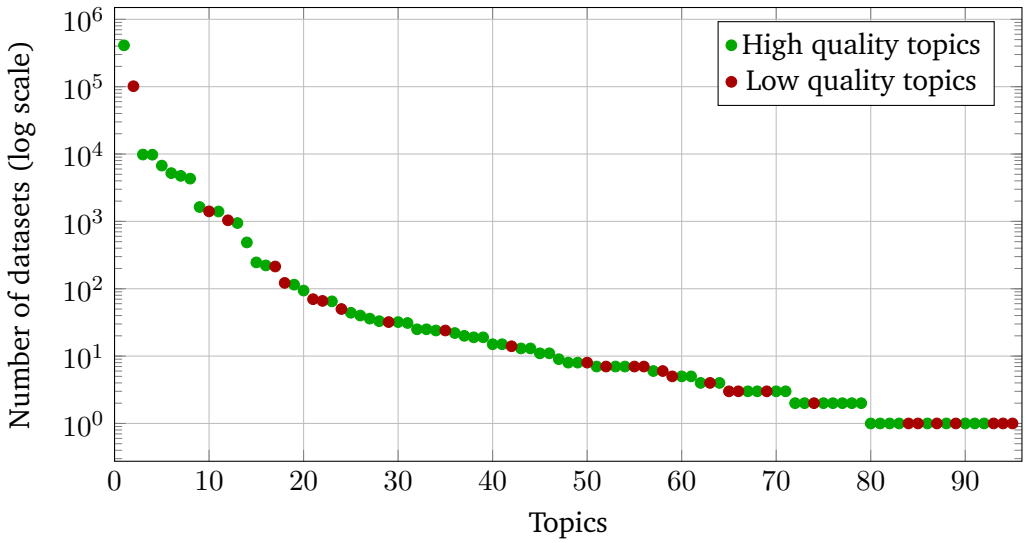


Figure 5.14.: Number of datasets per topic for which this topic has the highest probability. Topics with no datasets have been left out.

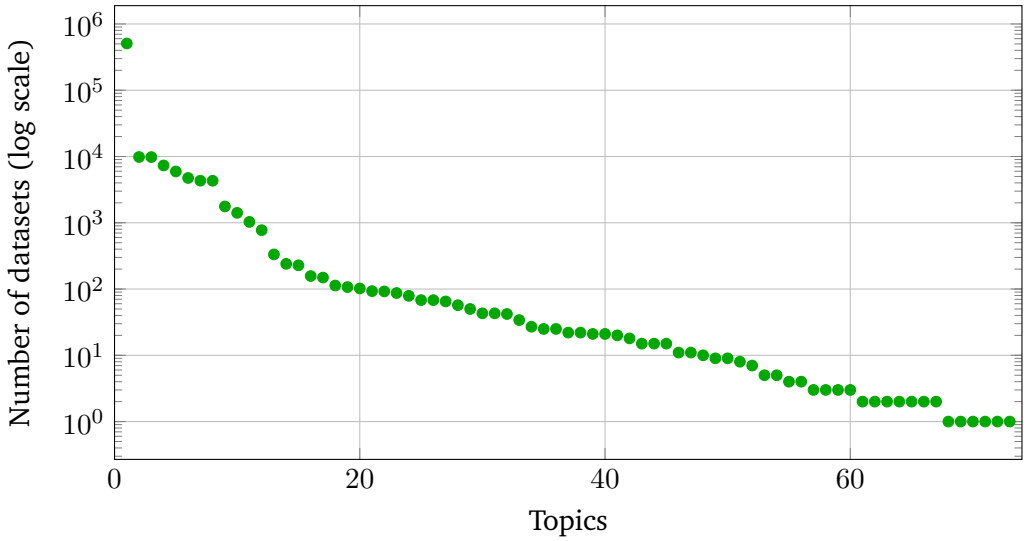


Figure 5.15.: Number of datasets per topic for which this topic has the highest probability after removing topics with a low coherence score. Topics with no datasets have been left out.

Table 5.10.: The topics which are the top topics for most of the datasets.

Id	Datasets	$\overline{W}$
1	508 095	water, storm, wind, tropical, nuclear, temperature, hurricane, damage, cause, system
2	9 828	station, road, route, line, street, bridge, railway, city, highway, east
3	9 794	car, engine, model, vehicle, first, use, point, motor, design, safe
4	7 328	use, system, software, user, datum, computer, include, information, support, service
5	5 946	airport, international, brazil, portuguese, são, romanian, portugal, brazilian, language, romania

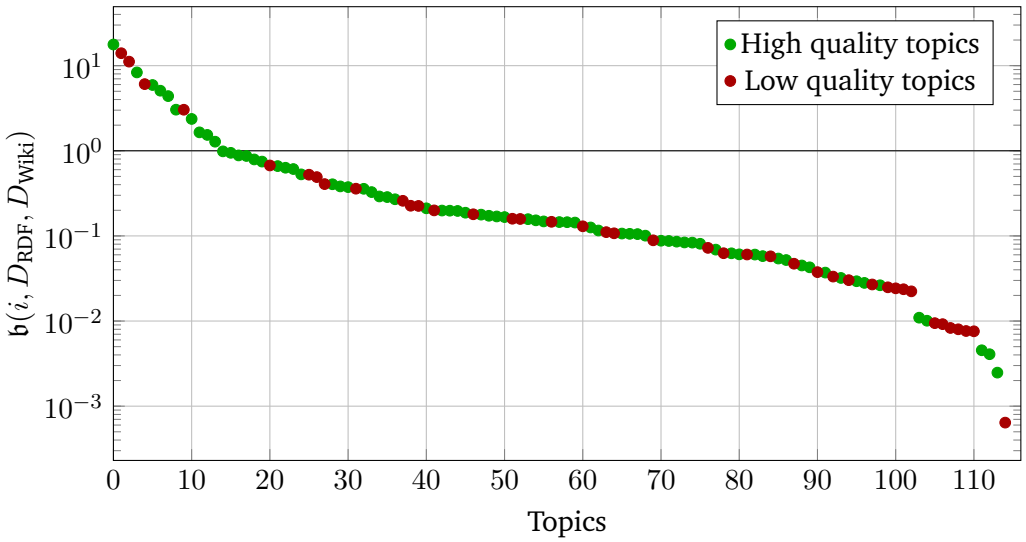


Figure 5.16.: Comparison of the importance (i.e., the number of tokens assigned) of a topic in the RDF dataset divided by the importance the topic has in the Wikipedia corpus. Values > 1 indicate an increase of importance while a value < 1 means that the topic is less important.

them covers nearly 10 thousand datasets. They are followed by a computer- and a travel-related topic.

We further analyze the RDF datasets with respect to the claim that the majority of them is related to weather. We analyze the namespaces that are used within the RDF datasets and count the number of datasets in which they occur. Figure 5.17 shows the result of this analysis for all 623 thousand RDF datasets. In the lower right corner of the figure, we can see that there is only a small number of namespaces that are used in many datasets. Table 5.11 shows the 12 namespaces that occur in more than 100 thousand datasets. The most often used namespace is the `rdf` namespace, which is expected. However, the namespaces on position 2–4 occur in more than 450 thousand RDF datasets. These three namespaces belong to datasets with sensor data described by Patni et al. [214]. A further search revealed that the datasets mainly contain weather data [213]. These datasets also make use of the fifth namespace from Table 5.11. The sixth namespace is the Data Cube namespace that is used to described statistical data in RDF [73]. This namespace occurs often together with the remaining namespaces (7–12). They occur in datasets that origin from the Climate Change Knowledge Portal of the World Bank Group.[33] These dataset contain climate data, e.g., the temperature for single countries and their forecast with respect to different climate change scenarios. We summarize that our analysis

[33] `https://climateknowledgeportal.worldbank.org/`; last accessed on 08.08.2022.

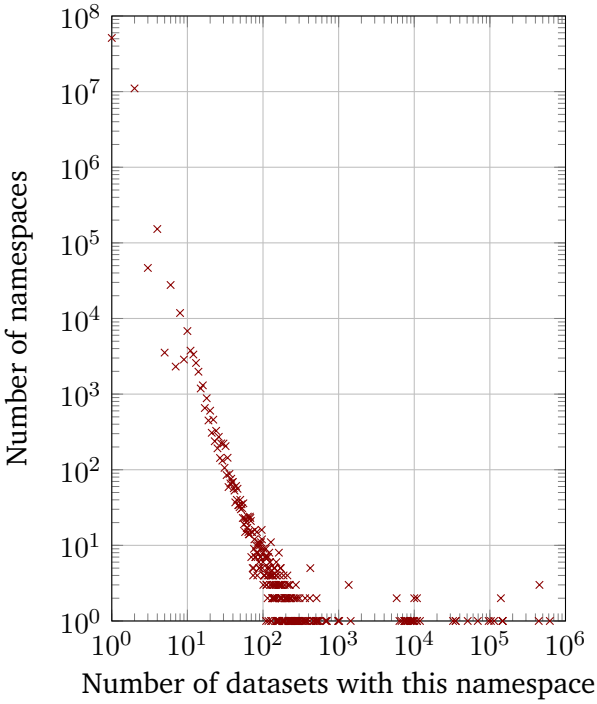


Figure 5.17.: The number of namespaces (y-axis) that occur in a number of datasets (x-axis).

shows that the majority of the datasets contain sensor data and statistical data that are related to weather.

Figure 5.17 gives another insight. The point in the left upper corner of the plot shows that there are more than 51 million namespaces that occur only in one out of the 623 thousand RDF datasets. Additional 11 million namespaces occur only in two datasets. These are already 99% of the 62.5 million namespaces that occur within the 623 thousand RDF datasets. This large number of rare namespaces reflects one of the difficulties that users may face when they try to identify the topic of an RDF dataset without using LODCat. There are a lot of namespaces that a user would have to look up and even if a user understands the topic of the namespace, it is not very likely that it will occur again in another dataset. Hence, a user may try to rely on namespaces that occur often. However, only 4 out of the 12 namespaces in Table 5.11 are dereferencable. These four namespaces (1, 5, 6, and 8 in the table) are generic. The other namespaces are more topic-related but cannot simply be opened since non of the domains exists at the time we carry out these experiments. A user would have to invest additional effort to find information about them. This underlines the need for a tool like LODCat that helps the user to avoid this effort.

Table 5.11.: The namespaces that occur in more than 100 000 datasets.

ID	Namespace IRI	Datasets
1	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`	620 653
2	`http://knoesis.wright.edu/ssw/ont/weather.owl#`	452 453
3	`http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#`	452 453
4	`http://knoesis.wright.edu/ssw/`	452 453
5	`http://www.w3.org/2006/time#`	442 719
6	`http://purl.org/linked-data/cube#`	147 731
7	`http://worldbank.270a.info/property/`	147 348
8	`http://purl.org/linked-data/sdmx/2009/dimension#`	147 305
9	`http://worldbank.270a.info/dataset/world-bank-climates/`	139 865
10	`http://worldbank.270a.info/classification/variable/`	139 865
11	`http://worldbank.270a.info/classification/scenario/`	114 064
12	`http://worldbank.270a.info/classification/percentile/`	103 202

Experiment III

Our questionnaire received 225 answers from 65 participants.[34] 20 participants went through all questions while the remaining 45 participants gave answers for a subset of questions. Figure 5.18 shows the results. The left side of the figure summarizes the results for the two groups of questions—those with 3 and 4 topics, respectively. The right side of the figure shows the detailed results for each of the questions. The plot shows that in the majority of cases, the intruder was successfully identified by the participants. The results look slightly different for datasets 4 and 5. In both cases, the third topic is not strongly related to the dataset and has been chosen quite often as intruder. However, since the first and second topic have been chosen much less often for these datasets, the result shows that the ranking of the topics make sense, i.e., the participants were able to identify the first two topics have a relation to the given dataset.

Figure 5.19 shows a box plot for the topic log odd values that have been measured for the single documents. The average value across the 10 datasets is -1.23 with dataset 4 getting the worst value. This value is visible as an outlier on the left side of the plot. This result is comparable to the results Chang et al. [63] present for various topic modeling models on two different corpora. This confirms our finding that the human-readable topics fit to the RDF datasets to which they have been assigned. However, the experiment setup comes with two restrictions that have to be made with respect to this result. First, RDF datasets are mainly created to be processed by

[34] We use LimeSurvey for the questionnaire (`https://www.limesurvey.org/`; last accessed on 08.08.2022.). The questionnaire allows users to skip questions. These skipped questions are not taken into account for the number of answers.

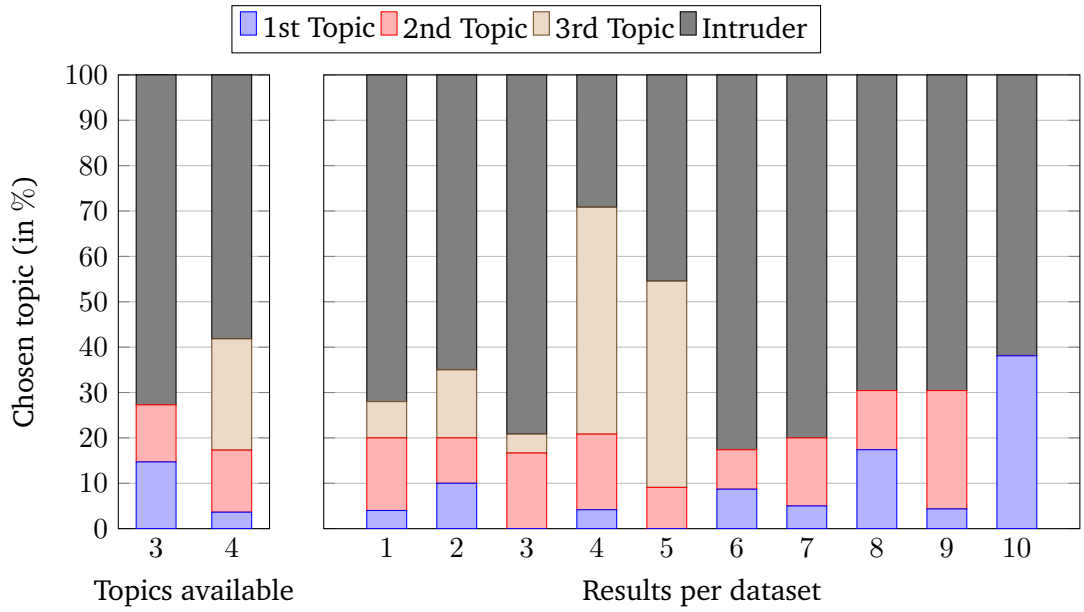


Figure 5.18.: Results of the questionnaire. Left: Average amount of topics that where chosen as intruder topic by the survey participants. Right: amount of topics that where chosen for each of the single datasets.

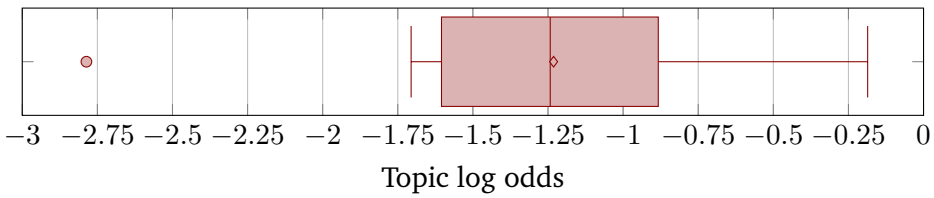


Figure 5.19.: The topic log odds ϑ per document. Values close to 0 are better. The diamond shaped point marks the arithmetic mean.

machines. We manually chose the datasets for this experiment with the requirement that the participants of the questionnaire have to be able to easily understand the content of the chosen datasets. This may have introduced a bias. However, it can be assumed that the results would be less reliable if the datasets would have been selected randomly since the experiment setup suggested by Chang et al. [63] relies on the assumption that the participants understand the target object to which the topics have been assigned (in our case, the RDF dataset). Second, we made use of topic coherence measures to filter low quality topics and we chose datasets that have at least two high quality topics within their top-3 topics. It can be assumed that the topic loss odd values would be lower if we would have included low quality topics, since they are less likely interpretable by humans. However, a dataset that has mainly low quality topics assigned as top topics could cause issues in a user application relying on LODCat since no human-interpretable description of the dataset could be provided. We measure the impact of this issue by counting the number of datasets that have not a single high quality topic within their top-3 topics. We find that out of the 561 944 RDF datasets, to which LODCat could assign topics, only 220 datasets have this problem. Hence, the filtering of low quality topics seems to have a minor impact on the number of RDF datasets for which LODCat is applicable.

5.5 Conclusion

The aim of this chapter was twofold. First, we presented LODCat—an approach to support the exploration of the Data Web based on human-interpretable topics. With this approach, we ease the identification of RDF datasets that might be interesting to a user since they neither have to go through all available datasets nor need to read through the single RDF triples. Instead, LODCat provides the user with human-interpretable topics that are automatically derived from a reference corpus and give the user an impression of a dataset's content. At the same time, our approach does not rely on manually created tags or classification systems and can be easily combined with existing explorative search engines or integrated into dataset portals.

Second, we presented Palmetto—a framework for topic coherence measures. Based on this framework, we evaluated 555 660 coherence measures and identified two new topic coherence measures that perform better than the previous state of the art.

Our future work has two main targets. First, the application of other topic modeling inference algorithms can be beneficial. At the moment, the user is provided with a sorted list of topics for each dataset. However, these topics have a flat hierarchy. Introducing a hierarchy of topics that can be used for a coarse or fine grained exploration could offer more opportunities to a user. Second, the application of the two coherence measures $\mathcal{C}_{V2}$ and $\mathcal{C}_P$, shows that they do not always agree with respect to the quality of single topics. Hence, a combination of these two measures might lead to a better performance.

6 Dataset Search for Linking

The Web of Data and the Linked Open Data Cloud have grown considerably in recent years. 1.5 billion Web pages with 82 billion embedded RDF triples and several thousands of RDF datasets can already be found online [22, 90, 220]. With the growth of the number of datasets available as well as the growth of their size comes the problem of effectively detecting not only the links between the datasets (as studied in previous works [200, 205]) but also of determining the datasets with which a novel dataset should be linked. A naive approach to linking these datasets would choose two datasets and check whether they can be linked with each other. Such an approach would lead to a quadratic number of pairwise comparisons with respect to the number of datasets. This is clearly impracticable when the effort entailed by the linking of two datasets is taken into account. Addressing the problem of finding relevant datasets for linking is however of crucial importance to facilitate the integration of novel datasets into existing Linked Data [34] as well as the discovery of relevant data sources in enterprise Linked Data [200] (**RG3**). Lopes et al. [170] name this the *dataset interlinking recommendation problem*. In this chapter, we address this problem and study the search for similar RDF datasets given an input dataset. In this context, we define two datasets as being similar if they cover the same topics and should thus be linked to each other. In particular, we aim to elucidate the question whether topic modeling (in particular LDA [43]) can be used to improve the search of similar datasets. To address this research question, we present six different approaches pertaining to how RDF datasets can be modeled for dataset search. We then compare these different modeling possibilities against the state of the art. Our findings are implemented into TAPIOCA [230]—a search engine that takes a description of a dataset and searches for topically similar datasets that could be candidates for link discovery. Our engine computes topics of datasets by analyzing their ontologies. It then uses these topics to map datasets to domains in a fuzzy manner. Based on this representation, TAPIOCA can compare the topic vector of an input dataset to datasets in its index so as to suggest topically similar datasets,

¶ Parts of this chapter have been published as conference article [230]. The author of this thesis is also the main author of the article and developed the main idea, designed and implemented major parts of the solution, and wrote the majority of the publication. A first attempt for GLISTEN is described within the master thesis of Kuhlmann [154]. The author of this PhD thesis defined the topic for said master thesis and was the advisor of the master student. Later, the approach has been further refined by the author of this thesis to reach the state described within this chapter.

which are assumed to be good candidates for linking. Note that we do not study the link discovery problem herein and address exclusively the search for data for linking under the assumption that datasets should be linked if they describe similar topics.[1]

We measure the effectiveness of TAPIOCA based on an intrinsic and an extrinsic evaluation [141]. The intrinsic evaluation uses a hand-crafted gold standard. Since the creation of such a gold standard is expensive, we also develop an external benchmark for the extrinsic evaluation of the dataset interlinking problem. Our new benchmark, dubbed GLISTEN, measures the effectiveness of a recommendation by linking the recommended dataset to the dataset for which it has been recommended. After that, the linked datasets are used as input to a system that solves an external task. The change in performance of the system fulfilling the external task is used as metric to measure whether the recommendation was good.

Our contributions within this chapter are as follows:

- We present six combinations of approaches for modeling data in RDF datasets that can be used for dataset search.
- We apply topic modeling to these combinations, compare them with state-of-the-art baselines, and show that topic modeling does lead to significant improvements over several baseline methods.
- We present GLISTEN—the first benchmark for dataset interlinking recommendation systems.

Since our work focuses on RDF datasets and these can be represented as knowledge graphs, we use the terms knowledge graph and dataset interchangeably. The rest of this chapter is structured as follows: First, we present other approaches related to our work in Section 6.1. In Section 6.2, our novel approach for a dataset recommendation engine is presented. We describe the proposed benchmark in Section 6.3 and subsequently evaluate TAPIOCA in Section 6.4. We conclude the chapter in Section 6.5.[2]

[1] The interested reader is referred to the survey of Nentwig et al. [198]. We also refer to the more recent work of Li et al. [166] who present an approach relying on embeddings.

[2] More information on TAPIOCA and the data we use for the evaluation can be found at `http://aksw.org/projects/tapioca`; last accessed on 14.08.2022. The detailed experiment results can be found at `https://hobbitdata.informatik.uni-leipzig.de/homes/mroeder/tapioca/`; last accessed on 14.08.2022.

6.1 Related Work

Link discovery is a task of central importance when publishing Linked Data [200]. While a large number of approaches have been devised for discovering links between datasets [205, 297], the task at hand is a precursor of link discovery and can be regarded as a similarity computation task. The usage of document similarities that are based on topic modeling is well known and have been widely studied in previous works, e.g., by Steyvers et al. [266]. Especially for information retrieval applications, topic modeling has been used for documents containing natural language. Buntime et al. [56] developed an information retrieval system that is based on a hierarchical topic modeling algorithm to retrieve documents topically related to a given query. Lu et al. [171] analyzed the effect of topic modeling for information retrieval. Their results show that while its performance is not good for a keyword search, it has a good performance for clustering and classification tasks in which only a coarse matching is needed and training data is sparse. Our results support the intuition underlying this chapter, i.e., that the task of retrieving similar linked datasets matches this task description.

The Semantic Web is already used for information retrieval tasks. For example, Hogan et al. [129] as well as Tummarello et al. [277] present approaches for Semantic Web search engines retrieving single entities and consolidated information about them given a keyword query. One of the problems that have to be solved for this task is the consolidation of retrieved entities. Since a single entity can have different IRIs in different datasets, the workflow of such a search engine has to have a consolidation step identifying IRIs mentioning the same entity. In both approaches, two resources are assumed to mention the same entity if 1) they are connected by an `owl:sameAs` property or 2) both resources have an Web Ontology Language (OWL) inverse functional property with the same value. The values of such inverse functional properties are typically assumed to be unique, e.g., an e-mail address. This problem is further studied by Herzig et al. [121]. These approaches differ from our dataset recommendation engine, since they cannot be used to identify topically similar datasets for linkage, because the entities must have been already linked—directly or indirectly by inverse functional properties. The aforementioned search engine proposed by Tummarello et al. [277] has an additional consolidation step summarizing properties that are assumed to describe the same fact. This summary is created by using the name of the property, i.e., the last part of its IRI. Additionally, the authors wrote that they want to use the labels of the properties in a future release of their search engine. This usage of labels or names of properties to decide

whether they stand for a similar fact overlaps with our approach to detect topically similar datasets based on the labels of their properties or classes.

Kunze et al. [155] propose a search engine for RDF datasets that is mainly based on filters that work similar to a faceted search. For ranking, the authors use a similarity function that comprises different aspects. One of these aspects is called topical aspect and is based on the vocabularies, that are used inside the different datasets. We will use this aspect as a baseline for comparison and explain it in more detail in Section 6.4.1.

Sleeman et al. [256] propose an approach to use topic modeling with RDF data. While their work has a similar basis it differs in many ways since it aims at other use cases. Their approach generates a single document for every entity described in a dataset while our approach creates a single document for every RDF dataset. Thus, their documents are based on a different set of triples and on different textual data gathered from the dataset.

Wagner et al. [299] propose an entity-centric search for datasets that are related to a user-defined keyword query. As described in Section 5.1.1, it relies on characterizing datasets by using entity clusters. Ellefi et al. [32] point out that this type of dataset characterization could be used to identify linking candidates.

Mehdi et al. [178] present an approach to recommend RDF datasets for linking with a new dataset based on keyword lists created by a domain expert. The keywords are used to run SPARQL queries on the available RDF datasets. After that, the datasets are ranked based on the number of matches that have been found. While the approach is comparable to TAPIOCA, it comes with two disadvantages. First, the keywords have to be created manually. Second, because of limitations in SPARQL, only exact matches are found. The authors propose the extension of the keyword list by generating different writings of the keywords. However, TAPIOCA has the advantage that it is based on a topic modeling approach and, hence, is not bound to single keywords.

Lopes et al. [170] present two approaches for recommending datasets for linking, which are built on previous publications. The first approach uses a Bayesian classifier and a set of features [163, 170]. The authors use the occurrence of properties, classes, and vocabularies IRIs as features. Hence, the main idea of the classifier is that datasets that are linked to a third datasets which is similar to the given query dataset are good candidates for linking. This approach is different to TAPIOCA since 1) it is a supervised approach and 2) it solely relies on IRIs. Especially the latter limits its usage for datasets that make use of different and sometimes even

automatically generated ontologies. The second approach relies on two metrics of the social network analysis area [169, 170]. The preferential attachment metric favors datasets that are already linked with many other datasets. The resource allocation metric favors datasets that have many common neighbors with the query dataset. Thus, the second approach mainly works for datasets that already have links to other datasets. This is not the scenario we look at since a newly created RDF dataset may not have any connection to other already published datasets.

Ellefi et al. [32] developed an approach similar to ours concurrently to our works. They suggest to rely on the concepts that are used within the datasets to generate documents that represent the datasets. For the comparison of two datasets, they rely on the WordNet-based measures Wu-Palmer [307] and Lin [167]. However, within their evaluation the best results are achieved using the UMBC similarity measure [115], which is a mixture of a WordNet-based similarity and a statistical similarity. All datasets that have a higher similarity than a given threshold are ranked based on their cosine similarity of their document's *tf-idf* vectors. In contrast to our approach, they use the concepts' natural language description in addition to concept labels. Another major difference is that they do not take properties into account. However, the source code of the approach is not available. Hence, a direct comparison with TAPIOCA is not possible at the time of writing.

Liu et al. [168] propose a supervised machine learning approach for the prediction of links between datasets. The approach relies on unsupervised link prediction algorithms. The results of these algorithms are used as input for a random forest classifier which is trained using examples from the Linked Open Data cloud. The unsupervised link prediction algorithms mainly rely on statistical similarities of the datasets within the graph [172]. These are the number of common neighbors (e.g., Adamic-Adar [7] or the Jaccard coefficient of common neighbors [172]), the number and length of paths that connect the datasets (e.g., Katz [143]), the number of connections the datasets already have (e.g., preferential attachment index [172]), or a random-walk-based metric (e.g., PageRank [54] or SimRank [136]). Hence, all these algorithms assume that both datasets already have connections with other datasets within the graph. This is not the case in the scenario we look at since a newly created RDF dataset may not have any connection to other already published datasets. Thus, the approach proposed by Liu et al. can only be used to increase the number of links between already linked datasets while our approach can also be used for newly, unconnected datasets.

Kopsachilis et al. [149, 150] propose a recommendation algorithm to identify potential geo-spatial RDF datasets for linking. To this end, they analyze RDF datasets,

extract geo-spatial data for each class that is used in the dataset and calculate the similarity of classes from different datasets based on the geo-spatial data. In comparison, our approach is not limited to geo-spatial data and access the search for similar datasets on a topical level.

6.2 Our Approach

The goal of TAPIOCA is to detect topically similar datasets with the aim of supporting the link discovery process. Ergo, given a query dataset represented as knowledge graph $\mathcal{G}_Q$ and a set of datasets $\mathscr{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots\}$, our aim is to rank the datasets by their likelihood of containing resources that should be linked to resources in $\mathcal{G}_Q$ [170]. The basic assumption behind our approach towards this goal is that datasets that should be linked should have similar topics. Hence, we adopt a topic-based modeling of the problem.

The TAPIOCA recommendation engine comprises three major components:

1. An index that contains known datasets,
2. A way to formulate a query, and
3. A method to calculate the topical similarity between a given query and the indexed datasets.

Of these three, the most challenging component is the definition of topical similarity between datasets. A definition of a similarity automatically results in requirements for the indexing and querying components. Therefore, we concentrate on this similarity calculation and present our new probabilistic topic-modeling-based approach. We will use the two example datasets `esd-columbia-gorge` and `esd-south-coast` to explain our approach. These examples are derived from real RDF datasets generated from open government data published by the State of Oregon. They contain contracts that have been concluded by different education service districts in 2013. The Listings 6.1 and 6.2 show the concise bounded descriptions [267] of two example entities of these datasets.[3]

[3]We use the prefixes `cg` and `sc` for the IRIs `http://data.oregon.gov/resource/i3bn-rwu4/` and `http://data.oregon.gov/resource/qhct-wumz/`, respectively. The original datasets have been available at `http://catalog.data.gov/dataset/contracts-esd-columbia-gorge-fiscal-year-2013-c3848` and `http://catalog.data.gov/dataset/contracts-esd-south-coast-fiscal-year-2013-3cb8d`, respectively. The first can still be accessed via `http://web.archive.org/web/20150928212350/https://catalog.data.gov/dataset/contracts-esd-columbia-gorge-fiscal-year-2013-c3848` while the latter does not seem to be available anymore (checked on 08.08.2022). For a better explanation of our approach, we made minor changes, e.g., we added two contract classes.

An RDF dataset contains two types of information that are relevant for our purposes: The first ones are the *individuals* that are described inside a dataset. However, data about individuals is not a good starting point for finding topically similarities between two datasets, since there would have to be at least one individual both datasets have in common. With respect to our example, this could lead to the comparison of data from the dataset, e.g., names, titles, keywords, and numbers, without the knowledge that the data comprises contract data. In such a case, we would only be able to identify these two datasets as similar if we are able to find individuals with the same name, title or other literals that occur in both datasets.

```
@prefix cg: <http://data.oregon.gov/resource/i3bn-rwu4/> .

cg:1
    a cg:Contract ,
    cg:type_of_contract_subcontract   "Material"  ,
    cg:esd_name "Columbia Gorge Education Service District" ,
    cg:award_title  "Technology Equipment" ,
    cg:award_type   "Price Agreement" ,
    cg:contractor_name  "TelCompany" ,
    cg:original_start_amendment_date  "03-07-12" ,
    cg:original_award_value  32456.92 ,
    cg:total_award_value_amendments   32456.92 .
```

Listing 6.1: Concise bounded description of an example entity of the `esd-columbia-gorge` dataset.

```
@prefix sc: <http://data.oregon.gov/resource/qhct-wumz/> .

sc:1
    a sc:Contract ,
    sc:esd_name  "South Coast ESD" ,
    sc:award_title  "Server" ,
    sc:award_type   "Lease" ,
    sc:contractor_information  "computer company" ,
    sc:start_date_expiration_date "7/1/10-6/30/14" ,
    sc:award_amount  5181.87 .
```

Listing 6.2: Concise bounded description of an example entity of the `esd-south-coast` dataset.

A much more promising approach is to look at the *structure* of the datasets. By doing so, we would know that both datasets contain a class and properties related to contracts. Following these assumptions, our approach is based on 1) extracting this structural metadata from a dataset and 2) transforming it into a description of the topically content of the dataset.

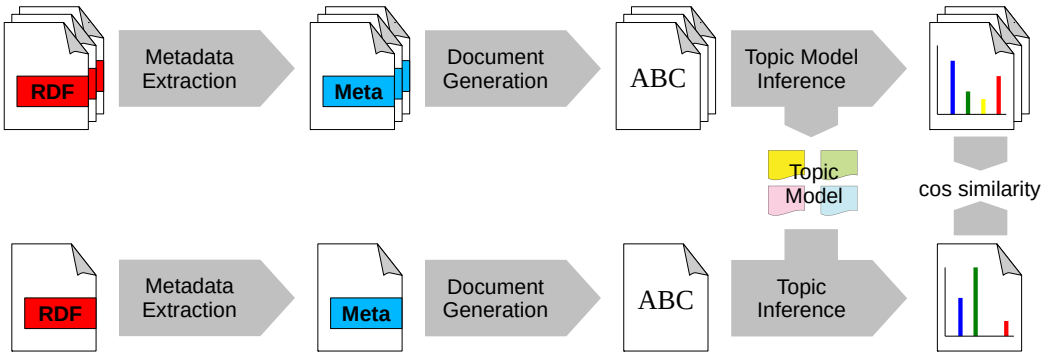


Figure 6.1.: The single steps of our approach. The upper part shows the index phase in which the topic model is generated while the lower part shows the handling of a query dataset.

Our approach is thus based on the three steps shown in Figure 6.1. First, the metadata of every single dataset in $\mathscr{G}$ is extracted. In the second step, the metadata is used to create a document describing the dataset. In the last step, a topic model is created based on the documents of the datasets. The resulting topic model and distributions enable a similarity calculation between single datasets based on their topic distribution. Additionally, the topic model can be used to determine the topic distribution of documents derived from new, unseen datasets. Thus, our approach is able to handle user input containing datasets that where not known during model inference. The steps underlying TAPIOCA are explained in more detail in the following subsections.

6.2.1 Metadata Extraction

Our approach for finding topical similarities between datasets is based on the metadata of these datasets and the RDF [52, 250] and OWL [124] semantics which underlie the Linked Data Web. The metadata comprises classes and properties used or defined inside a dataset. A frequency count f is assigned to every IRI of a class or property. This count stands for the number of entities of an extracted class or the number of triples of an extracted property. If a dataset contains metadata, i.e., triples with elements of the VoID vocabulary, then this information are extracted as well. After the extraction, classes and properties of the well-known vocabularies RDF, RDFS, OWL, Simple Knowledge Organization System (SKOS), and VoID are removed because these vocabularies do not contain any information about the topic of a dataset. Table 6.1 contains the IRIs that we extract from the two example datasets. Note, that the table does not contain the `rdf:type` property, because it has been removed as part of the RDF vocabulary.

Table 6.1.: Example IRIs extracted from the two example datasets for the triples shown in Listings 6.1 (upper part) and 6.2 (lower part of the table). The frequency count f is 1 for all of them in the example and is omitted in the table.

IRI	Type
cg:Contract	class
cg:type_of_contract_subcontract	property
cg:esd_name	property
cg:award_title	property
cg:award_type	property
cg:contractor_name	property
cg:original_start_amendment_date	property
cg:original_award_value	property
cg:total_award_value_amendments	property
sc:Contract	class
sc:esd_name	property
sc:award_title	property
sc:award_type	property
sc:contractor_information	property
sc:start_date_expiration_date	property
sc:award_amount	property

6.2.2 Document Generation

The generation of a document describing a certain dataset is based on the metadata extracted from this dataset. First, IRIs and their frequency counts f are selected from the metadata. After that, the labels of the IRIs are retrieved. The last step comprises the generation of the document corresponding to the dataset at hand by filtering stop words and determining the frequency of the single words.

There are three different possibilities to use the IRIs contained in the metadata of a dataset, leading to three different variants. Variant $\mathcal{V}_C$ uses only the class IRIs of the dataset, while $\mathcal{V}_P$ uses its property IRIs. $\mathcal{V}_A$ uses both IRI types—classes and properties. Depending on the variant, the IRIs and their counts are selected for the next step.

The labels of each of the selected IRIs are retrieved and tokenized. This label retrieval is based on the list of IRIs that have been identified as label containing properties by Ell et al. [87]. If there are no labels available, the vocabulary part of the IRI is removed and the remaining part is used as label. If this generated label is written in camel case or contains symbols like underscores, it is split into multiple words. The derived words inherit the counts f of their IRI. If more than one IRI created the same word, their counts are summed up.

After generating a list of words all stop words are removed[4]. After that the words are inserted into the document based on their frequency counts. Since LDA uses the bag-of-words assumption, only the frequency of the words matters while their order makes no difference. However, using the extracted counts directly could result in large documents, because a dataset can contain millions of triples. We tested two different variants to reduce the counts f to a manageable number of times the word type w occurs in the document of the i-th dataset. The first variant $\mathcal{V}_{*U}$ inserts every word only once, therewith creating a list of unique words with $\psi_{i,\mathrm{w}} = 1$. The second variant $\mathcal{V}_{*L}$ uses the logarithm of the counts leading to $\psi_{i,\mathrm{w}} = \mathrm{r}(\log_2(\mathrm{f}) + 1)$ where r is the rounding function which results the closest integer value preferring the higher value in case of a tie [3].

Thus, the whole document generation has six different variants—the product of three different IRI selections and two different word frequency definitions. Throughout this chapter we will use their abbreviations—$\mathcal{V}_{CU}$, $\mathcal{V}_{PU}$, and $\mathcal{V}_{AU}$ for the variants that use lists of unique words as well as $\mathcal{V}_{CL}$, $\mathcal{V}_{PL}$, and $\mathcal{V}_{AL}$ for the logarithm-based variants.

At the end of the Document Generation every dataset is represented by a single document. With the variant $\mathcal{V}_{AU}$, the following two documents are created for the two example datasets.

contract type subcontract esd name award title contractor original start amendment date value amendments

contract esd name award title type contractor information start date expiration amount

6.2.3 Topic Model Inference

At this stage of our approach, there is a corpus containing a single document for every dataset. This corpus is used to generate a topic model using the Gibbs-Sampling-based inference algorithm for LDA explained in Section 2.2.2.[5] As a result of the inference, we get a distribution over topics for every document d_i of the corpus (θ_i) and a distribution over words for every topic of the model (ϕ_i for the i-th topic). The second type of distribution allows the inference of a distribution over topics $\theta_{\mathcal{G}_Q}$ for

[4]The stop word list used can be found at `https://github.com/AKSW/topicmodeling/blob/master/topicmodeling.lang/src/main/resources/english.stopwords`; last accessed on 13.08.2022.

[5]We use Mallet [176] for our implementation of TAPIOCA.

a new document that has been created for a new dataset $\mathcal{G}_Q$ even if this document is not contained in the training corpus. However, this inference has two differences to the inference of a complete topic model described in Section 2.2.2. First, the word distributions Φ of the topic model are used to derive the probability that a topic can create a word. Second, only the topic indexes $Z_{d_{\mathcal{G}_Q}}$ are updated during the iterations while the word distributions Φ of all topics are treated as constants. This changes the calculation of the probability that a word token w at the j-th position in document $d_{\mathcal{G}_Q}$ and the word type w has the k-th topic assigned (Equation 2.18) to:

$$\mathbb{P}\left(z_{1,j} = k \,|\Phi, w_{1,j} = \mathrm{w}\right) = \left(\frac{\zeta_{1,k} + \alpha_k}{\sum_{i=1}^{\varrho} \zeta_{1,i} + \alpha_i}\right) \mathbb{P}\left(\mathrm{w}|\, \phi_k\right) . \tag{6.1}$$

Note that we replaced the document indexes in the equation with 1 since this inference is run per query document and the position of the query document within a larger corpus has no influence. Finally, the topic distribution $\theta_{d_{\mathcal{G}_Q}}$ of the document is calculated based on the final $Z_{d_{\mathcal{G}_Q}}$.

In our simple example, there might be three topics. While the words *subcontract*, *original*, *amendment*, *amendments*, and *value* are marked with the first topic, the second topic could contain the words *information*, *expiration*, and *amount*. The third topic contains the remaining words. To visualize this better, we apply a different format to the single word tokens in the example documents for the different topics.

contract type subcontract *esd name award title contractor*
original *start* amendment *date* value amendments

contract esd name award title type contractor information
start date expiration amount

6.2.4 Similarity Calculation

The last part of TAPIOCA is the definition of the similarity of two datasets $\mathcal{G}_i$ and $\mathcal{G}_j$. We define this similarity as the similarity of their topic distributions θ_i and θ_j. Since the topic distributions can be seen as vectors, we use the cosine similarity of these vectors as proposed by Steyvers et al. [266].[6] This similarity is defined as:

$$\mathrm{sim}_{\cos}(\mathcal{G}_i, \mathcal{G}_j) = \frac{\theta_i \cdot \theta_j}{|\theta_i|\,|\theta_j|} , \tag{6.2}$$

[6] Since we compare distributions, it is possible to use the Jensen-Shannon divergence instead of the cosine similarity. However, during the evaluation of our approach both similarity calculations have a similar performance.

where $\cdot$ denotes the dot product.

According to the example topic model we described above, The `esd-columbia-gorge` document would have $\theta_{\texttt{cg}} = \left(\frac{5}{14}, 0, \frac{9}{14}\right)$ while the `esd-south-coast` document has $\theta_{\texttt{sc}} = \left(0, \frac{3}{12}, \frac{9}{12}\right)$. Thus, the similarity of our example datasets would be 0.829.

6.3 Benchmarking Dataset Linking Recommendation Systems

The evaluation of a linking recommendation system like TAPIOCA is challenging. To measure the quality of the recommendations, a ground truth has to be available. Lopes et al. [170] and Ellefi et al. [32] suggest to use existing links between datasets in the LOD cloud as ground truth. However, this includes the assumption that only links that already exist in the cloud are good links and that all further linking between existing datasets is not good. This assumption is very strong and is unlikely to hold for the LOD cloud. Hence, it has to be assumed that such a ground truth would be incomplete.

Another approach proposed by us in our first experiments is to rely on domain experts and their decision whether two datasets of a given set of datasets are good candidates for linking. While we used 2 experts and measured their inter-rater agreement, the creation of the ground truth is closely bound to the opinion of these experts. Hence, it could be deduced that the created ground truth is subjective. A counter measure for this subjectivity is to involve more experts. However, this increases the costs of the creation of the ground truth.

Our goal is to create a benchmark for dataset linking recommendation systems that avoids the aforementioned drawbacks. To this end, we propose GLISTEN—a benchmark for dataset linking recommendation which relies on an external task to measure the quality of the recommendations. This follows the schema of an extrinsic evaluation [141], i.e., to evaluate the performance of an approach like TAPIOCA, it is embedded into an application. The approach that improves the overall performance of the application the most is the best approach among the evaluated approaches. Let f be a function that takes an RDF dataset as input, e.g., a classification task. Let e be a KPI that measures the performance of f, e.g., the F1 measure. The main idea of GLISTEN is to measure the performance of f when it receives only the query dataset. After that, the start dataset is extended by linking and fusing it with recommended datasets. The extended dataset is used as new input for f and

the performance is measured again. This is repeated until a maximum number of recommended datasets has been linked and fused with the start dataset. The earlier the measured performance increases and the larger the increase is, the better are the recommendations generated by the recommendation algorithm.

GLISTEN is based on a given set of RDF datasets from which at least one query dataset is chosen. The following steps are performed for each query dataset:

1. The benchmarked recommendation system is queried with the query dataset. A ranking of all other datasets is received.
2. The query dataset is linked to the datasets within the order of the ranking. Starting with the top rank, a fused dataset is created for each rank. This fused dataset comprises the query dataset and all recommended datasets that have a better or equal rank to the current rank.
3. The fused datasets are used as input for the external task system that implements f. In our current implementation of GLISTEN, we make use of the unsupervised fact checking algorithm COPAAL [270]. This system is executed once with the query dataset and each fused dataset. For each run, the performance of the system is measured using a chosen KPI. We follow the evaluation of Syed et al. [270] and use the AUC-ROC as e.
4. Finally, based on the measured performance for each of the input datasets, a rating for the recommendations is calculated.

The steps 2–4 are explained in more detail in the following. We start with a short introduction of fact checking, which is used as external task. This includes an explanation of COPAAL, which is the system for which the performance is measured depending on the query dataset and the fused datasets. After that, we explain the linking and fusion of the datasets before we introduce the measurements that GLISTEN uses.

6.3.1 Fact Checking

Syed et al. define the task of validating a statement as follows: "Given a fact, compute the likelihood that the given fact is true" [270]. In the context of the Semantic Web, a fact is a synonym for a triple as defined in Definition 2.6. Fact checking approaches for single facts can be separated into two groups. The first group are text-based approaches. These approaches rely on a reference corpus and use it to identify textual evidence for the given fact [269]. The second group comprises approaches that rely on a reference knowledge graph. Such a knowledge-graph-based fact checking approach makes use of structured information of the

reference knowledge graph to identify pieces of evidence that support or refute the given fact. These pieces of evidence can be paths that connect the subject and object of the given fact and have a statistical relation to the fact's predicate. Syed et al. [270, 271] present a fact checking approach named COPAAL that relies on this idea. We will use fact checking as external task and COPAAL as external system within GLISTEN and explain its functionality in the following.

COPAAL

Definition 6.1 (Path). *A path of length $\mathfrak{l}$ in a knowledge graph $\mathcal{G}$ is a cycle-free sequence of triples from $\mathcal{G}$ of the form $\{(v_0, p_1, v_1), (v_1, p_2, v_2), \ldots, (v_{\mathfrak{l}-1}, p_{\mathfrak{l}}, v_{\mathfrak{l}})\}$, where $\forall i \in [0, \mathfrak{l}-1] : (v_i, p_{i+1}, v_{i+1}) \in \mathcal{G} \vee (v_{i+1}, p_{i+1}, v_i) \in \mathcal{G}$. This also means that $\forall i, j \in [0, \mathfrak{l}], i \neq j \rightarrow v_i \neq v_j$ and $\mathfrak{l} > 0$ [270].*[7]

There can be several paths between two nodes v_0 and $v_{\mathfrak{l}}$ with the length $\mathfrak{l}$ within a knowledge graph. We use $\Pi(\mathfrak{l}, v_0, v_{\mathfrak{l}})$ to denote the set of all paths of length $\mathfrak{l}$ between the nodes v_0 and $v_{\mathfrak{l}}$ in $\mathcal{G}$. We refer to the k-th path within this set with $\pi_k(\mathfrak{l}, v_0, v_{\mathfrak{l}})$ [270].

Definition 6.2 (Typed paths). *Let C_s and C_o be two sets of RDF classes. The set of typed paths $\Pi'(\mathfrak{l}, C_s, C_o)$ of length $\mathfrak{l}$ between vertices that are instances of all classes C_s and C_o in a knowledge graph $\mathcal{G}$, respectively, are defined as follows [270]:*

$$\Pi'(\mathfrak{l}, C_s, C_o) = \{\pi_k(\mathfrak{l}, v_0, v_{\mathfrak{l}}) |\ C_s \subseteq \mathfrak{c}_{\mathcal{G}}(v_0) \wedge C_o \subseteq \mathfrak{c}_{\mathcal{G}}(v_{\mathfrak{l}})\}\ . \tag{6.3}$$

These typed paths are used by Syed et al. [270] to identify paths that corroborate the correctness of the given fact (s, p, o).

Definition 6.3 (Corroborative paths). *Using the domain and range of the given fact's predicate p the set of corroborative paths with a maximum length of $\mathfrak{l}$ is defined as [270]:*

$$\Pi(\mathfrak{l}, p) = \bigcup_{j=1}^{\mathfrak{l}} \Pi'(j, \mathfrak{d}(p), \mathfrak{r}(p))\,. \tag{6.4}$$

Following Syed et al. [270], we further restrict typed paths by using a vector of properties $\vec{\mathfrak{q}} = \{\mathfrak{q}_1, \ldots, \mathfrak{q}_{\mathfrak{l}}\}$.

[7] Syed et al. [270] distinguish between directed and undirected paths. Since their evaluation shows that undirected paths give the best results, we only use undirected paths within this thesis.

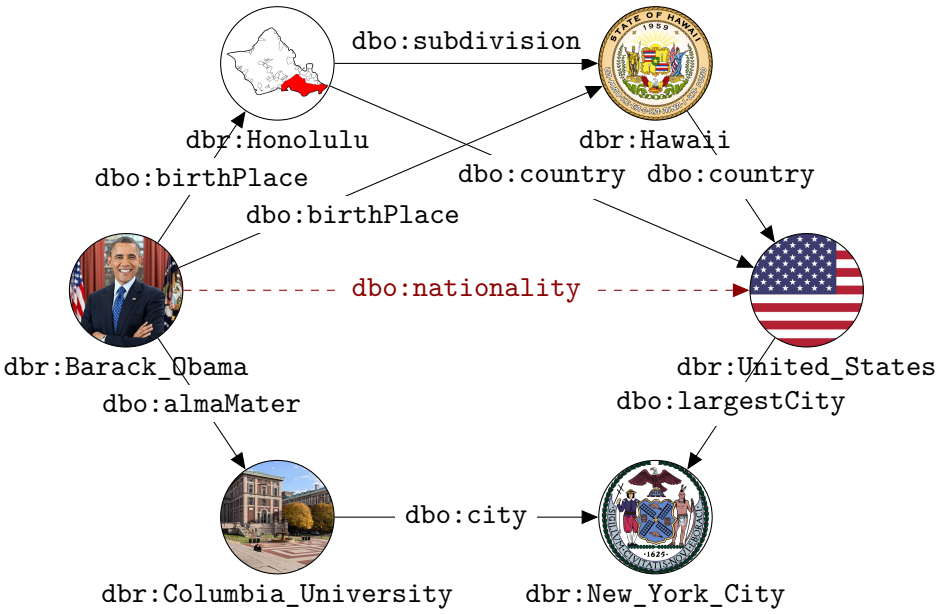


Figure 6.2.: Example paths for checking the given triple (`dbr:Barack_Obama` , `dbo:nationality` , `dbr:United_States`) [270].

Definition 6.4 ($\vec{\mathfrak{q}}$-restricted typed paths)**.** *We define the set of $\vec{\mathfrak{q}}$-restricted typed paths $\Pi'(\mathfrak{l}, C_s, C_o, \vec{\mathfrak{q}}) \subseteq \Pi'(\mathfrak{l}, C_s, C_o)$ as follows [270]:*

$$\begin{aligned}\Pi'(\mathfrak{l}, C_s, C_o, \vec{\mathfrak{q}}) = \{\pi_k(\mathfrak{l}, v_0, v_\mathfrak{l}) \mid \pi_k(\mathfrak{l}, v_0, v_\mathfrak{l}) \in \Pi'(\mathfrak{l}, C_s, C_o) \wedge \\ \forall i \in [0, \mathfrak{l}-1] : (v_i, p_{i+1}, v_{i+1}) \in \pi_k(\mathfrak{l}, v_0, v_\mathfrak{l}) \\ \wedge\, p_{i+1} = \mathfrak{q}_{i+1}\}\,.\end{aligned} \tag{6.5}$$

The implementation of COPAAL comprises three main steps. The first step is the search for corroborative paths $\Pi(\mathfrak{l}, p)$. We use a breadth-first search that identifies all paths between s and o of the given triple (s, p, o).[8] As expressed in Equation 6.4 the search identifies paths with a length up to $\mathfrak{l}$. Figure 6.2 shows an example for the triple (`dbr:Barack_Obama`, `dbo:nationality`, `dbr:United_States`). The search identified several paths that have a maximum length of $\mathfrak{l} = 3$ and connect the subject and object. Based on the identified paths, a set of $\vec{\mathfrak{q}}$-restricted typed paths is created by extracting the properties of the paths. Listing 6.3 shows the restrictions that would be extracted for the example above. It is worth noticing that the first restriction is only taken into account once although there are two paths in

[8] Although using s and o as start for the search will not identify all possible corroborative paths for p, it will identify those paths that exist between s and o. Only those paths will have an effect on the final score that is calculated by COPAAL. Hence, the focus on s and o will find all paths that are used by the calculation and is less costly than searching for all possible corroborating paths for p.

```
dbo:birthPlace/dbo:country
dbo:birthPlace/dbo:subdivision/dbo:country
dbo:almaMater/dbo:city/^dbo:largestCity
```

Listing 6.3: The path restrictions $\vec{q}$ extracted for the example graph (written as property paths following Harris et al. [116]).

the example that fulfill this restriction. One of the two paths goes via `dbr:Honolulu` while the second goes via `dbr:Hawaii`.

The second step comprises the scoring of each $\vec{q}$-restricted typed path. The calculated score expresses to which extend a path supports the existence of the given fact. Syed et al. [270] suggest to measure the co-occurrence of paths with the same $\vec{q}$ that connect the same subject and object as facts with p. Their measure is inspired by the NPMI [49]. However, they argue that the calculation of the exact NPMI is computationally expensive and suggest an approximation. Instead of counting the exact number of subject and object pairs, the approximation is based on the idea to count the number of paths. Although this number is higher than the number of pairs, it is computationally cheaper to determine. However, this approximation leads to an overestimation of the probability of the paths. We identified this as a drawback and adapt COPAAL to count the exact number of pairs. Hence, we calculate the NPMI for a $\vec{q}$-restricted typed path and a given property p as follows:

$$\mathbb{P}\left(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}})\right) = \frac{\left|\left\{(v_i, v_j) \,|\pi_k(\mathfrak{l}, v_i, v_j) \in \Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}})\right\}\right|}{|\mathfrak{i}_{\mathcal{G}}(\mathfrak{d}(p))||\mathfrak{i}_{\mathcal{G}}(\mathfrak{r}(p))|}, \tag{6.6}$$

$$\mathbb{P}\left(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p\right) = \frac{\left|\left\{(v_i, v_j) \,|\pi_k(\mathfrak{l}, v_i, v_j) \in \Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}) \wedge (v_i, p, v_j) \in \mathcal{G}\right\}\right|}{|\mathfrak{i}_{\mathcal{G}}(\mathfrak{d}(p))||\mathfrak{i}_{\mathcal{G}}(\mathfrak{r}(p))|}, \tag{6.7}$$

$$\text{NPMI}\left(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p\right) = \frac{\log\left(\frac{\mathbb{P}(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p)}{\mathbb{P}(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}))\mathbb{P}(p)}\right)}{-\log \mathbb{P}\left(\Pi'(\mathfrak{l}, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p\right)}. \tag{6.8}$$

Equation 6.6 defines the probability that two nodes, which fulfill the domain and range of the property of the given triple, are connected by a path with a given restriction vector $\vec{q}$. Equation 6.7 defines the joint probability of such a pair being connected with the property of the given triple in addition to the path. Finally, Equation 6.8 defines the NPMI of a $\vec{q}$-restricted typed path and the property of the given triple. We define the probability $\mathbb{P}(p)$ as the probability that a randomly chosen triple of $\mathcal{G}$ has p as predicate.

The third and last step of COPAAL is to summarize the scores of the identified paths. We do not use the summarization function proposed by Syed et al. [270] since it does not handle negative NPMI values well. Instead, we separate the paths based on their NPMI scores in two groups. Let $\mathfrak{Q}_j^+$ be the set of all restrictions $\vec{\mathfrak{q}}$ with length j that have been identified in the first step and that have got an NPMI value larger 0. Let $\mathfrak{Q}_j^-$ be defined in a similar way for paths with a negative NPMI value.[9] The veracity score $\mathfrak{z}$ of a given triple (s, p, o) is defined as:

$$\begin{aligned}\mathfrak{z}(s,p,o) = &\left(1 - \prod_{j=1}^{\mathfrak{l}} \prod_{\vec{\mathfrak{q}} \in \mathfrak{Q}_j^+} \left(1 - \text{NPMI}\left(\Pi'(j, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p\right)\right)\right) \\ &- \left(1 - \prod_{j=1}^{\mathfrak{l}} \prod_{\vec{\mathfrak{q}} \in \mathfrak{Q}_j^-} \left(1 + \text{NPMI}\left(\Pi'(j, \mathfrak{d}(p), \mathfrak{r}(p), \vec{\mathfrak{q}}), p\right)\right)\right) .\end{aligned} \tag{6.9}$$

Integration

To integrate a fact validation system like COPAAL into GLISTEN's workflow, we have to evaluate the fact validation system's performance given different knowledge graphs. These knowledge graphs will be used as reference knowledge graphs by COPAAL to evaluate a set of RDF statements. To this end, we create a set of RDF statements $\mathscr{S}$ for which the veracity is known. It consists of two subsets $\mathscr{S}^+$ and $\mathscr{S}^-$ comprising true and false statements, respectively. The true statements are taken from the given query dataset, i.e., $\mathscr{S}^+ \subset T_{\mathcal{G}_Q}$. The false statements are generated as suggested by Gerber et al. [107]. We use a triple from the query dataset and replace either the subject or the object with another resource from the dataset. This takes into account that the domain and range of the chosen triple's property have to be fulfilled by the resources used for the replacement. After the creation of a new triple, it is ensured that the triple does not already exist in the query dataset.

After creating the test dataset $\mathscr{S}$, we use COPAAL to determine the veracity scores of all statements in the test set based on the different reference knowledge graphs that are created by linking and fusing the query dataset with the recommended datasets. It is important to note that before running COPAAL, the set of true statements $\mathscr{S}^+$ is removed from the reference knowledge graph. Otherwise, the validation task would be trivial.

[9] The definition of these two sets implies that paths with an NPMI score of exactly 0 are ignored.

The performance of COPAAL is measured using the area under receiver operating characteristic curve (AUC-ROC) [269, 270]. We make use of an established GERBIL instance to calculate this metric [204, 216, 260].

6.3.2 Linking and Fusion

The second step of the GLISTEN workflow has the goal to create new, fused datasets which comprise the query dataset and the recommended datasets according to their order. This can be separated into two smaller steps: 1) the generation of links between the two RDF knowledge graphs $\mathcal{G}_\mathbf{S}$ and $\mathcal{G}_\mathbf{T}$ and 2) their fusion to create a new, fused dataset. The first step is named Link Discovery (or just linking).

Definition 6.5 (Link Discovery)**.** *Given two sets of source and target resources $R_{\mathcal{G}_\mathbf{S}}$ and $R_{\mathcal{G}_\mathbf{T}}$, respectively, and a property p, Link Discovery has the goal to find the following set $\mathbb{L}$ [205]:*

$$\mathbb{L} = \{(v_i, v_j) | (v_i, v_j) \in R_{\mathcal{G}_\mathbf{S}} \times R_{\mathcal{G}_\mathbf{T}} \wedge (v_i, p, v_j) \text{ is true } \}. \tag{6.10}$$

Within GLISTEN, we are interested in linking resources of the query dataset $\mathcal{G}_Q$ with resources of the recommended datasets using the `owl:sameAs` relation. This relation expresses that two resources stand for the same real-world entity [189, 212]. Hence, the result of the linking will be a link set that comprises pairs of resources that represent the same real-world entity.

Assume the linking of the two knowledge graphs DBpedia [21, 162] and Wikidata [298]. Figure 6.3 shows an excerpt of these two graphs.[10] On the left, there are triples from the DBpedia that are mainly about the resource `dbr:Barack_Obama`. On the right, there are two Wikidata resources to which the resource from the left could be linked. A linking algorithm typical relies on additional information about the resources that it gathers from the source and target knowledge graphs. The example includes the names (`rdfs:label`) and birth names (`dbo:birthName` and `wdt:P1477`) of the entities. Based on this additional information, a linking algorithm may identify `dbr:Barack_Obama` and `wd:Q76` as the same real-world entity. Based on further information, it may also link `dbr:United_States` to `wd:Q30`.

There are several linking frameworks and approaches available [166, 198, 205, 297]. For GLISTEN, we make use of the Wombat [252] algorithm implemented in the LIMES

[10] We use the prefixes `wd` and `wdt` for the IRIs `http://www.wikidata.org/entity/` and `http://www.wikidata.org/prop/direct/`, respectively.

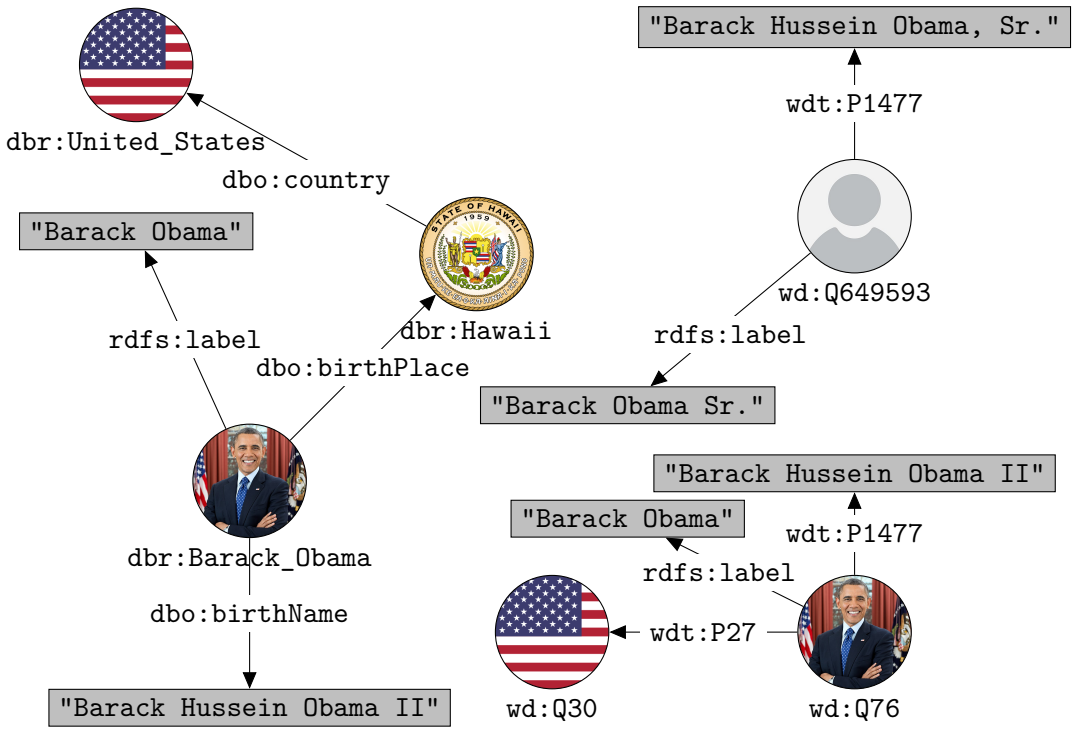


Figure 6.3.: Example entities from DBpedia (left) and Wikidata (right). IRIs have been shortened using prefixes.

framework [205]. Wombat is a supervised machine-learning approach that solely relies on positive examples. It uses link specifications which are defined as a set of conditions under which it is assumed that the relation p between v_i and v_j holds. It starts with a set of atomic link specification and generates combinations of them searching of the the least general generalization of the given positive examples [252]. In a more recent version, Wombat is available as unsupervised machine learning algorithm [84] and can be used by GLISTEN.

The fusion step merges the two linked datasets into a single, new dataset.[11] Hence, we define the fusion operation as follows:

Definition 6.6 (Fusion)**.** *Given two knowledge graphs $\mathcal{G}_\mathbf{S}$ and $\mathcal{G}_\mathbf{T}$, and a link set $\mathbb{L}$, knowledge graph fusion creates a new knowledge graph $\mathcal{G}_F$ with a set of merged triples that is defined as follows:*

$$T_{\mathcal{G}_F} = \{(b(v_i), p, b(v_j)) | (v_i, p, v_j) \in T_{\mathcal{G}_\mathbf{S}} \cup T_{\mathcal{G}_\mathbf{T}}\} \ , \tag{6.11}$$

[11] Note that the term "knowledge graph fusion" is used with different meanings in the literature. In a recent survey, Zhao et al. [311] distinguish 4 different types of knowledge graph fusion approaches. The term fusion as we define it fits best into the "multi-knowledge graph fusion" category. However, in difference to Zhao et al. [311], we separate the fusion from the linking (which is named entity alignment in their work) and do not align the properties or classes of the two datasets.

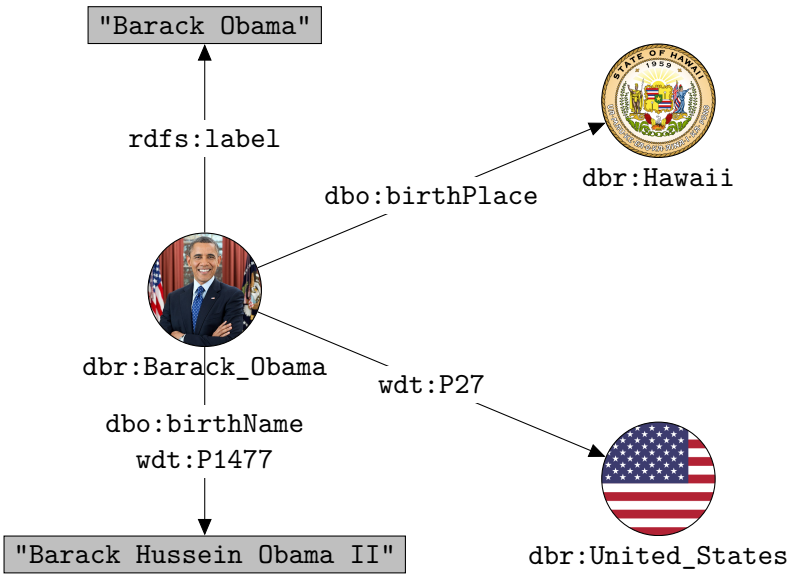


Figure 6.4.: Example entities from the fused dataset. IRIs have been shortened using prefixes.

where b is a replacement function.[12] *This function replaces all resources from $R_{\mathcal{G}_\mathbf{T}}$ with their linked resource from $R_{\mathcal{G}_\mathbf{S}}$. We define the replacement function as follows:*

$$b(v_i) = \begin{cases} v_k & \text{if } (v_k, v_i) \in \mathbb{L}\,, \\ v_i & \text{else}\,. \end{cases} \tag{6.12}$$

For the example in Figure 6.3, the linking created the link pairs (`dbr:Barack_Obama`, `wd:Q76`) and (`dbr:United_States`, `wd:Q30`). Figure 6.4 shows the result of the fusion based on these links with a focus on the resource `dbr:Barack_Obama`.[13] It can be seen that as a result of the fusion the resource `dbr:Barack_Obama` has a direct link to the resource `dbr:United_States` via the "country of citizenship" property (`wdt:P27`). This is a piece of information that is not part of the DBpedia but comes from the Wikidata knowledge graph.

The linking and fusion step is computationally costly. First, a naive approach for linking two datasets $\mathcal{G}_\mathbf{S}$ and $\mathcal{G}_\mathbf{T}$ has a complexity of $O(|V_{\mathcal{G}_\mathbf{S}}||V_{\mathcal{G}_\mathbf{T}}|)$. This complexity can be reduced [205]. However, linking remains a costly operation. Second, linking approaches rely on the idea of pairwise linking, i.e., only two datasets are linked to each other at a time. If more than two datasets are linked, the order in which the

[12] Our definition of fusion is bound to a link set that represents links which express that two linked nodes represent the same real-world entity. This allows the usage of the replacement function.

[13] We removed the resource `wd:Q649593` from the example, to keep it small. The linking algorithm would link it to the DBpedia resource `dbr:Barack_Obama_Sr.` and the fusion would merge their triples.

datasets are linked pairwise and fused into the newly created dataset can have an influence on the result.

The first challenge could be solved by precomputing the results of the linking and fusion step. In this case, the linking would have to be performed only once and the implemented benchmark could rely on the already generated fused datasets. However, the second challenge prohibits this. Since the order of the datasets has an influence on the result and intermediate results are needed for the evaluation, there are $|\mathscr{G}|!$ different possible combinations that would have to be generated. The amount of datasets that would have to be precomputed becomes intractable even for a small $\mathscr{G}$.

Kistowski et al. [293] point out that the design of a benchmark has to take usability into account. To keep the runtime of the benchmark low and, hence, ensure the usability of the benchmark we make the following assumptions:

1. The linking of a new dataset to the fused dataset should only take resources of the query dataset into account.
2. The order in which datasets are linked to the query dataset has no influence on the result.

For the first assumption, we argued that the benchmarked recommendation systems suggest datasets that should be linked to the query dataset. Taking possible connections between the suggested datasets into account is not an explicit part of the recommendation task.

These assumptions allow to tackle the linking and fusion step in a computationally efficient way. We link and fuse only the query dataset with each of the other datasets. For fusing more than one dataset, we will use the already created fusions of the query dataset with the single datasets. Let $\boxplus$ denote the binary operator that links and fuses two datasets and let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the first and second recommended dataset, respectively.[14] Based on our assumptions, we define that the fusion of these two datasets with the query dataset $\mathcal{G}_Q$ can be formulated as follows:

$$((\mathcal{G}_Q \boxplus \mathcal{G}_1) \boxplus \mathcal{G}_2) = (\mathcal{G}_Q \boxplus \mathcal{G}_1) \cup (\mathcal{G}_Q \boxplus \mathcal{G}_2)\,. \tag{6.13}$$

[14] The task to discover links between two datasets and fuse them are own fields of research [166, 198, 205, 311]. However, this thesis does not focus on them and we rely on existing implementations like the LIMES framework [205] and Wombat [252].

where $\cup$ creates the union of the triples of the fusion operations' results. We further extend this up to the fusion of the query dataset and all other datasets by defining:

$$(((\mathcal{G}_Q \boxplus \mathcal{G}_1) \boxplus \ldots) \boxplus \mathcal{G}_{|\mathscr{G}|}) = (\mathcal{G}_Q \boxplus \mathcal{G}_1) \cup \ldots \cup (\mathcal{G}_Q \boxplus \mathcal{G}_{|\mathscr{G}|}) \,, \tag{6.14}$$

Further, it should be noted that the second assumption leads to:

$$(\mathcal{G}_Q \boxplus \mathcal{G}_1) \cup (\mathcal{G}_Q \boxplus \mathcal{G}_2) = (\mathcal{G}_Q \boxplus \mathcal{G}_2) \cup (\mathcal{G}_Q \boxplus \mathcal{G}_1) \,. \tag{6.15}$$

These assumptions give two advantages. First, it reduces the number of possible combinations of datasets to $O(|\mathscr{G}|)$. Second, it allows the precomputation of all pairs $(\mathcal{G}_Q \boxplus \mathcal{G}_i)$ (with $\mathcal{G}_i \in \mathscr{G}$).

6.3.3 Measurement

Let $\mathcal{G}_Q$ be the query dataset and let $\mathcal{G}_N$ be the result of the fusion step, i.e., the result of linking and fusing $\mathcal{G}_Q$ with the first N recommended datasets. GLISTEN measures the performance of a recommender by comparing the performance that the external function f achieves with and without the recommended datasets. To this end, we use the given performance metric e to measure the performance of f on the fused dataset and compare it with its performance on the query dataset. We call this measure Δ_e. For the first N recommended datasets, the value of the measure is formally define it as follows:

$$\Delta_\mathrm{e}@N = \mathrm{e}(f(\mathcal{G}_N)) - \mathrm{e}(f(\mathcal{G}_Q)) \,. \tag{6.16}$$

This measure gives a value for each rank in the produced list of recommended datasets. To summarize these values, we use the area under the curve that Δ_e defines for consecutive values of N. We name this metric *area under delta curve* (▲). For the first j recommended datasets, we calculate this area using the following equation:

$$\blacktriangle_{\mathrm{e},j} = \int_1^j \Delta_\mathrm{e} = \sum_{N=1}^{j} \Delta_\mathrm{e}@N \,. \tag{6.17}$$

As explained above, the AUC-ROC is used to measure the performance of fact checking algorithms [269, 270]. Hence, we use this metric as e and measure the performance of linking recommendation approaches with $\Delta_{\text{AUC-ROC}}@N$and $\blacktriangle_{\text{AUC-ROC},j}$.

6.4 Evaluation

The aim of our evaluation is fourfold. In the first experiment, we focus on the evaluation of the different variants of TAPIOCA. We compare their performance against three baselines. In our second experiment, we evaluate the two approaches for detecting the best number of topics presented in Section 2.2.3 to test whether they can be applied to TAPIOCA. In the third experiment, we repeat the first two experiments at a larger scale to show that our approach works with a larger amount of data as well. In the fourth experiment, we test whether GLISTEN can be used to compare linking recommendation approaches.

The dataset used for the first three experiments is based on RDF datasets that have been indexed by LODStats [22, 90]. We remove those datasets that have no English description or not at least one class IRI or one property IRI of a vocabulary, that is not filtered out by our approach. The remaining evaluation dataset contains 1680 RDF datasets with 776 213 346 triples.

6.4.1 Baselines

We compare our approach with three baselines from the field of Information Retrieval as well as the Semantic Web. The first baseline is *tf-idf* [24] for which we extract the metadata and generate a document for every dataset as described in Sections 6.2.1 and 6.2.2. Let D be the set of known documents and $\mathbb{V}_D$ the corpus vocabulary containing all known word types w. Let $\psi_{i,\mathrm{w}}$ be the number of times the word type w occurs inside the i-th document. Then, a vector of length $|\mathbb{V}|$ can be generated for every document d_i by calculating a *tf-idf* value for every word type w using

$$\textit{tf-idf}\left(\mathrm{w}, d_i, D\right) = \psi_{i,\mathrm{w}} \log\left(\frac{|D|}{|\{d_i | d_i \in D \wedge \psi_{i,\mathrm{w}} > 0\}|}\right). \tag{6.18}$$

The first factor is called term frequency. The second factor in the equation is called inverse document frequency and reduces the impact of words that occur in many documents. Since *tf-idf* uses term frequencies and an instantiation of the single words is not needed, we use the pure frequencies instead of the logarithm or unique variant. After generating a vector for every document, the cosine similarity can be calculated.

The second baseline is the topical aspect (BL_T) used by Kunze et al. [155] as part of their RDF search engine described in Section 6.1. The main idea of this topical

aspect is to identify topically similar datasets based on the vocabularies that are used inside the datasets, i.e., similar datasets are expected to contain IRIs of the same vocabularies. Let $\mathscr{G}$ be the set of all known datasets and $\mathcal{G}_i, \mathcal{G}_j \in \mathscr{G}$. Let $\mathfrak{V}$ be the union of the vocabularies used in $\mathcal{G}_1$ or $\mathcal{G}_2$ and let $\mathscr{G}_\mathfrak{v} \subseteq \mathscr{G}$ be the set of all known datasets that use the vocabulary $\mathfrak{v}$. Then, the BL_T is defined as

$$BL_T\left(\mathcal{G}_i, \mathcal{G}_j\right) = \sum_{\mathfrak{v} \in \mathfrak{V}} -\log\left(\frac{|\mathscr{G}_\mathfrak{v}|}{|\mathscr{G}|}\right) \mathrm{u}(\mathcal{G}_i, \mathfrak{v})\mathrm{u}(\mathcal{G}_j, \mathfrak{v})\,, \tag{6.19}$$

where $\mathrm{u}(\mathcal{G}, \mathfrak{v})$ is a function that returns 1 if the vocabulary $\mathfrak{v}$ is used inside the dataset $\mathcal{G}$ or 0 otherwise. the first term in the sum is a weight of the vocabulary inspired by the inverse document frequency of *tf-idf*. Thus, the more datasets use the vocabulary, the less important it is for the topical similarity and the lower its weighting [155].

The last baseline uses Apache Lucene[15]. The generated documents are indexed using the standard analysis of Lucene, i.e., the documents are tokenized, the tokens are transformed into their lower-cased form and Lucene's stop word filter is applied. For every dataset, its document is used to generate a weighted boolean query containing the words of the documents and their counts as weights. This query is used to retrieve similar documents from the index together with Lucene's similarity score for them.

6.4.2 Experiment I

For the first experiment, we randomly select 100 RDF datasets to generate a gold standard. Two researchers independently determine topically similar datasets. For solving this task, they get the description of those datasets as well as the possibility to take a deeper look inside the data itself. We compared the ratings of both researchers and measured an inter-rater agreement of 97.58%.[16] Cases in which the ratings differ are discussed to compile a final rating. With this approach 86 dataset pairs are identified as topically similar.[17] Table 6.2 shows the features of the corpora that have been created by the different variants of our approach based on these 100 datasets (3 659 152 triples).

[15] `http://lucene.apache.org/`; last accessed on 13.08.2022.

[16] We measured the inter-rater agreement as the number of dataset combinations for which both raters made the same decision divided by the number of all possible dataset combinations.

[17] The gold standard can be found at `https://hobbitdata.informatik.uni-leipzig.de/homes/mroeder/tapioca/experiment_1/`; last accessed on 13.08.2022.

Table 6.2.: Features of the corpora generated by the different variants.

Variant	Word types	Word tokens
$\mathcal{V}_{AL}$	10 182	252 406
$\mathcal{V}_{AU}$	10 182	34 264
$\mathcal{V}_{CL}$	9 500	239 108
$\mathcal{V}_{CU}$	9 500	32 020
$\mathcal{V}_{PL}$	1 173	14 078
$\mathcal{V}_{PU}$	1 173	2 501

For all six approaches presented above, we calculate the similarities of every dataset to every other dataset using the leave-one-out method: we use one dataset as query while the different approaches are trained using the other 99 datasets of the gold standard. The result of this step is a ranked and scored list of corresponding datasets for each of the datasets in our gold standard. We then search for a similarity threshold that leads to a maximal micro F1-score over all datasets. For every variation of our approach, we run experiments in the range of $[2, 200]$ topics. Since the F1-score of the variant $\mathcal{V}_{AL}$ is still rising near 200 topics, we further increase the maximum number of topics for this variant to 500.[18]

The best micro F1-scores achieved by the different variants and the different baselines are shown in Table 6.3. Based on this data, our approach achieves a higher F1-score than all baselines if the document generation is based on properties and logarithmic counts. The variant with logarithmic counts for classes and properties works nearly as good as the variant that only uses properties. Moreover, our approach performs much better with logarithmic counts than with unique word frequencies. In Figure 6.5, we also see that with varying numbers of topics $\mathcal{V}_{AU}$ and $\mathcal{V}_{CU}$ stay at a low level. Only $\mathcal{V}_{PU}$ achieves competitive F1-scores. We think that this has two causes. First, the unique-based variants do not assign a weight to the labels regarding the importance that a class or a property has inside a dataset. Secondly, it has already been shown that LDA does not perform well on short documents in which many different words appear rarely, e.g., messages of short messaging services [310].

Regarding the IRIs used for the document creation, it can be seen that all approaches show a poor performance if they are only based on classes. These variants are only able to find similar datasets if the similarity is very obvious, e.g., different datasets of

[18] For all topic numbers, the inference was carried out with 1040 iterations, an asymmetric $\alpha = 0.1$ for each topic, $\beta = 0.01$ and a hyper parameter optimization after every 50-th iteration starting after iteration 200.

Table 6.3.: F1-scores achieved by the different variants and the baselines. In the most left column, there are the results for the variants $\mathcal{V}_{CL}$, $\mathcal{V}_{PL}$, and $\mathcal{V}_{AL}$ while the results of $\mathcal{V}_{CU}$, $\mathcal{V}_{PU}$, and $\mathcal{V}_{AU}$ are in the second most left column.

IRIs used	TAPIOCA (log.)	TAPIOCA (unique)	*tf-idf*	BL_T	Lucene
classes	0.128	0.083	0.103	0.292	0.096
properties	**0.505**	0.350	0.436	0.356	0.418
both	0.495	0.078	0.444	0.333	0.241

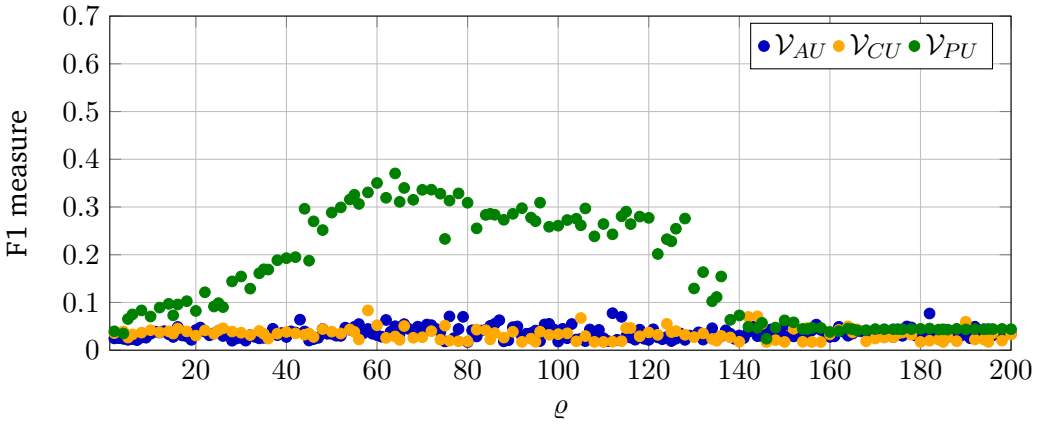


Figure 6.5.: The F1-scores of the three unique word based variants for different numbers of topics in the range $[2, 200]$.

the eagle-i that use the same vocabularies.[19] Additionally, they have the drawback, that only 88 out of the 100 datasets define or use classes which makes them unable to calculate similarities for 12 datasets.

Another observation of the experiment is that BL_T does not perform well. Thus, the assumption that topically similar datasets use the same vocabularies does not hold for all real-world datasets in our experiment. One core reason might be that many of the datasets we consider have been generated automatically from tables or CSV files. Every generated dataset has an own, generated vocabulary IRI like the two example datasets in Section 6.2.

The Figures 6.5, 6.6, and 6.7 show the influence of the number of topics on the models performance. For $\mathcal{V}_{PL}$, $\mathcal{V}_{AL}$, and $\mathcal{V}_{PU}$, there is a range of numbers of topics in which the F1-score is maximized. Models with too few topics have a much worse performance while—especially for $\mathcal{V}_{PL}$ and $\mathcal{V}_{AL}$—the performance deterioration caused by too many topics is rather small. Thus, we can summarize that finding a good number of topics is important for our approach. However, in case an exact

[19]The gold standard contains datasets of the eagle-i project. `https://open.catalyst.harvard.edu/products/eagle-i/`; last accessed on 13.08.2022.

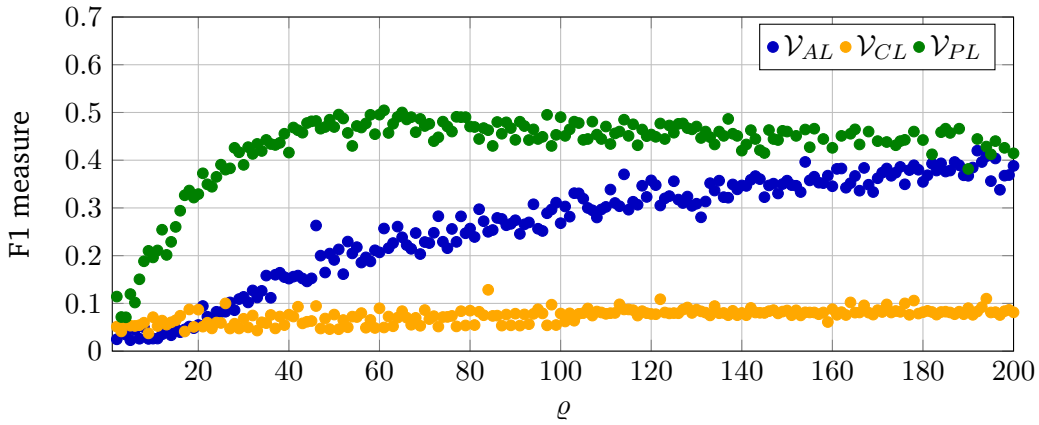


Figure 6.6.: The F1-scores of the three logarithm based variants for different numbers of topics in the range $[2, 200]$.

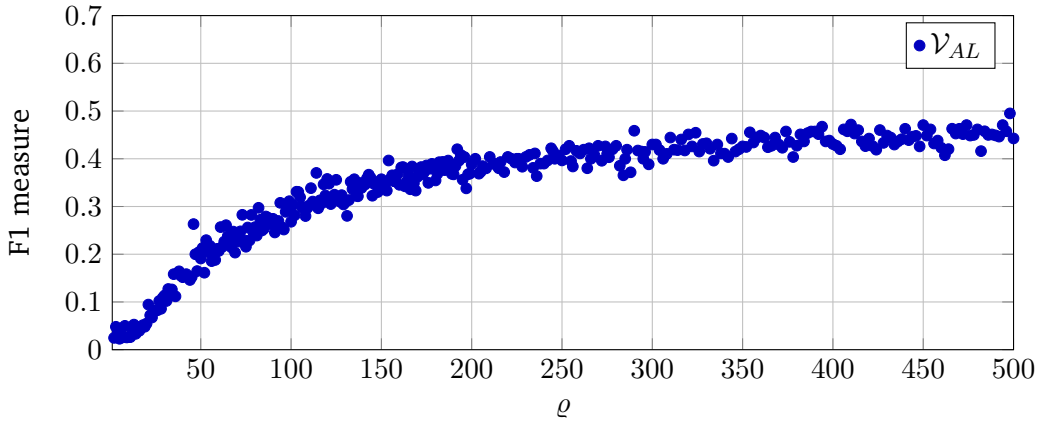


Figure 6.7.: The F1-scores of $\mathcal{V}_{AL}$ for different numbers of topics in the range $[2, 500]$.

number cannot be determined, a high number of topics should be preferred. This is in line with the findings of Wallach et al. [300], who show that hyper parameter optimization can lead to a higher robustness against a high number of topics.

6.4.3 Experiment II

Based on the results of the first experiment, we evaluate whether the two approaches for determining a good number of topics presented in Section 2.2.3 are useful in the present use case. Thus, for the topic range $[2, 200]$, we generate topic models using all documents of the gold standard datasets that have been generated by the $\mathcal{V}_{PL}$ variant of our approach. For every number of topics we generate five models, calculate $\mathbb{P}(D|\Phi)$ as well as $\mathcal{A}$ and determine the average values of these five runs.

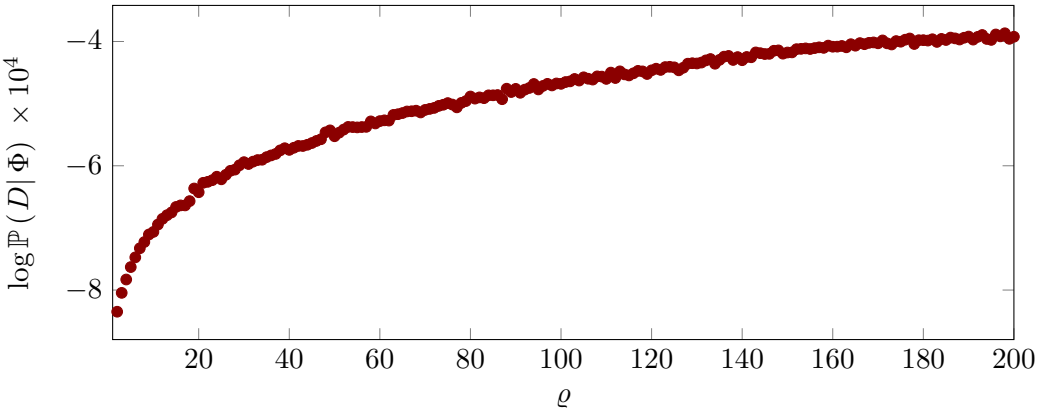


Figure 6.8.: Average $\log(\mathbb{P}(D|\Phi))$ calculated on the gold standard corpus for different models of the $\mathcal{V}_{PL}$ variant with different numbers of topics.

Figure 6.8 shows the average logarithm of $\mathbb{P}(D|\Phi)$ and reveals that the probability increases steadily with an increasing number of topics. Thus, this method would recommend a much higher number of topics than the 61 topics with which the $\mathcal{V}_{PL}$ variant performs best. The average value of $\mathcal{A}$ is shown in Figure 6.9. The curve shows a dip as described by Arun et al. [19]. But the minimum value of this dip has been achieved by models with 11 topics with which $\mathcal{V}_{PL}$ has only an F1-score of 0.21.

From this experiment, we can summarise that none of these approaches seems to be appropriate to determine a good number of topics for our use case. Therefore, we have to fall back on a simple alternative that we will present during the third experiment.

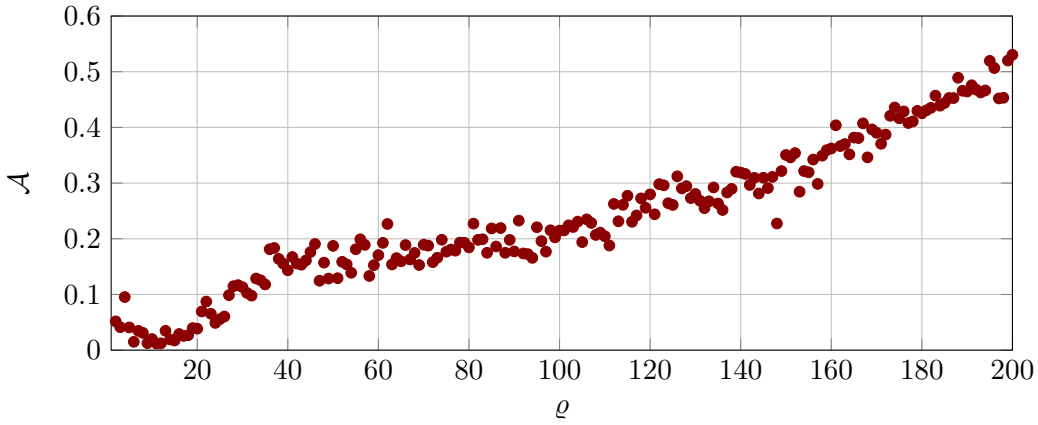


Figure 6.9.: Average values of the measure $\mathcal{A}$ calculated on the gold standard corpus for different models of the $\mathcal{V}_{PL}$ variant with different numbers of topics.

6.4.4 Experiment III

To evaluate whether our approach can handle a larger number of datasets, we repeat the first two experiments but train the model on the complete LODStats dataset. In detail, this means that for every query dataset of the gold standard, we remove this dataset from the set of all 1 680 LODStats datasets. We train the variant $\mathcal{V}_{PL}$ of our approach on the 1 679 remaining datasets and calculate the similarity between the removed query dataset and the other 99 datasets contained in the model. After that we compare the similarities with the gold standard and search the similarity threshold that maximizes the F1-score. Using the $\mathcal{V}_{PL}$ document creation, the 1 680 documents of the complete LODStats dataset comprise 175 080 word tokens of 5 816 different word types.

Table 6.4.: F1-scores of TAPIOCA and the baselines on the complete LODStats corpus.

Approach	Classes	Properties	Both
TAPIOCA (log.)	—	**0.538**	—
tf-idf	0.103	0.436	0.444
BL_T	0.014	0.014	0.014
Lucene	0.214	0.241	0.385

Figure 6.10 shows the F1-score achieved by $\mathcal{V}_{PL}$. The maximum F1-score of 0.538 is achieved by a model with 284 topics. The results in Table 6.4 show that even with a much larger input our approach is able to achieve an F1-score that is higher than the scores of the baselines and comparable to the score achieved in the first experiment.

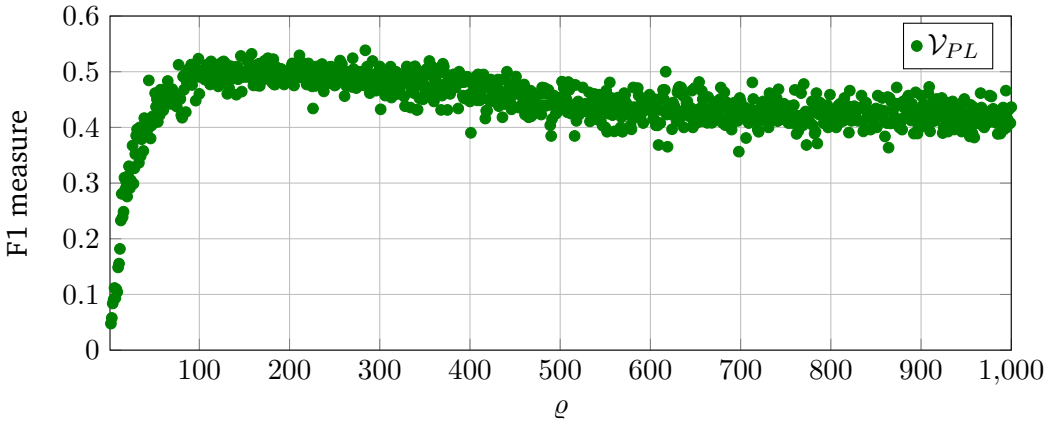


Figure 6.10.: The F1-scores of $\mathcal{V}_{PL}$ calculated on the complete LODStats corpus for different numbers of topics.

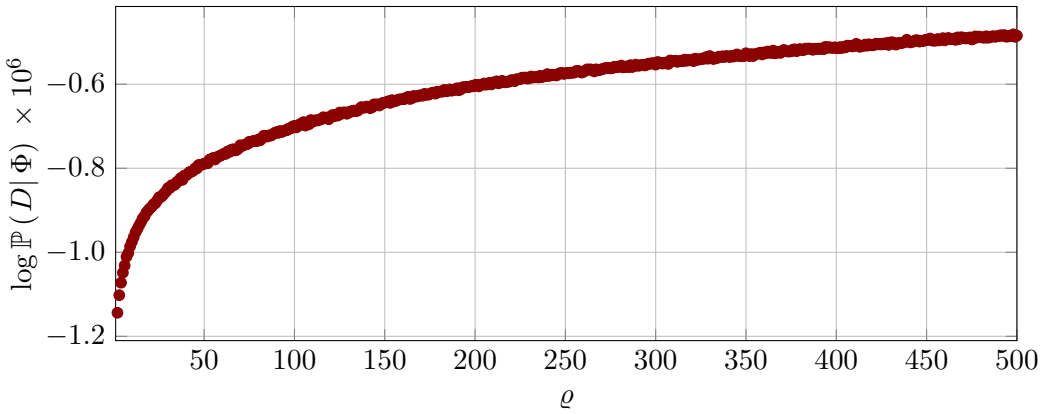


Figure 6.11.: Average $\log(\mathbb{P}(D|\Phi))$ calculated on the complete corpus for different models of the $\mathcal{V}_{PL}$ variant with different numbers of topics.

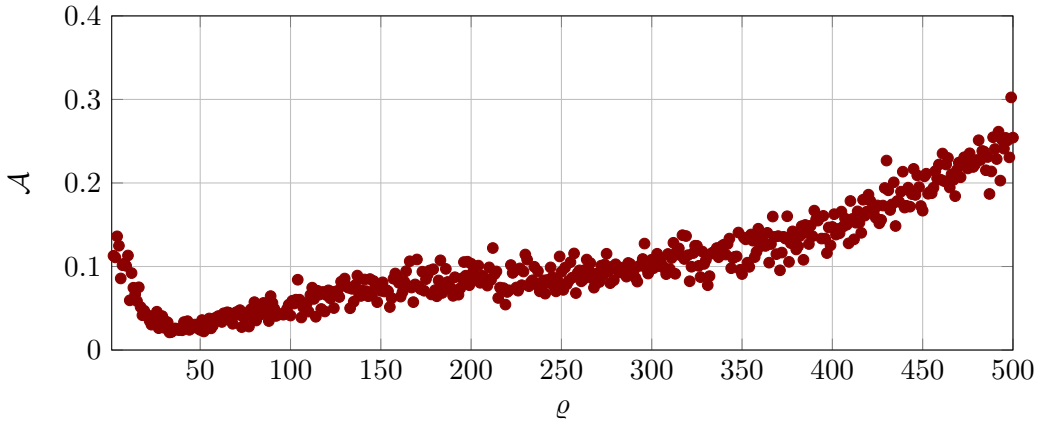


Figure 6.12.: Average values of the measure $\mathcal{A}$ calculated on the complete corpus for different models of the $\mathcal{V}_{PL}$ variant with different numbers of topics.

The Figures 6.11 and 6.12 show the average logarithm of $\mathbb{P}(D|\Phi)$ as well as the average value of the measure $\mathcal{A}$, respectively. While the average value of $\mathbb{P}(D|\Phi)$ increases steadily with a larger number of topics, the minimum of the average value of $\mathcal{A}$ is at 33 where the F1-score is only 0.336. However, since the gold standard is part of the dataset our search engine indexes, we can use it as a pragmatic way to determine a good number of topics. This pragmatic method assumes that a good topic model that has been trained on the datasets of the gold standard and additional datasets should give a high F1-score if it is compared to the gold standard. Thus, in practice we shall train multiple models with different numbers of topics on the same large dataset that comprises the gold standard datasets and use the model that achieves the highest F1-score compared to the gold standard.

6.4.5 Experiment IV

In this fourth experiment, we use GLISTEN to compare the logarithm-based variants of TAPIOCA with two of the aforementioned baselines, namely BL_T and *tf-idf*.

Dataset

We use the DBpedia [21, 162] as basis to create datasets from which we know that they are related to each other.[20] We gather the domain and range information for all `dbo` properties and all sub class and super class relations of the classes of the `dbo` namespace. Then, we use the domain and range as well as the class hierarchy information to add class information to single entities if they are not already explicitly stated. The result of this is a so-called materialized version of the DBpedia. We split this DBpedia into subsets according to the classes of entities. For each class that is high up in the class hierarchy and that has more than 10 000 instances, we extract the instances of the class and all triples in which they occur.[21] Since `dbo:Agent` has a comparatively large number of instances compared to the other classes, we use three of its subclasses instead, namely `dbo:Person`, `dbo:Organisation` and `dbo:FictionalCharacter`. We further preprocess the created subsets by removing triples that have a predicate with a different namespace than `rdf`, `rdfs`, and `dbo`. In addition, we only keep object properties since COPAAL does not make use of literals. Table 6.5 lists the 16 created datasets together with their size.

[20] We use the DBpedia dump of July 2021.

[21] We use the DBpedia dashboard to identify the top classes in the class hierarchy and to get their instance counts. See `https://github.com/dbpedia/gsoc-2020-dashboard/wiki` for details (last accessed on 13.08.2022).

Table 6.5.: The DBpedia classes for which subsets have been created, and the name and size of the subset in number of triples.

DBpedia ontology class	Dataset name	Dataset size
`dbo:Activity`	Activity	44 876
`dbo:Award`	Award	60 824
`dbo:Biomolecule`	Biomolecule	53 316
`dbo:ChemicalSubstance`	Chemical	47 498
`dbo:Device`	Device	121 405
`dbo:Event`	Event	1 452 273
`dbo:FictionalCharacter`	Fictional	210 448
`dbo:Language`	Language	74 332
`dbo:MeanOfTransportation`	Transport	506 162
`dbo:Organisation`	Organisation	4 367 669
`dbo:Person`	Person	25 760 123
`dbo:Place`	Place	8 667 962
`dbo:Species`	Species	1 788 086
`dbo:SportsSeason`	Sports	1 331 468
`dbo:TimePeriod`	Time	6 736 126
`dbo:Work`	Work	6 050 491

Table 6.6.: The properties that have been chosen to generate test data. For each property, the domain, range, and the number of triples in the query dataset are shown.

$\mathcal{G}_Q$	Property	Domain	Range	#triples
Person	dbo:careerStation	dbo:Person	dbo:CareerStation	1 563 181
	dbo:birthPlace	dbo:Animal	dbo:Place	1 410 036
	dbo:deathPlace	dbo:Animal	dbo:Place	401 378
	dbo:almaMater	dbo:Person	dbo:EducationalInstitution	199 945
	dbo:nationality	dbo:Person	dbo:Country	154 359
Place	dbo:politicalLeader	dbo:Place	dbo:PersonFunction	105 317
	dbo:locatedInArea	dbo:Place	dbo:Place	59 221
	dbo:routeJunction	dbo:RouteOfTransportation	dbo:Station	35 915
	dbo:architect	dbo:ArchitecturalStructure	dbo:Architect	27 281
	dbo:routeStart	dbo:RouteOfTransportation	dbo:Station	26 823

Setup

We use the Person and Place datasets as query datasets. For each query dataset, we use all other datasets as possible link candidates, i.e., they are the datasets that can be recommended by the different approaches given the query dataset. We use different values $\varrho = [2, 20]$ for the TAPIOCA variants and execute the whole TAPIOCA process (i.e., the creation of the index and the querying using the query dataset) three times since the index creation is based on random numbers that are used within the topic modeling inference algorithm. We combine the results of the three different models by calculating the arithmetic mean of the similarities between the query dataset and each of the recommended datasets. Table 6.7 shows the features of the corpora generated by the TAPIOCA variants.

Table 6.7.: Features of the corpora generated by the different TAPIOCA variants.

$\mathcal{G}_Q$	Variant	Word types	Word tokens
Person	$\mathcal{V}_{AL}$	79 393	769 077
	$\mathcal{V}_{CL}$	72 375	710 710
	$\mathcal{V}_{PL}$	11 516	87 143
Place	$\mathcal{V}_{AL}$	83 326	866 571
	$\mathcal{V}_{CL}$	76 222	809 643
	$\mathcal{V}_{PL}$	11 392	85 241

In addition, we create a test and a train dataset for each query dataset that is used to measure the performance of COPAAL as described in Section 6.3.1. To this end, we take the five properties, that occur the most in this dataset and that have a domain and range within the `dbo` namespace. Table 6.6 shows these chosen properties for the two query datasets. For each chosen property, we randomly select 20 triples that have the property as predicate. These chosen triples will be used as positive examples in the test dataset ($\mathscr{S}^+$) and are removed from the query dataset to form the training dataset. We create the same amount of negative examples similar to the approach presented by Gerber et al. [107]. For each chosen property p, we randomly select two entities s' and o' which fulfill the properties domain and range restrictions, respectively. If the triple (s', p, o') does not exist in the query dataset, it is used as negative example in the test dataset.[22] This leads to a test dataset ($\mathscr{S}$) with 100 true and 100 false triples for each query dataset.

We follow the GLISTEN workflow as described in Section 6.3. However, since all datasets already origin from the DBpedia, the linking and fusion are replaced with a trivial concatenation of the dataset's triples.

[22] It is worth noticing that this approach follows the closed world assumption.

Table 6.8.: The first five recommended datasets of the different approaches for the query datasets. For each variant of TAPIOCA, we show only the best performing parameterizations of the TAPIOCA variants (all have $\varrho = 2$, except $\mathcal{V}_{CL}$ with $\varrho = 4$ for the Person dataset).

$\mathcal{G}_Q$	N	BL_T	*tf-idf*	$\mathcal{V}_{AL}$	$\mathcal{V}_{CL}$	$\mathcal{V}_{PL}$
Person	1	Activity	Organisation	Organisation	Organisation	Time
	2	Device	Work	Place	Event	Award
	3	Language	Fictional	Time	Sports	Activity
	4	Chemical	Place	Sports	Time	Language
	5	Fictional	Event	Event	Place	Fictional
Place	1	Activity	Organisation	Organisation	Organisation	Organisation
	2	Device	Event	Person	Person	Sports
	3	Language	Person	Sports	Sports	Event
	4	Chemical	Transport	Time	Event	Person
	5	Transport	Work	Event	Time	Time

Results

```
1 dbr:Aaron_Manasses_McMillan dbo:almaMater dbr:Bishop_College .
2 dbr:2016_UCLA_shooting dbo:deathPlace dbr:Westwood,_Los_Angeles .
```

Listing 6.4: The two triples that are identified as true by COPAAL after adding the Organisation dataset.

```
1 dbo:almaMater/dbo:type/dbo:type
2 dbo:almaMater/dbo:country/^dbo:country
```

Listing 6.5: The q-restricted typed paths that give evidence for the first triple from Listing 6.4 (written as property paths following Harris et al. [116]).

Table 6.8 shows the first five recommended datasets. For the TAPIOCA variants, we report the results for the models that achieve the best performance. It can be seen that the different approaches lead to different recommendations. Table 6.9 shows the measured AUC-ROC of COPAAL for the query dataset ($N = 0$) and the fused datasets that include the recommended datasets for $N \leq 5$. The effect of fusing additional recommended datasets can be also seen in Figure 6.13 which shows the KPI difference ($\Delta_{\text{AUC-ROC}}@N$). For both query datasets, the effect of adding additional datasets is similar. Only a small number of datasets improve the performance of COPAAL. For the Person dataset, these are the Time, Organisation or Place datasets while COPAAL's performance on the Place dataset profits from fusing the Person or Organisation dataset. These improvements can be seen as an increase in the curve of a recommendation approach in Figure 6.13 and are typically caused

Table 6.9.: COPAAL's performance on the fused datasets for the first 5 ranks (N). The approaches share the result for $N = 0$. For each dataset, we also report the summary for the first 5 ranks for each approach ($\blacktriangle_{\text{AUC-ROC},5}$).

$\mathcal{G}_Q$	N	BL_T	*tf-idf*	$\mathcal{V}_{AL}$	$\mathcal{V}_{CL}$	$\mathcal{V}_{PL}$
Person	0			0.8435		
	1	0.8435	0.8537	0.8537	0.8537	0.9023
	2	0.8434	0.8537	0.8649	0.8536	0.9023
	3	0.8434	0.8537	0.9251	0.8537	0.9023
	4	0.8435	0.8651	0.9249	0.9125	0.9041
	5	0.8434	0.8655	0.9251	0.9251	0.9023
	$\blacktriangle_{\text{AUC-ROC},5}$	-0.0003	0.0742	0.2762	0.1811	**0.2958**
Place	0			0.8319		
	1	0.8319	0.8338	0.8338	0.8338	0.8338
	2	0.8317	0.8339	0.8395	0.8395	0.8338
	3	0.8319	0.8393	0.8397	0.8397	0.8338
	4	0.8319	0.8396	0.8397	0.8391	0.8391
	5	0.8317	0.8398	0.8397	0.8397	0.8397
	$\blacktriangle_{\text{AUC-ROC},5}$	-0.0004	0.0276	**0.0329**	0.0323	0.0207

by a small number of triples in $\mathscr{S}$. For example, the improvement after fusing the Organisation dataset is caused by the two triples shown in Listing 6.4. Both are true facts from the DBpedia.[23] For both facts, COPAAL is not able to identify any evidence based on the Person dataset. However, when fused with the Organization dataset, COPAAL finds two paths as evidence for each of these triples. Listing 6.5 shows the two q-restricted typed paths that give evidence for the first triple from Listing 6.4. The paths express that `dbr:Aaron_Manasses_McMillan` has several alma maters which share the same type of school and the same country.[24] Although these paths do not achieve high NPMI values with 0.36 and 0.40, respectively, they increase the triple's veracity score from 0.0 to 0.78. Similarly, the second triple's veracity score is increased from 0.0 to 0.62.[25] Improving the score of only one of these two triples leads to an increase of COPAAL's AUC-ROC from 0.8435 to 0.8484. Both improvements together lead to an AUC-ROC of 0.8533. A similar observation can be made when fusing the Place or Time dataset. The Place dataset leads to an improvement of 0.0112 while the Time dataset gives the highest improvement of

[23] The subject of the second triple has the type `dbo:Person` in the DBpedia. It seems like the person-related facts collected to this entity belong to the perpetrator of the crime.

[24] The Bishop College and the Meharry Medical College, which is another alma mater of Aaron Manasses McMillan, are both located in the United States and have the type "Historically black colleges and universities".

[25] The paths that are found for this triple are `dbo:almaMater/dbo:city` and `dbo:almaMater/^dbo:headquarter/dbo:location` with an NPMI of 0.57 and 0.49, respectively.

0.0588 AUC-ROC points. This improvement origins from identifying 12 facts with the `dbo:careerStation` as true for which no evidence could be found without the Time dataset. When the Place dataset is used as query dataset, the Organisation dataset gives an improvement of 0.0019 while fusing the Person dataset leads to a 0.0057 points better AUC-ROC value.

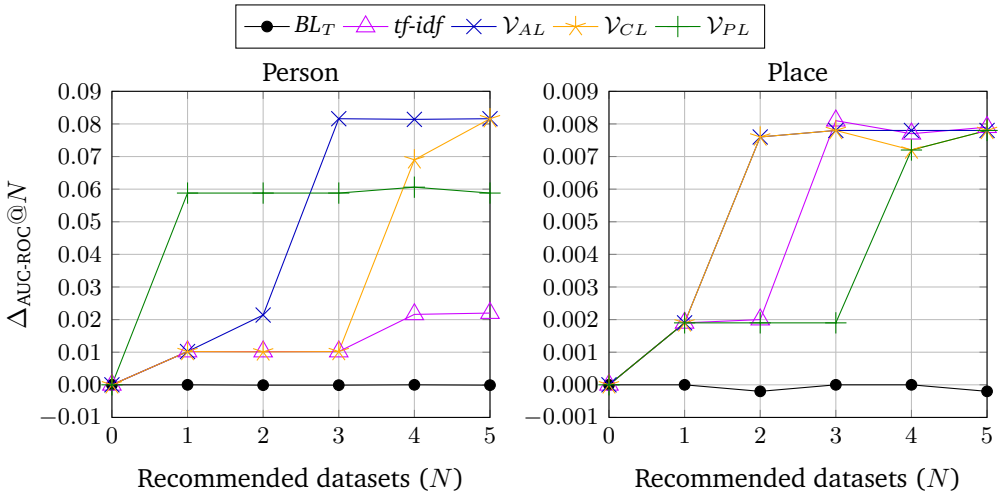


Figure 6.13.: The $\Delta_{\text{AUC-ROC}}@N$ values achieved by COPAAL based on the query dataset ($@N = 0$, Person left, Place right) and the datasets recommended by the different approaches. The diagrams only contain the best performing runs of the TAPIOCA variants (all have $\varrho = 2$, except $\mathcal{V}_{CL}$ with $\varrho = 4$ for the Person dataset).

Figure 6.13 shows that all three variants of TAPIOCA lead to an early improvement of the AUC-ROC value. The *tf-idf* approach leads to some improvements as well, especially for the Place dataset. However, for the Person dataset, the important Time dataset is ranked only as the 10-th dataset in the recommendation list of the *tf-idf* baseline. The first five recommendations of BL_T do not lead to any improvement for both query datasets.

We execute COPAAL on fusing all available datasets with the query dataset and achieve an AUC-ROC value of 0.9253 and 0.8397 for Person and Place, respectively. While all approaches except BL_T achieve this score with $N \leq 5$ for the Place dataset, only $\mathcal{V}_{AL}$ and $\mathcal{V}_{CL}$ recommend the three important datasets for the Person dataset within the first 5 ranks. Table 6.9 also shows the summary of the difference measure for the first 5 ranks ($\blacktriangle_{\text{AUC-ROC},5}$). $\mathcal{V}_{PL}$ achieves the highest $\blacktriangle_{\text{AUC-ROC},5}$ score for the Person dataset. At a first glance, this might be contradicting the statement that this variant does not achieve the best AUC-ROC score within $N \leq 5$. However, the $\blacktriangle_{,5}$ takes into account the rank of the recommendation, i.e., the earlier a good dataset is recommended, the more often its positive effect on the AUC-ROC value is added to

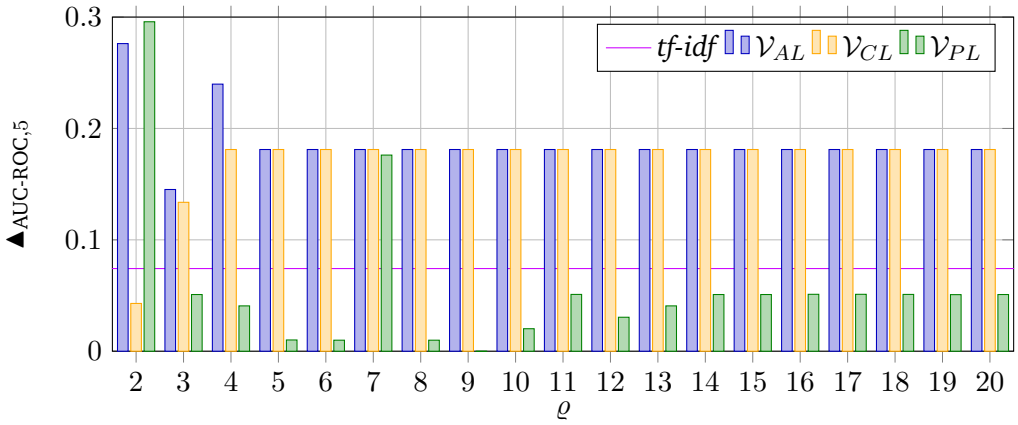


Figure 6.14.: The $\blacktriangle_{\text{AUC-ROC},5}$ values achieved by the log-based TAPIOCA variants for different numbers of topics with the Person query dataset. The value achieved by *tf-idf* is visualized as horizontal line in the background for comparison.

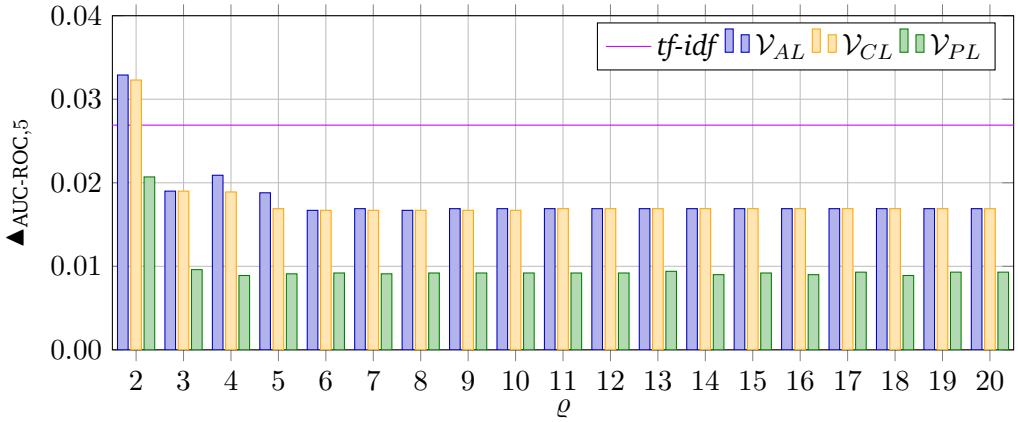


Figure 6.15.: The $\blacktriangle_{\text{AUC-ROC},5}$ values achieved by the log-based TAPIOCA variants for different numbers of topics with the Place query dataset. The value achieved by *tf-idf* is visualized as horizontal line in the background for comparison.

the score. The best performing $\mathcal{V}_{PL}$ variant has the Time dataset on the first place of its recommendation list. Because this dataset has the highest impact, the area under the $\Delta_{\text{AUC-ROC}}@N$ curve of $\mathcal{V}_{PL}$ shown in Figure 6.13 is slightly larger than the area under the curve of $\mathcal{V}_{AL}$. For the Place dataset, $\mathcal{V}_{AL}$ achieves the $\blacktriangle_{\text{AUC-ROC},5}$ best score. Figures 6.14 and 6.15 show the $\blacktriangle_{\text{AUC-ROC},5}$ values that the three TAPIOCA variants achieve for different ϱ values.

6.5 Conclusion

The aim of this chapter was twofold. First, we presented TAPIOCA—a search engine that tackles the problem of recommending topically similar datasets for linking. With this approach, we address the gap between creating an RDF dataset and linking it to other datasets. Our evaluation shows that our approach is better than several baselines and performs well on a large number of datasets. We could identify different parts of a datasets metadata and show that the properties are most important for determining the datasets topic. Second, we presented GLISTEN—the first extrinsic benchmark for dataset linking recommendation. It relies on an external task to measure the quality of the recommendations. Our evaluation showed that an extrinsic evaluation can be used to measure the performance of linking recommendation approaches. We used the task of Fact Checking but GLISTEN is not bound to that particular task.

A challenging future task is the search for a good number of topics that can be used to generate the topic model for TAPIOCA and that is not bound to the gold standard created by us. Besides this, another challenge is the handling of classes and properties that only have labels in foreign languages instead of English. With respect to the evaluation, we aim to include other tasks into GLISTEN.

7 Summary

Within this thesis, we looked at the Semantic Web and its two prerequesites to have an impact. It needs to have more data and the data needs to have a high quality [34, 251]. However, the growth of the Web increases the complexity of publishing more five-star datasets. Our work tackles this dilemma by looking at four research gaps. First, we look at the benchmarking of complex, distributed systems that process Linked Data and the need for fair benchmarks and benchmarking platforms (**RG4**). We propose HOBBIT—a holistic benchmarking platform that supports the benchmarking of all steps of the Linked Data life cycle [23]. This platform allows the benchmarking of distributed systems in a controlled environment. In addition, we propose LEMMING—an approach to generate synthetic knowledge graphs of arbitrary size that mimic real-world knowledge graphs. Both proposed approaches create the basis for our evaluations within this thesis.

Second, we address the distributed nature of the Semantic Web (**RG1**). We propose SQUIRREL—a distributed open-source crawler for the Data Web that supports a large set of formats of structured data and is built on a modularized architecture that allows the extension for future formats. We also present ORCA—the first extensible FAIR benchmark for Data Web crawlers, which measures the efficiency and effectiveness of crawlers in a comparable and repeatable way. Our evaluation shows that SQUIRREL outperforms a competing crawler with respect to the efficiency. In addition, SQUIRREL is able to crawl faster when provided with enough computational resources.

Third, we look at the creators of datasets and their need to be aware of already existing datasets (**RG2**). Searching for datasets on the Web is different to a classic Web search [65]. Existing solutions are mainly based on the dataset's meta data [53] or user-defined tags [217]. We propose LODCAT—an approach to support the exploration of the Data Web based on human-interpretable topics. LODCAT provides the user with human-interpretable topics that are automatically derived from a reference corpus and give the user an impression of a dataset's content. Additionally, we present PALMETTO—a framework for topic coherence measures. Based on this framework, we evaluate 555 660 coherence measures and identify two new topic

coherence measures. One of them performs better on all and the second on five out of six datasets than the previous state of the art.

Fourth, we address the situation of dataset publishers that need to know other, relevant datasets to which they can link their newly published dataset to (**RG3**). Addressing this gap is important to lift newly published datasets to five-star datasets [34] and, hence, improve the quality of the datasets on the Web. We propose TAPIOCA—a search engine that tackles the problem of recommending topically similar datasets for link discovery. Our evaluation shows that our approach is better than several baselines and performs well on a large number of datasets. We also present GLISTEN—the first extrinsic benchmark for dataset linking recommendation. It relies on an external task to measure the quality of the recommendations. Our evaluation shows that an extrinsic evaluation can be used to measure the performance of linking recommendation approaches.

Our work shows that the process of publishing data as five-star dataset can be eased with our presented solutions. Several steps can be either automated or the user can be supported with tools to ease the work. However, an adoption of the presented techniques may take some time. For example, a study from August 2016 showed that the usage of structured data in form of `schema.org` annotations on business web pages was limited. Only 17% of the participating marketers stated that they make use of these annotations to improve their ranking in search results [55]. This is surprising since the `schema.org` project exists since 2011 and is supported by major search engines. The study authors argue that the marketers might not be aware of this modern technique [55]. Hence, one of the next steps after easing the publication of data as five-star dataset could be to increase the awareness of non-experts about the possibilities that arise with the usage of machine-readable data.

Our work mainly focuses on Linked Open Data. Although the same techniques are interesting for other areas, we do not take the special needs of companies and private persons into account. A company may want to link its internal datasets with each other or with external datasets, without sharing its internal data with other parties. In a similar way, a private person might be interested in linking its private data with other datasets while keeping privacy. Future work will have to look at legal regulations like the General Data Protection Regulation of the European Union. It will also have to take the technical solutions for data accessibility into account. These may range from classic access control lists [253] to recent developments like the data pods of the Solid project [58].

Bibliography

[1] Using robots.txt. Website, Yandex Support. URL https://yandex.com/support/webmaster/controlling-robot/robots-txt.xml. Last time accessed, March 28th 2021.

[2] To crawl or not to crawl, that is BingBot's question. Website, Microsoft Bing Blog, March 2012. URL https://blogs.bing.com/Webmaster/2012/05/03/to-crawl-or-not-to-crawl-that-is-bingbots-question/. Last time accessed, March 28th 2021.

[3] Java Platform Standard Ed. 8: Class Math. Website, 2014. URL https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html. Last time accessed, May 18th, 2022.

[4] Data Never Sleeps 8.0. Website, Domo blog, 2020. URL https://www.domo.com/learn/data-never-sleeps-8. Last time accessed, January 12th, 2021.

[5] Microformats. Website, 2020. URL http://microformats.org/wiki/microformats. Last time accessed, June 16th, 2022.

[6] GS1 General Specifications. GS1 Standard, version 22.0, GS1, January 2022. URL https://www.gs1.org/genspecs.

[7] Lada A. Adamic and Eytan Adar. Friends and neighbors on the Web. *Social Networks*, 25(3):211–230, 2003. ISSN 0378-8733. doi: 10.1016/S0378-8733(03)00009-1. URL https://doi.org/10.1016/S0378-8733(03)00009-1.

[8] Ben Adida, Ivan Herman, Manu Sporny, and Mark Birbeck. RDFa 1.1 Primer – third edition. W3C Note, W3C, March 2015. URL http://www.w3.org/TR/rdfa-primer/.

[9] Réka Albert and Albert-László Barabási. Statistical Mechanics of Complex Networks. *Reviews of modern physics*, 74(1), 2002. doi: 10.1103/RevModPhys.74.47. URL https://doi.org/10.1103/RevModPhys.74.47.

[10] Nikolaos Aletras and Mark Stevenson. Evaluating Topic Coherence Using Distributional Ssemantics. In *Proceedings of the 10th International Conference*

on Computational Semantics (IWCS 2013) – Long Papers, pages 13–22, Potsdam, Germany, mar 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-0102.

[11] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing Linked Datasets with the VoID Vocabulary. W3C Note, W3C, March 2011. URL http://www.w3.org/TR/2011/NOTE-void-20110303/.

[12] Alsayed Algergawy, Michelle Cheatham, Daniel Faria, Alfio Ferrara, Irini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Naouel Karam, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Stefano Montanelli, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Daniela Schmidt, Pavel Shvaiko, Andrea Splendiani, Elodie Thiéblin, Cássia Trojahn, Jana Vataščinová, Ondřej Zamazal, and Lu Zhou. Results of the Ontology Alignment Evaluation Initiative 2018. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *Proceedings of the 13th International Workshop on Ontology Matching (OM 2018), Monterey, CA, USA, October 8, 2018.*, pages 76–116. CEUR-WS, 2018. URL http://ceur-ws.org/Vol-2288/oaei18_paper0.pdf.

[13] Alsayed Algergawy, Daniel Faria, Alfio Ferrara, Irini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Naouel Karam, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Stefano Montanelli, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Elodie Thiéblin, Cássia Trojahn, Jana Vataščinová, Ondrej Zamazal, and Lu Zhou. Results of the Ontology Alignment Evaluation Initiative 2019. In Pavel Shvaiko, editor, *OM 2019 : Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019) Auckland, New Zealand, October 26, 2019*, volume 2536, pages 46–85, Aachen, 2019. RWTH. URL https://madoc.bib.uni-mannheim.de/53428/.

[14] Awrad Mohammed Ali, Hamidreza Alvari, Alireza Hajibagheri, Kiran Lakkaraju, and Gita Sukthankar. Synthetic Generators for Cloning Social Network Data. In *Proceedings of the Fifth ASE International Conference on Big Data/SocialInformatics/PASSAT/BioMedCom*, Dec 2014. ISBN 978-1-62561-003-4. URL http://eecs.ucf.edu/~halvari/10.pdf.

[15] Loulwah Alsumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic Significance Ranking of LDA Generative Models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases:*

Part I, ECML PKDD '09, pages 67–82, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04179-2. doi: 10.1007/978-3-642-04180-8_22. URL http://dx.doi.org/10.1007/978-3-642-04180-8_22.

[16] Güneş Aluç, Olaf Hartig, M Tamer Özsu, and Khuzaima Daudjee. Diversified Stress Testing of RDF Data Management Systems. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 197–212, Cham, 2014. Springer, Springer International Publishing. ISBN 978-3-319-11964-9. doi: 10.1007/978-3-319-11964-9_13. URL https://link.springer.com/chapter/10.1007/978-3-319-11964-9_13.

[17] Carlos Buil Aranda, Olivier Corby, Souripriya Das, Lee Feigenbaum, Paula Gearon, Birte Glimm, Steve Harris, Sandro Hawke, Ivan Herman, Nicholas Humfrey, Nico Michaelis, Chimezie Ogbuji, Matthew Perry, Alexandre Passant, Axel Polleres, Eric Prud'hommeaux, Andy Seaborne, and Gregory Todd Williams. SPARQL 1.1 Overview. Recommendation, W3C, March 2013. URL http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/.

[18] Dörthe Arndt, Jeen Broekstra, Bob DuCharme, Ora Lassila, Peter F. Patel-Schneider, Eric Prud'hommeaux, Ted Jr. Thibodeau, and Bryan Thompson. RDF-star and SPARQL-star. Final Community Group Report, W3C, March 2021. URL https://www.w3.org/2021/12/rdf-star.html.

[19] R. Arun, V. Suresh, C. E Veni Madhavan, and M. Narasimha Murty. On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6118 LNAI(PART 1): 391–402, 2010. ISSN 03029743. doi: 10.1007/978-3-642-13657-3_43. URL https://link.springer.com/chapter/10.1007/978-3-642-13657-3_43.

[20] Hagai Attias. A Variational Baysian Framework for Graphical Models. In S. Solla, T. Leen, and K. Müller, editors, *Proceedings of the 12th International Conference on Neural Information Processing Systems*, volume 12 of *NIPS'99*, page 209–215, Cambridge, MA, USA, 1999. MIT Press. URL https://proceedings.neurips.cc/paper/1999/file/74563ba21a90da13dacf2a73e3ddefa7-Paper.pdf.

[21] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee,

Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, pages 722–735. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-76298-0. doi: 10.1007/978-3-540-76298-0_52. URL https://link.springer.com/chapter/10.1007/978-3-540-76298-0_52.

[22] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. LODStats – An Extensible Framework for High-Performance Dataset Analytics. In Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *Knowledge Engineering and Knowledge Management*, pages 353–362, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33876-2. doi: 10.1007/978-3-642-33876-2_31. URL https://link.springer.com/chapter/10.1007/978-3-642-33876-2_31.

[23] Sören Auer, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, and Amrapali Zaveri. Introduction to Linked Data and Its Lifecycle on the Web. In *Reasoning Web. Reasoning on the Web in the Big Data Era: 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, pages 1–99. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10587-1. doi: 10.1007/978-3-642-23032-5_1. URL https://link.springer.com/chapter/10.1007/978-3-319-10587-1_1.

[24] Ricardo A. Baeza Yates and Berthier R. Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 020139829X.

[25] Guillaume Bagan, Angela Bonifati, Radu Ciucanu, George H. L. Fletcher, Aurélien Lemay, and Nicky Advokaat. gMark: Schema-Driven Generation of Graphs and Queries. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):856–869, 2017. doi: 10.1109/TKDE.2016.2633993. URL https://ieeexplore.ieee.org/document/7762945.

[26] Peter Bailey, Ryen W. White, Han Liu, and Giridhar Kumaran. Mining Historic Query Trails to Label Long and Rare Search Engine Queries. *ACM Transactions on the Web*, 4(4), sep 2010. ISSN 1559-1131. doi: 10.1145/1841909.1841912. URL https://doi.org/10.1145/1841909.1841912.

[27] Jason Baldridge. The Apache OpenNLP project, 2005. URL http://opennlp.apache.org/index.html. Last time accessed, July 20th 2022.

[28] Albert-László Barábasi and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, Oct 1999. doi: 10.1126/science.286.

5439.509. URL https://www.science.org/doi/abs/10.1126/science.286.5439.509.

[29] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. RDF 1.1 Turtle. Recommendation, W3C, February 2014. URL http://www.w3.org/TR/2014/REC-turtle-20140225/.

[30] Wouter Beek, Laurens Rietveld, Hamid R. Bazoobandi, Jan Wielemaker, and Stefan Schlobach. LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 213–228, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11964-9. doi: 10.1007/978-3-319-11964-9_14. URL https://link.springer.com/chapter/10.1007/978-3-319-11964-9_14.

[31] Natalie Beisch and Wolfgang Koch. 25 Jahre ARD/ZDF-Onlinestudie: Unterwegsnutzung steigt wieder und Streaming/Mediatheken sind weiterhin Treiber des medialen Internets. *Media Perspektiven*, (10):486–503, 2021. URL https://www.ard-zdf-onlinestudie.de/files/2021/Beisch_Koch.pdf.

[32] Mohamed Ben Ellefi, Zohra Bellahsene, Stefan Dietze, and Konstantin Todorov. Dataset Recommendation for Data Linking: An Intensional Approach. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *The Semantic Web. Latest Advances and New Domains*, pages 36–51, Cham, 2016. Springer International Publishing. ISBN 978-3-319-34129-3. doi: 10.1007/978-3-319-34129-3_3. URL https://link.springer.com/chapter/10.1007/978-3-319-34129-3_3.

[33] Michael Bendersky and W. Bruce Croft. Discovering Key Concepts in Verbose Queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 491–498, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390419. URL https://doi.org/10.1145/1390334.1390419.

[34] Tim Berners-Lee. Linked Data. Website, June 2009. URL https://www.w3.org/DesignIssues/LinkedData.html. Last time accessed, January 4th, 2021.

[35] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001. URL https://www.jstor.org/stable/26059207.

[36] Tim Berners-Lee, Roy T. Fielding, and Larry M Masinter. Uniform Resource Identifier (URI): Generic Syntax. Technical Report 3986, Internet Engineering Task Force (IETF), January 2005. URL https://rfc-editor.org/rfc/rfc3986.txt.

[37] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Automatic Labelling of Topics with Neural Embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 953–963, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1091.

[38] Dania Bilal and Jacek Gwizdka. Children's query types and reformulations in Google search. *Information Processing & Management*, 54(6): 1022–1041, 2018. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2018.06.008. URL https://www.sciencedirect.com/science/article/pii/S0306457317308889.

[39] Christian Bizer and Andreas Schultz. The Berlin SPARQL Benchmark. *Int. J. Semantic Web Inf. Syst.*, 5(2):1–24, 2009. doi: 10.4018/jswis.2009040101.

[40] Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, and Johanna Völker. Deployment of rdfa, microdata, and microformats on the web–a quantitative analysis. In *International Semantic Web Conference*, pages 17–32. Springer, 2013.

[41] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782.

[42] David M. Blei. Probabilistic Topic Models: Origins and Challenges. Presentation slides, Website, December 2013. URL http://www.cs.columbia.edu/~blei/talks/Blei_Topic_Modeling_Workshop_2013.pdf. Last time accessed, August 3rd 2022.

[43] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435.

[44] Daniel Blum and Sara Cohen. Generating RDF for Application Testing. In *Proceedings of the 2010 International Conference on Posters & Demonstrations Track - Volume 658*, ISWC-PD'10, pages 105–108, Aachen, DEU, 2010. CEUR-WS.org.

[45] Paolo Boldi, Andrea Marino, Massimo Santini, and Sebastiano Vigna. Bubing: Massive crawling for the masses. *ACM Trans. Web*, 12(2), June 2018. ISSN 1559-1131.

[46] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie 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, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581026. doi: 10.1145/1376616.1376746. URL https://doi.org/10.1145/1376616.1376746.

[47] Angela Bonifati, Hannes Fletcher, George andVoigt, and Nikolay Yakovets. *Querying Graphs*. Number 51 in Synthesis Lectures on Data Management. Morgan & Claypool, 2018. ISBN 9781681734309. doi: 10.2200/S00873ED1V01Y201808DTM051. URL https://www.morganclaypool.com/doi/10.2200/S00873ED1V01Y201808DTM051.

[48] Elena Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. Case studies on ontology reuse. *Proceedings of the IKNOW05 International Conference on Knowledge Management*, 74:345–353, July 2005.

[49] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. In *From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, volume Normalized, pages 31–40, Tübingen, 2009.

[50] Luc Bovens and Stephan Hartmann. *Bayesian Epistemology*. Oxford University Press, 2003.

[51] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of Topic Models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296, 2017. ISSN 1554-0669. doi: 10.1561/1500000030. URL http://dx.doi.org/10.1561/1500000030.

[52] Dan Brickley, R.V. Guha, and Brian McBride. RDF Schema 1.1. W3C Note, W3C, February 2014. URL http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[53] Dan Brickley, Matthew Burgess, and Natasha Noy. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, WWW '19, page 1365–1375, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313685. URL https://doi.org/10.1145/3308558.3313685.

[54] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998. ISSN 0169-7552. doi: 10.1016/S0169-7552(98)00110-X. Proceedings of the Seventh International World Wide Web Conference.

[55] Mark Brozek. Prioritize Search To Maximize ROI Of Marketing. A Forrester Consulting Thought Leadership Paper commissioned by Microsoft and Catalyst, January 2017. URL `https://advertiseonbing.blob.core.windows.net/blob/bingads/media/library/insight/prioritize-search-to%20boost-roi/forrester-prioritize-search-whitepaper.pdf?ext=.pdf`. Last time accessed, July 24th, 2022.

[56] W.L. Buntine, J. Lofstrom, J. Perkio, S. Perttu, V. Poroshin, T. Silander, H. Tirri, A. Tuominen, and V. Tuulos. A scalable topic-based open source search engine. In *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on*, pages 228–234, Sept 2004. doi: 10.1109/WI.2004.10094.

[57] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. LodLive, Exploring the Web of Data. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, page 197–200, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311120. doi: 10.1145/2362499.2362532. URL `https://doi.org/10.1145/2362499.2362532`.

[58] Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, and Kjetil Kjernsmo. Solid Protocol. Specification, Solid Community Group, December 2021. URL `https://solidproject.org/TR/2021/protocol-20211217`. Version 0.9.0.

[59] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, and Kuansan Wang. ERD 2014: Entity recognition and disambiguation challenge. *SIGIR Forum*, 2014. doi: 10.1145/2701583.2701591.

[60] Jeremy J. Carroll and Jeff Z. Pan. Xml schema datatypes in rdf and owl. W3c working group note, W3C, March 2006. URL `http://www.w3.org/TR/2006/NOTE-swbp-xsch-datatypes-20060314/`.

[61] José M Cavanillas, Edward Curry, and Wolfgang Wahlster. *New horizons for a data-driven economy: a roadmap for usage and exploitation of big data in Europe*. Springer, 2016. doi: 10.1007/978-3-319-21569-3.

[62] Sheriffo Ceesay, Adam David Barker, and Blesson Varghese. Plug and Play Bench: Simplifying Big Data Benchmarking Using Containers. In *2017 IEEE International Conference on Big Data*, 2017. doi: 10.1109/BigData.

2017.8258249. URL https://ieeexplore.ieee.org/abstract/document/8258249.

[63] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf.

[64] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. Searchlens: Composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 498–509, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. doi: 10.1145/3301275.3302321. URL https://doi.org/10.1145/3301275.3302321.

[65] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. Dataset search: a survey. *The International Journal on Very Large Data Bases*, 29:251–272, 2020. doi: 10.1007/s00778-019-00564-x. URL https://doi.org/10.1007/s00778-019-00564-x.

[66] Philipp Cimiano, Vanessa López, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. Multilingual question answering over linked data (QALD-3): lab overview. In Pamela Forner, Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Visualization - 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings*, volume 8138 of *Lecture Notes in Computer Science*, pages 321–332. Springer, 2013. doi: 10.1007/978-3-642-40802-1_30. URL https://doi.org/10.1007/978-3-642-40802-1_30.

[67] Felix Conrads, Jens Lehmann, Muhammad Saleem, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. IGUANA: A generic framework for benchmarking the read-write performance of triple stores. In *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer International Publishing, 2017. doi: 10.1007/978-3-319-68204-4_5.

[68] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 249–260, New York,

NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488411. URL https://doi.org/10.1145/2488388.2488411.

[69] Simon Cox. RDF representation of 2016 edition of International Chronostratigraphic Chart (Geologic Timescale) v1. Data collection, CSIRO, 2017.

[70] Simon Cox. RDF representation of 2017 edition of International Chronostratigraphic Chart (Geologic Timescale) v3. Data collection, CSIRO, 2018.

[71] Simon Cox and Stephen Richard. RDF representation of International Chronostratigraphic Chart (Geologic Timescale) v2. Data collection, CSIRO, 2014.

[72] Simon Cox and Stephen Richard. RDF representation of 2018 edition of International Chronostratigraphic Chart (Geologic Timescale) v1. Data collection, CSIRO, 2019.

[73] Richard Cyganiak and Dave Reynolds. The rdf data cube vocabulary. W3c recommendation, W3C, January 2014. URL http://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/.

[74] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, W3C, February 2014. URL http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[75] Ana Alexandra Morim da Silva, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Using Compositional Embeddings for Fact Checking. In Andreas Hotho, Eva Blomqvist, Stefan Dietze, Achille Fokoue, Ying Ding, Payam Barnaghi, Armin Haller, Mauro Dragoni, and Harith Alani, editors, *The Semantic Web – ISWC 2021*, pages 270–286, Cham, 2021. Springer International Publishing. ISBN 978-3-030-88361-4. doi: 10.1007/978-3-030-88361-4_16. URL https://link.springer.com/chapter/10.1007/978-3-030-88361-4_16.

[76] Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, and Tzanina Saveta. Instance matching benchmarks in the era of linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2016. ISSN 1570-8268. doi: 10.1016/j.websem.2016.06.002.

[77] Jean Carlos Oliveira de Abreu, Renato Fileto, Axel-Cyrille Ngonga Ngomo, Michael Röder, Matthias Wittwer, and Horacio Saggion. Characterizing Mention Mismatching Problems for Improving Recognition Results. In *Proceedings of the 19th International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '17, page 85–94, New York, NY, USA, 2017.

Association for Computing Machinery. ISBN 9781450352994. doi: 10.1145/3151759.3151794. URL https://doi.org/10.1145/3151759.3151794.

[78] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, 7:1–30, dec 2006. ISSN 1532-4435. URL https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf.

[79] Abdelmoneim Amer Desouki, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Ranking on Very Large Knowledge Graphs. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, HT '19, page 163–171, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368858. doi: 10.1145/3342220.3343660. URL https://doi.org/10.1145/3342220.3343660.

[80] Abdelmoneim Amer Desouki, Felix Conrads, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Synthg: Mimicking rdf graphs using tensor factorization. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 76–79, 2021. doi: 10.1109/ICSC50631.2021.00017. URL https://ieeexplore.ieee.org/abstract/document/9364498.

[81] Anusuriya Devaraju and Shlomo Berkovsky. A hybrid recommendation approach for open research datasets. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, page 207–211, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355896. doi: 10.1145/3209219.3209250. URL https://doi.org/10.1145/3209219.3209250.

[82] Djellel Eddine Difallah, Andrew Pavlo, Carlo Curino, and Philippe Cudre-Mauroux. OLTP-Bench: An Extensible Testbed for Benchmarking Relational Databases. *Proceedings of the VLDB Endowment*, 7(4):277–288, December 2013. ISSN 2150-8097. doi: 10.14778/2732240.2732246. URL http://dx.doi.org/10.14778/2732240.2732246.

[83] Igor Douven and Wouter Meijs. Measuring coherence. *Synthese*, 156(3): 405–425, 2007. URL http://dx.doi.org/10.1007/s11229-006-9131-z.

[84] Kevin Dressler. LIMES manual: MLPipeline. Website, October 2019. URL https://dice-group.github.io/LIMES/#/developer_manual/ml_pipeline. Last time accessed, April 8th 2022.

[85] Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, and Octavian Udrea. Apples and oranges: A comparison of rdf benchmarks and real rdf datasets. In *Proceedings of the 2011 ACM SIGMOD International Conference on*

Management of Data, SIGMOD '11, page 145–156, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306614. doi: 10.1145/1989323.1989340. URL https://doi.org/10.1145/1989323.1989340.

[86] Martin J. Dürst and Michel Suignard. Internationalized Resource Identifiers (IRIs). Technical Report 3987, Internet Engineering Task Force (IETF), January 2005. URL https://rfc-editor.org/rfc/rfc3987.txt.

[87] Basil Ell, Denny Vrandečić, and Elena Simperl. Labels in the web of data. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 162–176. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-25073-6_11.

[88] Paul Erdös and Alfréd Rényi. On random graphs. i. *Publicationes Mathematicae*, 6:290–297, 1959.

[89] Orri Erling, Alex Averbuch, Josep Larriba-Pey, Hassan Chafi, Andrey Gubichev, Arnau Prat, Minh-Duc Pham, and Peter Boncz. The LDBC Social Network Benchmark: Interactive Workload. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 619–630, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450327589. doi: 10.1145/2723372.2742786. URL https://doi.org/10.1145/2723372.2742786.

[90] Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. Lodstats: The data web census dataset. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 38–46, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46547-0. doi: 10.1007/978-3-319-46547-0_5. URL https://doi.org/10.1007/978-3-319-46547-0_5.

[91] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Berlin, Heidelberg, 2013. ISBN 978-3-662-50042-2. doi: 10.1007/978-3-642-38721-0. URL https://doi.org/10.1007/978-3-642-38721-0.

[92] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1):77–129, 2018. doi: 10.3233/SW-170275.

[93] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.

2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[94] Javier D Fernández, Miguel A Martínez-Prieto, Pablo de la Fuente Redondo, and Claudio Gutiérrez. Characterizing RDF datasets. *Journal of Information Science*, 1:1–27, 2016.

[95] Javier D. Fernández, Wouter Beek, Miguel A. Martínez-Prieto, and Mario Arias. Lod-a-lot. In Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web – ISWC 2017*, pages 75–83, Cham, 2017. Springer International Publishing. ISBN 978-3-319-68204-4.

[96] Javier D. Fernández, Miguel A. Martínez-Prieto, Claudio Gutiérrez, Axel Polleres, and Mario Arias. Binary rdf representation for publication and exchange (hdt). *Web Semantics: Science, Services and Agents on the World Wide Web*, 19:22–41, 2013. URL http://www.websemanticsjournal.org/index.php/ps/article/view/328.

[97] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, 2005. doi: 10.3115/1219840.1219885.

[98] Rand Fishkin. The State of Searcher Behavior Revealed Through 23 Remarkable Statistics. Website, March 2017. URL https://moz.com/blog/state-of-searcher-behavior-revealed. Last time accessed, July 3rd, 2022.

[99] Branden Fitelson. A probabilistic theory of coherence. *Analysis*, 63(279): 194–199, 2003. ISSN 1467-8284. doi: 10.1111/1467-8284.00420. URL http://dx.doi.org/10.1111/1467-8284.00420.

[100] George Forman and Martin Scholz. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explor. Newsl.*, 12(1):49–57, November 2010. ISSN 1931-0145. doi: 10.1145/1882471.1882479. URL https://doi.org/10.1145/1882471.1882479.

[101] Martin Fowler. Inversion of Control Containers and the Dependency Injection pattern. Website, January 2004. URL https://martinfowler.com/articles/injection.html. Last time accessed, August 4th 2022.

[102] Irini Fundulaki. Deliverable 1.2.1: Requirements Specification from the Community. Project deliverable, HOBBIT – Holistic Benchmarking of Big Linked Data, 2016. URL http://project-hobbit.eu/about/deliverables/. Last

time accessed, December 15th, 2021. This publication is also available via the Community Research and Development Information Service of the European Commission using the grand agreement ID 688227. See https://cordis.europa.eu/project/rcn/199489/results/en.

[103] Kleanthi Georgala, Mirko Spasić, Milos Jovanovik, Henning Petzka, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MOCHA2017: The Mighty Storage Challenge at ESWC 2017. In *Semantic Web Challenges: Fourth SemWebEval Challenge at ESWC 2017*, 2017. doi: 10.1007/978-3-319-69146-6_1.

[104] Kleanthi Georgala, Mirko Spasić, Milos Jovanovik, Vassilis Papakonstantinou, Claus Stadler, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MOCHA2018: The Mighty Storage Challenge at ESWC 2018. In Davide Buscaldi, Aldo Gangemi, and Diego Reforgiato Recupero, editors, *Semantic Web Challenges*, pages 3–16, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00072-1. doi: 10.1007/978-3-030-00072-1_1.

[105] Kleanthi Georgala, Michael Röder, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. Applying edge-counting semantic similarities to Link Discovery: Scalability and Accuracy. In *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020)*. CEUR-WS.org, 2020. URL https://papers.dice-research.org/2020/OM_hECATE/public.pdf.

[106] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Bootstrapping the Linked Data Web. In *1st Workshop on Web Scale Knowledge Extraction @ISWC*, 2011. URL http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/Workshops/WeKEx/paper_3.pdf.

[107] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. DeFacto—temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:85–101, 2015.

[108] Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. BigBench: Towards an Industry Standard Benchmark for Big Data Analytics. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1197–1208, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2037-5. doi: 10.1145/2463676.2463712. URL http://doi.acm.org/10.1145/2463676.2463712.

[109] Jim Gray and Charles Levine. Thousands of DebitCredit Transactions-Per-Second: Easy and Inexpensive. *arXiv preprint cs/0701161*, 2007. URL https://arxiv.org/abs/cs/0701161.

[110] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004. doi: 10.1073/pnas.0307752101.

[111] R. V. Guha, Dan Brickley, and Steve Macbeth. Schema.org: Evolution of structured data on the web. *Commun. ACM*, 59(2):44–51, jan 2016. ISSN 0001-0782. doi: 10.1145/2844544. URL https://doi.org/10.1145/2844544.

[112] Vincenzo Gulisano, Zbigniew Jerzak, Roman Katerinenko, Martin Strohbach, and Holger Ziekow. The debs 2017 grand challenge. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, DEBS '17, pages 271–273. ACM, 2017. doi: 10.1145/3093742.3096342.

[113] Vincenzo Gulisano, Zbigniew Jerzak, Pavel Smirnov, Martin Strohbach, Holger Ziekow, and Dimitris Zissis. The debs 2018 grand challenge. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 191–194. ACM, 2018. ISBN 978-1-4503-5782-1. doi: 10.1145/3210284.3220510. URL http://doi.acm.org/10.1145/3210284.3220510.

[114] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics*, 3(2):158–182, 2005. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2005.06.005. URL http://www.sciencedirect.com/science/article/pii/S1570826805000132. Selected Papers from the International Semantic Web Conference, 2004.

[115] Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, page 44–52. Association for Computational Linguistics, June 2013.

[116] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. SPARQL 1.1 Query Language. Recommendation, W3C, March 2013. URL http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[117] Andreas Harth. Billion Triples Challenge data set. Website and dataset, 2012. URL http://km.aifb.kit.edu/projects/btc-2012/. Last time accessed, November 11th 2021.

[118] Andreas Harth, Jürgen Umbrich, and Stefan Decker. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora M. Aroyo, editors, *The Semantic Web - ISWC 2006*, pages 258–271, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-49055-5.

[119] Patrick J. Hayes and Peter F. Patel-Schneider. RDF 1.1 Semantics. W3C Recommendation, W3C, February 2014. URL http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/.

[120] James Hendler. Where Are All the Intelligent Agents? *IEEE Intelligent Systems*, 22(03):2–3, 2007. ISSN 1941-1294. doi: 10.1109/MIS.2007.62.

[121] Daniel M. Herzig, Peter Mika, Roi Blanco, and Thanh Tran. Federated entity search using on-the-fly consolidation. In *The Semantic Web - ISWC 2013*, pages 167–183, 2013.

[122] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *Word Wide Web*, 1999.

[123] Alexander Hinneburg, Rico Preiss, and René Schröder. Topicexplorer: Exploring document collections with topic models. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 838–841, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33486-3. doi: 10.1007/978-3-642-33486-3_59.

[124] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer (Second Edition). Recommendation, W3C, December 2012. URL http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[125] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering Emerging Entities with Ambiguous Names. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 385–396, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2568003. URL http://doi.acm.org/10.1145/2566486.2568003.

[126] Matthew Hoffman, Francis Bach, and David Blei. Online Learning for Latent Dirichlet Allocation. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and

A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf.

[127] Aidan Hogan. *Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora*. PhD thesis, National University of Ireland, Galway, 2011. URL http://aidanhogan.com/docs/thesis/.

[128] Aidan Hogan, Andreas Harth, Jürgen Umbrich, Sheila Kinsella, Axel Polleres, and Stefan Decker. Searching and browsing linked data with SWSE: The semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):365 – 401, 2011. ISSN 1570-8268. doi: 10.1016/j.websem.2011.06.004. URL http://www.sciencedirect.com/science/article/pii/S1570826811000473. JWS special issue on Semantic Search.

[129] Aidan Hogan, Andreas Harth, Juergen Umrich, Sheila Kinsella, Axel Polleres, and Stefan Decker. Searching and browsing linked data with swse: the semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4), 2011.

[130] Aidan Hogan, Jürgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. An empirical survey of linked data conformance. *Journal of Web Semantics*, 14:14 – 44, 2012. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2012.02.001. URL http://www.sciencedirect.com/science/article/pii/S1570826812000352. Special Issue on Dealing with the Messiness of the Web of Data.

[131] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data. Morgan & Claypool, 2021. doi: 10.2200/S01125ED1V01Y202109DSK022. URL https://kgbook.org/.

[132] Bernadette Hyland, Ghislain Atemezing, Michael Pendleton, and Biplav Srivastava. Linked data glossary. W3C Working Group Note, W3C, June 2013. URL http://www.w3.org/TR/ld-glossary/.

[133] Robert Isele, Jürgen Umbrich, Christian Bizer, and Andreas Harth. LDspider: An open-source crawling framework for the Web of Linked Data. In *Proceed-*

ings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, volume 658, pages 29–32. CEUR-WS, 2010.

[134] ISO. 26324:2012 Information and documentation – Digital object identifier system. Standard, International Organization for Standardization, May 2012. URL https://www.iso.org/standard/43506.html.

[135] Rricha Jalota, Nikit Srivastava, Daniel Vollmers, René Speck, Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. *Finding datasets in publications: The University of Paderborn approach*, pages 129–141. SAGE Publications Ltd, 2020. ISBN 978-1-5297-0586-7. URL https://uk.sagepub.com/en-gb/eur/rich-search-and-discovery-for-research-datasets/book270223#contents.

[136] Glen Jeh and Jennifer Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 538–543, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775126. URL https://doi.org/10.1145/775047.775126.

[137] Kunal Jha, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Eaglet – a Named Entity Recognition and Entity Linking Gold Standard Checking Tool. In Eva Blomqvist, Katja Hose, Heiko Paulheim, Agnieszka Ławrynowicz, Fabio Ciravegna, and Olaf Hartig, editors, *The Semantic Web: ESWC 2017 Satellite Events*, pages 149–154, Cham, 2017. Springer International Publishing. ISBN 978-3-319-70407-4. doi: 10.1007/978-3-319-70407-4_28. URL https://link.springer.com/chapter/10.1007/978-3-319-70407-4_28.

[138] Kunal Jha, Michael Röder, and Axel-Cyrille Ngonga Ngomo. All That Glitters is not Gold – Rule-Based Curation of Reference Datasets for Named Entity Recognition and Entity Linking. In *The Semantic Web. Latest Advances and New Domains: 14th International Conference, ESWC 2017, Proceedings*. Springer International Publishing, 2017. URL https://svn.aksw.org/papers/2017/ESWC_EAGLET_2017/public.pdf.

[139] Ernesto Jiménez-Ruiz, Tzanina Saveta, Ondrej Zamazal, Sven Hertling, Michael Röder, Irini Fundulaki, Axel Ngonga Ngomo, Mohamed Sherif, Amina Annane, Zohra Bellahsene, Sadok Ben Yahia, Gayo Diallo, Daniel Faria, Marouen Kachroudi, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Maximilian Mackeprang, Majid Mohammadi, Maciej Rybinski, Booma Sowkarthiga Balasubramani, and Cássia Trojahn. Introducing the HOBBIT platform into

the ontology alignment evaluation campaign. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *Proceedings of the 13th International Workshop on Ontology Matching (OM 2018), Monterey, CA, USA, October 8, 2018.*, pages 49–60. CEUR-WS, 2018. URL http://ceur-ws.org/Vol-2288/om2018_LTpaper5.pdf.

[140] Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong. LinkGen: Multipurpose Linked Data Generator. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 113–121, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46547-0. doi: 10.1007/978-3-319-46547-0_12. URL https://doi.org/10.1007/978-3-319-46547-0_12.

[141] Dan Jurafsky and James H. Martin. *Speech and Language Processing*. 3rd (draft) edition, December 2020. URL http://stanford.edu/~jurafsky/slp3/. Last time accessed, July 20th, 2021.

[142] Richard M. Karp. *Reducibility Among Combinatorial Problems*, pages 85–103. Plenum Press, 1972.

[143] Loe Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, March 1953. doi: 10.1007/BF02289026.

[144] Gregg Kellogg. Microdata – second edition. W3C Note, W3C, December 2014. URL https://www.w3.org/TR/microdata-rdf/.

[145] Simon Kemp. Digital 2022: Global Overview Report. Website, January 2022. URL https://datareportal.com/reports/digital-2022-global-overview-report. Last time accessed, January 17th, 2022.

[146] Rohit Khare, Doug Cutting, Kragen Sitaker, and Adam Rifkin. Nutch: A flexible and scalable open-source web search engine. *Oregon State Uni.*, 1, 2004.

[147] James G. Kim and Michael Hausenblas. 5 ⋆ open data. Website, 2012. URL https://5stardata.info/en/. Last time accessed, January 4th, 2021.

[148] Wooju Kim, Dae Woo Choi, and Sangun Park. Product Information Metasearch Framework for Electronic Commerce Through Ontology Mapping. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *The Semantic Web: Research and Applications*, pages 408–422, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31547-6.

[149] Vasilis Kopsachilis and Michail Vaitis. Geolod: A spatial linked data catalog and recommender. *Big Data and Cognitive Computing*, 5(2), 2021. ISSN 2504-2289. doi: 10.3390/bdcc5020017. URL https://www.mdpi.com/2504-2289/5/2/17.

[150] Vasilis Kopsachilis, Michail Vaitis, Nikos Mamoulis, and Dimitris Kotzinos. Recommending Geo-semantically Related Classes for Link Discovery. *Journal on Data Semantics*, 9:151–177, 2020. ISSN 1861-2040. doi: 10.1007/s13740-020-00117-4. URL https://doi.org/10.1007/s13740-020-00117-4.

[151] M. Koster, G. Illyes, H. Zeller, and L. Harvey. Robots Exclusion Protocol. Internet-draft, Internet Engineering Task Force (IETF), July 2019. URL https://tools.ietf.org/html/draft-rep-wg-topic-00.

[152] Wanqiu Kou, Fang Li, and Timothy Baldwin. Automatic labelling of topic models using word vectors and letter trigram vectors. In Guido Zuccon, Shlomo Geva, Hideo Joho, Falk Scholer, Aixin Sun, and Peng Zhang, editors, *Information Retrieval Technology*, pages 253–264, Cham, 2015. Springer International Publishing.

[153] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82, Sep 2010. doi: 10.1103/PhysRevE.82.036106. URL https://link.aps.org/doi/10.1103/PhysRevE.82.036106.

[154] Sven Kuhlmann. Benchmarking of Dataset Linkage Recommendation Systems based on real-world Linked Data Application Scenarios. Master's thesis, Paderborn University, October 2020.

[155] S.R. Kunze and S. Auer. Dataset retrieval. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 1–8, Sept 2013. doi: 10.1109/ICSC.2013.12.

[156] Tobias Käfer, Andreas Harth, Andrei Ciortea, and Victor Charpenay. All the Agents Challenge: Preface. In Tobias Käfer, Andreas Harth, Andrei Ciortea, and Victor Charpenay, editors, *Proceedings of the All the Agents Challenge (ATAC 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021)*, volume 3111 of *CEUR Workshop Proceedings*. CEUR-WS.org, October 2021. URL http://ceur-ws.org/Vol-3111/xpreface.pdf.

[157] Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In *Proceedings of the 23rd International*

Conference on Computational Linguistics: Posters, COLING '10, page 605–613, USA, 2010. Association for Computational Linguistics.

[158] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, page 1536–1545, USA, 2011. Association for Computational Linguistics. ISBN 9781932432879.

[159] Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2014.

[160] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR. URL `http://proceedings.mlr.press/v32/le14.html`.

[161] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The PROV Ontology. W3C Recommendation, W3C, April 2013. URL `http://www.w3.org/TR/2013/REC-prov-o-20130430/`.

[162] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015. doi: 10.3233/SW-140134.

[163] Luiz André P. Paes Leme, Giseli Rabello Lopes, Bernardo Pereira Nunes, Marco Antonio Casanova, and Stefan Dietze. Identifying Candidate Datasets for Data Interlinking. In Florian Daniel, Peter Dolog, and Qing Li, editors, *Web Engineering*, volume 7977 of *Lecture Notes in Computer Science book series (LNCS)*, pages 354–366, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-39200-9. doi: 10.1007/978-3-642-39200-9_29. URL `https://doi.org/10.1007/978-3-642-39200-9_29`.

[164] Douglas B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, Nov 1995. ISSN 0001-0782. doi: 10.1145/219717.219745. URL `https://doi.org/10.1145/219717.219745`.

[165] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 177–187, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081893. URL https://doi.org/10.1145/1081870.1081893.

[166] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.*, 14(1):50–60, Sep 2020. ISSN 2150-8097. doi: 10.14778/3421424.3421431. URL https://doi.org/10.14778/3421424.3421431.

[167] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568.

[168] Haichi Liu, Ting Wang, and Jintao Tang. Prediction of Datasets sameAs Interlinking on Web of Data. *Journal of Web Systems and Applications*, 1: 25–29, 2017. doi: 10.23977/jwsa.2017.11005.

[169] Giseli Rabello Lopes, Luiz André P. Paes Leme, Bernardo Pereira Nunes, Marco Antonio Casanova, and Stefan Dietze. Recommending Tripleset Interlinking through a Social Network Approach. In Xuemin Lin, Yannis Manolopoulos, Divesh Srivastava, and Guangyan Huang, editors, *Web Information Systems Engineering – WISE 2013*, volume 8180 of *Lecture Notes in Computer Science book series (LNCS)*, pages 149–161, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41230-1. doi: 10.1007/978-3-642-41230-1_13. URL https://doi.org/10.1007/978-3-642-41230-1_13.

[170] Giseli Rabello Lopes, Luiz André P. Paes Leme, Bernardo Pereira Nunes, Marco Antonio Casanova, and Stefan Dietze. Two Approaches to the Dataset Interlinking Recommendation Problem. In Boualem Benatallah, Azer Bestavros, Yannis Manolopoulos, Athena Vakali, and Yanchun Zhang, editors, *Web Information Systems Engineering – WISE 2014*, volume 8786 of *Lecture Notes in Computer Science book series (LNCS)*, pages 324–339, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11749-2. doi: 10.1007/978-3-319-11749-2_25. URL https://link.springer.com/chapter/10.1007/978-3-319-11749-2_25.

[171] Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203, 2011. ISSN 1386-4564.

[172] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011. ISSN 0378-4371. doi: 10.1016/j.physa.2010.11.027.

[173] Fadi Maali and John Erickson. Data Catalog Vocabulary (DCAT). W3c recommendation, W3C, January 2014. URL `https://www.w3.org/TR/2014/REC-vocab-dcat-20140116/`.

[174] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *7th Biennial Conference on Innovative Data Systems Research (CIDR 2015)*, 2015.

[175] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics, 2014. URL `http://aclweb.org/anthology/P/P14/P14-5010.pdf`.

[176] Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. 2002. URL `http://mallet.cs.umass.edu`.

[177] John P. McCrae. The Linked Open Data Cloud. Website, May 2021. URL `https://www.lod-cloud.net/`. Last time accessed, August 24th 2021.

[178] Muntazir Mehdi, Aftab Iqbal, Aidan Hogan, Ali Hasnain, Yasar Khan, Stefan Decker, and Ratnesh Sahay. Discovering Domain-Specific Public SPARQL Endpoints: A Life-Sciences Use-Case. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, IDEAS '14, page 39–45, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450326278. doi: 10.1145/2628194.2628220. URL `https://doi.org/10.1145/2628194.2628220`.

[179] Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7th International Conference on Semantic Systems (I-Semantics)*, 2011. doi: 10.1145/2063518.2063519.

[180] Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014. ISSN 1075-3583. URL http://dl.acm.org/citation.cfm?id=2600239.2600241.

[181] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[182] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38 (11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

[183] David Milne and Ian H. Witten. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 509–518, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939913. doi: 10.1145/1458082.1458150. URL https://doi.org/10.1145/1458082.1458150.

[184] David Mimno and David Blei. Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 227–237. Association for Computational Linguistics, 2011.

[185] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

[186] Maryam Mohsin. Google search statistics you need to know in 2022. Website, Jan 2022. URL https://www.oberlo.com/blog/google-search-statistics. Last time accessed, July 3rd, 2022.

[187] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel-Cyrille Ngonga Ngomo. DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, pages 454–469, 2011. doi: 10.1007/978-3-642-25073-6_29.

[188] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel-Cyrille Ngonga Ngomo. Usage-Centric Benchmarking of RDF Triple Stores. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012. URL http://jens-lehmann.org/files/2012/aaai_dbpedia_benchmark.pdf.

[189] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. OWL 2 Web Ontology Language Direct Semantics (Second Edition). Recommendation, W3C, December 2012. URL http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/.

[190] Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In *K-CAP 2017: Knowledge Capture Conference*. ACM, 2017. doi: 10.1145/3148011.3148024.

[191] Diego Moussallem, Ricardo Usbeck, Michael Röeder, and Axel-Cyrille Ngonga Ngomo. MAG: A Multilingual, Knowledge-Base Agnostic and Deterministic Entity Linking Approach. In *Proceedings of the Knowledge Capture Conference*, K-CAP 2017, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450355537. doi: 10.1145/3148011.3148024. URL https://svn.aksw.org/papers/2017/KCAP_MAG/public.pdf.

[192] Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Entity Linking in 40 Languages Using MAG. In Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z Pan, and Mehwish Alam, editors, *The Semantic Web: ESWC 2018 Satellite Events*, pages 176–181, Cham, 2018. Springer International Publishing. ISBN 978-3-319-98192-5. doi: 10.1007/978-3-319-98192-5_33. URL https://link.springer.com/chapter/10.1007/978-3-319-98192-5_33.

[193] Diego Moussallem, Paramjot Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, Michael Röder, René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. A General Benchmarking Framework for Text Generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 27–33, Dublin, Ireland (Virtual), 12 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.webnlg-1.3.

[194] Diego Moussallem, Paramjot Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, Michael Röder, René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. A General Benchmarking Framework for Text Generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 27–33, Dublin, Ireland (Virtual), 12 2020. Associa-

tion for Computational Linguistics. URL https://aclanthology.org/2020.webnlg-1.3.

[195] Stephen Muggleton and Luc de Raedt. Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679, 1994. ISSN 0743-1066. doi: https://doi.org/10.1016/0743-1066(94)90035-3. URL https://www.sciencedirect.com/science/article/pii/0743106694900353. Special Issue: Ten Years of Logic Programming.

[196] David Nadeau. Balie–baseline information extraction: Multilingual information extraction from text with machine learning and natural language techniques. Technical report, Technical report, University of Ottawa, 2005. URL http://balie.sourceforge.net/dnadeau05balie.pdf.

[197] Giulio Napolitano, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. The scalable question answering over linked data (sqa) challenge 2018. In Davide Buscaldi, Aldo Gangemi, and Diego Reforgiato Recupero, editors, *Semantic Web Challenges*, pages 69–75, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00072-1. doi: 10.1007/978-3-030-00072-1_6.

[198] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semantic Web*, 8(3): 419–436, 2017. doi: 10.3233/SW-150210.

[199] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[200] Axel-Cyrille Ngonga Ngomo, Sören Auer, Jens Lehmann, and Amrapali Zaveri. Introduction to Linked Data and Its Lifecycle on the Web. In Manolis Koubarakis, Giorgos Stamou, Giorgos Stoilos, Ian Horrocks, Phokion Kolaitis, Georg Lausen, and Gerhard Weikum, editors, *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, pages 1–99, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10587-1. doi: 10.1007/978-3-319-10587-1_1. URL https://link.springer.com/chapter/10.1007/978-3-319-10587-1_1.

[201] Axel-Cyrille Ngonga Ngomo and Michael Röder. HOBBIT: Holistic Benchmarking for Big Linked Data. In *ESWC, EU networking session*, 2016. URL http://svn.aksw.org/papers/2016/ESWC_HOBBIT_EUNetworking/public.pdf.

[202] Axel-Cyrille Ngonga Ngomo, Michael Röder, and Ricardo Usbeck. Cross-Document Coreference Resolution Using Latent Features. In *Proceedings of the Second International Conference on Linked Data for Information Extraction - Volume 1267*, LD4IE'14, page 33–44, Aachen, DEU, 2014. CEUR-WS.org. URL https://dl.acm.org/doi/abs/10.5555/2878575.2878580.

[203] Axel-Cyrille Ngonga Ngomo, Michael Röder, Diego Moussallem, Ricardo Usbeck, and René Speck. BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 339–349, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6541. URL https://aclanthology.org/W18-6541.

[204] Axel-Cyrille Ngonga Ngomo, Michael Röder, and Zafar Habeeb Syed. Semantic web challenge 2019. Website, 2019. URL https://github.com/dice-group/semantic-web-challenge.github.io/. Last time accessed, March 30th 2022.

[205] Axel-Cyrille Ngonga Ngomo, Mohamed Ahmed Sherif, Kleanthi Georgala, Mofeed Hassan, Kevin Dreßler, Klaus Lyko, Daniel Obraczka, and Tommaso Soru. LIMES - A Framework for Link Discovery on the Semantic Web. *KI - Künstliche Intelligenz, German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.*, 2021. doi: 10.1007/s13218-021-00713-x. URL https://papers.dice-research.org/2021/KI_LIMES/public.pdf.

[206] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-Scale Knowledge Graphs: Lessons and Challenges. *Commun. ACM*, 62(8):36–43, jul 2019. ISSN 0001-0782. doi: 10.1145/3331166. URL https://doi.org/10.1145/3331166.

[207] Erik Olsson. What is the problem of coherence and truth? *The Journal of Philosophy*, 99(5):246–272, 2002.

[208] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[209] Vassilis Papakonstantinou, Irini Fundulaki, and Giorgos Flouris. Second version of the versioning benchmark. In *Holistic Benchmarking of Big Linked Data*, 2018.

[210] Andrea Papenmeier, Alfred Sliwa, Dagmar Kern, Daniel Hienert, Ahmet Aker, and Norbert Fuhr. *'A Modern Up-To-Date Laptop' - Vagueness in Natural Language Queries for Product Search*, page 2077–2089. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450369749. URL https://doi.org/10.1145/3357236.3395489.

[211] Andrea Papenmeier, Dagmar Kern, Daniel Hienert, Alfred Sliwa, Ahmet Aker, and Norbert Fuhr. Starting conversations with search engines - interfaces that elicit natural language queries. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, CHIIR '21, page 261–265, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380553. doi: 10.1145/3406522.3446035. URL https://doi.org/10.1145/3406522.3446035.

[212] Peter F. Patel-Schneider and Boris Motik. OWL 2 Web Ontology Language Mapping to RDF Graphs (Second Edition). Recommendation, W3C, December 2012. URL http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/.

[213] Harshal Patni. Linkedsensordata. Website in the web archive, September 2010. URL https://web.archive.org/web/20190816202119/http://wiki.knoesis.org/index.php/SSW_Datasets. Last time accessed, May 11th, 2022.

[214] Harshal Patni, Cory Henson, and Amit Sheth. Linked sensor data. In *2010 International Symposium on Collaborative Technologies and Systems*, pages 362–370, 2010. doi: 10.1109/CTS.2010.5478492.

[215] Heiko Paulheim and Sven Hertling. Discoverability of SPARQL Endpoints in Linked Open Data. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, volume 1035, pages 245–248, Aachen, Germany, Germany, 2013. CEUR-WS.org. URL http://ceur-ws.org/Vol-1035/iswc2013_poster_17.pdf.

[216] Heiko Paulheim, Axel-Cyrille Ngonga Ngomo, and Dan Bennett. Semantic web challenge 2018. Website, 2018. URL http://iswc2018.semanticweb.org/semantic-web-challenge-2018/index.html. Last time accessed, March 30th 2022.

[217] Emmanuel Pietriga, Hande Gözükan, Caroline Appert, Marie Destandau, Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Browsing Linked Data Catalogs with LODAtlas. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée

Kaffee, and Elena Simperl, editors, *The Semantic Web – ISWC 2018*, pages 137–153, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00668-6. doi: 10.1007/978-3-030-00668-6_9.

[218] Mina Abd Nikooie Pour, Alsayed Algergawy, Reihaneh Amini, Daniel Faria, Irini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Clement Jonquet, Naouel Karam, Abderrahmane Khiat, Amir Laadhar, Patrick Lambrix, Huanyu Li, Ying Li, Pascal Hitzler, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Elodie Thiéblin, Cássia Trojahn, Jana Vataščinová, Beyza Yaman, Ondrej Zamazal, and Lu Zhou. Results of the Ontology Alignment Evaluation Initiative 2020. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn, editors, *OM 2020Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020) Virtual conference (originally planned to be in Athens, Greece), November 2, 2020*, volume 2788, pages 92–138, Aachen, 2020. RWTH. URL `https://madoc.bib.uni-mannheim.de/58061/`.

[219] Bruno R. Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in Java*. John Wiley & Sons, Inc., USA, 1999. URL `https://eduinformer.com/data-structures-algorithms-object-oriented-design-patterns-java/`.

[220] Anna Primpeli, Alexander Brinkmann, and Chris Bizer. Web Data Commons - RDFa, Microdata, Embedded JSON-LD, and Microformats Data Sets - October 2021. Website, December 2021. URL `http://webdatacommons.org/structureddata/2021-12/stats/stats.html`. Last time accessed, May 11th 2022.

[221] Martin Przyjaciel-Zablocki, Alexander Schätzle, Thomas Hornung, and Io Taxidou. Towards a sparql 1.1 feature benchmark on real-world social network data. In *Proceedings of the First International Workshop on Benchmarking RDF Systems*, 2013. URL `http://ceur-ws.org/Vol-981/BeRSys2013paper1.pdf`.

[222] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155. Association for Computational Linguistics, 2009. doi: 10.3115/1596374.1596399.

[223] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New*

Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[224] Michael Röder, Maximilian Speicher, and Ricardo Usbeck. Investigating Quality Raters' Performance Using Interface Evaluation Methods. In Matthias Horbach, editor, *Informatik 2013, 43. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Informatik angepasst an Mensch, Organisation und Umwelt, 16.-20. September 2013, Koblenz*, pages 137–139. GI, 2013.

[225] Michael Röder, Andreas Both, and Alexander Hinneburg. Evaluation des Konfigurationsraumes von Kohärenzmaßen fur Themenmodelle. In *Proceedings of the 16th LWA Workshops: KDML, IR and FGWM, Aachen, Germany, September 8-10*, 2014. URL http://svn.aksw.org/papers/2014/KDML_TopicEvaluation/public.pdf.

[226] Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N^3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3529–3533, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/856_Paper.pdf.

[227] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450333177. doi: 10.1145/2684822.2685324. URL https://doi.org/10.1145/2684822.2685324.

[228] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Developing a Sustainable Platform for Entity Annotation Benchmarks. In Fabien Gandon, Christophe Guéret, Serena Villata, John Breslin, Catherine Faron-Zucker, and Antoine Zimmermann, editors, *The Semantic Web: ESWC 2015 Satellite Events*, pages 190–196, Cham, 2015. Springer International Publishing. ISBN 978-3-319-25639-9. doi: 10.1007/978-3-319-25639-9_36. URL http://svn.aksw.org/papers/2015/ESWC_GERBIL_semdev/public.pdf.

[229] Michael Röder, Ricardo Usbeck, René Speck, and Axel-Cyrille Ngonga Ngomo. CETUS – A Baseline Approach to Type Extraction. In Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann, editors, *Semantic Web*

Evaluation Challenges, pages 16–27, Cham, 2015. Springer International Publishing. ISBN 978-3-319-25518-7. doi: 10.1007/978-3-319-25518-7_2. URL https://link.springer.com/chapter/10.1007/978-3-319-25518-7_2.

[230] Michael Röder, Axel-Cyrille Ngonga Ngomo, Ivan Ermilov, and Andreas Both. Detecting Similar Linked Datasets Using Topic Modelling. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *The Semantic Web. Latest Advances and New Domains. ESWC 2016*, pages 3–19, Cham, 2016. Springer International Publishing. ISBN 978-3-319-34129-3. doi: 10.1007/978-3-319-34129-3_1. URL http://svn.aksw.org/papers/2016/ESWC_Tapioca/public.pdf.

[231] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. GERBIL – Benchmarking Named Entity Recognition and Linking Consistently. *Semantic Web*, 9(5):605–625, 2018. doi: 10.3233/SW-170286. URL http://semantic-web-journal.org/system/files/swj1671.pdf.

[232] Michael Röder, Philip Frerk, Felix Conrads, and Axel-Cyrille Ngonga Ngomo. Applying Grammar-Based Compression to RDF. In Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Oscar Corcho, Petar Ristoski, and Mehwish Alam, editors, *The Semantic Web. ESWC 2021*, pages 93–108, Cham, 2021. Springer International Publishing. ISBN 978-3-030-77385-4. doi: 10.1007/978-3-030-77385-4_6. URL https://link.springer.com/chapter/10.1007/978-3-030-77385-4_6.

[233] Frank Rosner, Alexander Hinneburg, Michael Röder, Martin Nettling, and Andreas Both. Evaluating topic coherence measures. *CoRR*, abs/1403.6397, 2014. URL http://arxiv.org/abs/1403.6397. Accepted at the NIPS 2013 Workshop Topic Models: Computation, Application, and Evaluation, https://sites.google.com/site/nips2013topicmodels/papers.

[234] Tony Russell-Rose, Jon Chamberlain, and Leif Azzopardi. Information Retrieval in the Workplace: A Comparison of Professional Search Practices. *Information Processing & Management*, 54(6):1042–1057, 2018. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2018.07.003. URL https://www.sciencedirect.com/science/article/pii/S0306457318300220.

[235] Michael Röder, Tzanina Saveta, Irini Fundulaki, and Axel-Cyrille Ngonga Ngomo. HOBBIT Link Discovery Benchmarks at Ontology Matching 2017. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *Proceedings of the 12th International Workshop on Ontology Matching (OM-2017), Vienna, Austria, October 21, 2017.*,

pages 209–210. CEUR-WS, 2017. URL http://ceur-ws.org/Vol-2032/om2017_poster2.pdf.

[236] Michael Röder, Geraldo de Souza Jr, and Axel-Cyrille Ngonga Ngomo. Squirrel – Crawling RDF Knowledge Graphs on the Web. In *The Semantic Web – ISWC 2020*. Springer International Publishing, 2020. URL https://papers.dice-research.org/2020/ISWC_Squirrel/public.pdf.

[237] Michael Röder, Denis Kuchelev, and Axel-Cyrille Ngonga Ngomo. HOBBIT: A platform for benchmarking Big Linked Data. *Data Science*, 3(1):15–35, 2020. doi: 10.3233/DS-190021. URL https://content.iospress.com/articles/data-science/ds190021.

[238] Michael Röder, Mohamed Ahmed Sherif, Muhammad Saleem, Felix Conrads, and Axel-Cyrille Ngonga Ngomo. *Benchmarking the Lifecycle of Knowledge Graphs*, pages 73–97. IOS Press, 2020. doi: 10.3233/SSW200012.

[239] Michael Röder, Geraldo de Souza Jr., Denis Kuchelev, Abdelmoneim Amer Desouki, and Axel-Cyrille Ngonga Ngomo. ORCA – a Benchmark for Data Web Crawlers. In *Proceedings of the 15th IEEE International Conference on Semantic Computing (ICSC)*, pages 62–69. IEEE Computer Society, 2021. doi: 10.1109/ICSC50631.2021.00054. URL https://papers.dice-research.org/2021/ICSC2021_ORCA/ORCA_public.pdf.

[240] Michael Röder, Pham Thuy Sy Nguyen, Felix Conrads, Ana Alexandra Morim da Silva, and Axel-Cyrille Ngonga Ngomo. LEMMING – Example-based Mimicking of Knowledge Graphs. In *Proceedings of the 15th IEEE International Conference on Semantic Computing (ICSC)*, pages 62–69. IEEE Computer Society, 2021. doi: 10.1109/ICSC50631.2021.00015. URL https://papers.dice-research.org/2021/ICSC2021_Lemming/Lemming_public.pdf.

[241] Peter Saint-Andre and John C. Klensin. Uniform Resource Names (URNs). Technical Report 8141, Internet Engineering Task Force (IETF), April 2017. URL https://rfc-editor.org/rfc/rfc8141.txt.

[242] Muhammad Saleem, Intizar Ali, Aidan Hogan, Qaiser Mehmood, and Axel-Cyrille Ngonga Ngomo. LSQ: The Linked SPARQL Queries Dataset. In *International Semantic Web Conference (ISWC)*, 2015. doi: 10.1007/978-3-319-25010-6_15.

[243] Muhammad Saleem, Qaiser Mehmood, and Axel-Cyrille Ngonga Ngomo. FEASIBLE: A Feature-Based SPARQL Benchmark Generation Framework. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 52–69,

2015. doi: 10.1007/978-3-319-25007-6_4. URL http://svn.aksw.org/papers/2015/ISWC_FEASIBLE/public.pdf.

[244] Muhammad Saleem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. SPARQL Querying Benchmarks. In *Tutorial at ISWC*, 2016. URL http://svn.aksw.org/papers/2016/ISWC_SQBenchmarks/public.pdf.

[245] Arun V. Sathanur, Sutanay Choudhury, Cliff Joslyn, and Sumit Purohit. When Labels Fall Short: Property Graph Simulation via Blending of Network Structure and Vertex attributes. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2287–2290, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133065. URL https://doi.org/10.1145/3132847.3133065.

[246] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 245–260. Springer International Publishing, 2014. ISBN 978-3-319-11964-9.

[247] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. SP2Bench: A SPARQL performance benchmark. In *International Conference on Data Engineering (ICDE)*, pages 222–233. IEEE, 2009. doi: 10.1007/978-3-642-04329-1_16.

[248] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. SP^{2}Bench: A SPARQL Performance Benchmark. In *2009 IEEE 25th International Conference on Data Engineering*, pages 222–233, 2009. doi: 10.1109/ICDE.2009.28. URL https://doi.org/10.1109/ICDE.2009.28.

[249] Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, and Thanh Tran. FedBench: A Benchmark Suite for Federated Semantic Data Query Processing. In *International Semantic Web Conference (ISWC)*, pages 585–600. Springer, 2011. doi: 10.1007/978-3-642-25073-6_37.

[250] Guus Schreiber and Yves Raimond. RDF 1.1 Primer. W3C Note, W3C, June 2014. URL http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/.

[251] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006. doi: 10.1109/MIS.2006.62. URL https://doi.org/10.1109/MIS.2006.62.

[252] Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. WOMBAT - A Generalization Approach for Automatic Link Discovery. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *14th Extended Semantic Web Conference, Portorož, Slovenia, 28th May - 1st June 2017*, pages 103–119. Springer International Publishing, 2017. doi: https://doi.org/10.1007/978-3-319-58068-5_7. URL `http://svn.aksw.org/papers/2017/ESWC_WOMBAT/public.pdf`.

[253] Robert Shirey. Internet Security Glossary, Version 2. Technical Report 4949, Internet Engineering Task Force (IETF), August 2007. URL `https://rfc-editor.org/rfc/rfc4949.txt`.

[254] Tomoji Shogenji. Is coherence truth conducive? *Analysis*, 59(264):338–345, 1999. ISSN 1467-8284. doi: 10.1111/1467-8284.00191. URL `http://dx.doi.org/10.1111/1467-8284.00191`.

[255] Ayush Singhal, Ravindra Kasturi, and Jaideep Srivastava. Datagopher: Context-based search for research datasets. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pages 749–756, 2014. doi: 10.1109/IRI.2014.7051964.

[256] Jennifer Sleeman, Tim Finin, and Anupam Joshi. Topic modeling for rdf graphs. In *3rd International Workshop on Linked Data for Information Extraction, 14th International Semantic Web Conference*, 2015.

[257] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer International Publishing, 2014. doi: 10.1007/978-3-319-11964-9_33.

[258] René Speck, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. Open Knowledge Extraction Challenge 2017. In *Semantic Web Challenges: Fourth SemWebEval Challenge at ESWC 2017*, Communications in Computer and Information Science. Springer International Publishing, 2017. doi: 10.1007/978-3-319-69146-6_4.

[259] René Speck, Michael Röder, Felix Conrads, Hyndavi Rebba, Catherine Camilla Romiyo, Gurudevi Salakki, Rutuja Suryawanshi, Danish Ahmed, Nikit Srivastava, Mohit Mahajan, and Axel-Cyrille Ngonga Ngomo. Open Knowledge Extraction Challenge 2018. In Davide Buscaldi, Aldo Gangemi, and Diego Reforgiato Recupero, editors, *Semantic Web Challenges*, pages 39–51, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00072-1. doi: 10.1007/978-3-030-00072-1_4.

[260] René Speck and Axel-Cyrille Ngonga Ngomo. Leopard — a baseline approach to attribute prediction and validation for knowledge graph population. *Journal of Web Semantics*, 55:102–107, 2019. ISSN 1570-8268. doi: 10.1016/j.websem.2018.12.006.

[261] Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, and Niklas Lindström. JSON-LD 1.0 – A JSON-based Serialization for Linked Data. W3C Recommendation, W3C, January 2014. URL `http://www.w3.org/TR/2014/REC-json-ld-20140116/`.

[262] Padmini Srinivasan, Filippo Menczer, and Gautam Pant. A general evaluation framework for topical crawlers. *Information Retrieval*, 8(3):417–447, 2005.

[263] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012. doi: 10.3233/SW-2011-0052. URL `http://jens-lehmann.org/files/2012/linkedgeodata2.pdf`.

[264] Stergios Stergiou, Zygimantas Straznickas, Rolina Wu, and Kostas Tsioutsiouliklis. Distributed Negative Sampling for Word Embeddings. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2569–2575. AAAI Press, 2017. URL `https://dl.acm.org/doi/abs/10.5555/3298483.3298609`.

[265] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 952–961, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390948.2391052`.

[266] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[267] Patrick Stickler. CBD - Concise Bounded Description. W3C Member Submission, W3C, June 2005. URL `http://www.w3.org/Submission/2005/SUBM-CBD-20050603/`.

[268] Beth M. Sundheim. Tipster/MUC-5: Information extraction system evaluation. In *Proceedings of the 5th Conference on Message Understanding*, 1993. doi: 10.3115/1072017.1072023.

[269] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. FactCheck: Validating RDF Triples Using Textual Evidence. In *Proceedings of*

the 27th ACM International Conference on Information and Knowledge Management, International Conference on Information and Knowledge Management, page 1599–1602, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3269308. URL https://svn.aksw.org/papers/2018/CIKM_FACTCHECK/public.pdf.

[270] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Unsupervised discovery of corroborative paths for fact validation. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web – ISWC 2019*, pages 630–646, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30793-6. doi: 10.1007/978-3-030-30793-6_36. URL https://papers.dice-research.org/2019/ISWC2019_COPAAL/public.pdf.

[271] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. COPAAL – An Interface for Explaining Facts using Corroborative Paths. In Mari Carmen Suárez-Figueroa, Gong Cheng, Anna Lisa Gentile, Christophe Guéret, Maria Keet, and Abraham Bernstein, editors, *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas)*, volume 2456, pages 201–204. Springer International Publishing, 2019. URL https://papers.dice-research.org/2019/ISWC2019_COPAAL_Demo/public.pdf.

[272] Ruben Taelman, Pieter Colpaert, Erik Mannens, and Ruben Verborgh. Generating Public Transport Data based on Population Distributions for RDF Benchmarking. *Semantic Web Journal*, 10(2):305–328, Jan 2019. doi: 10.3233/SW-180319. URL http://rubensworks.net/raw/publications/2018/podigg.pdf.

[273] Tatiana Tarasova and Maarten Marx. Parlbench: a sparql benchmark for electronic publishing applications. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 5–21. Springer, 2013. doi: 10.1007/978-3-642-41242-4_2.

[274] Yannis Theoharis, Yannis Tzitzikas, Dimitris Kotzinos, and Vassilis Christophides. On Graph Features of Semantic Web Schemas. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):692–702, 2008. doi: 10.1109/TKDE.2007.190735. URL https://doi.org/10.1109/TKDE.2007.190735.

[275] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, 2003. doi: 10.3115/1119176.1119195.

[276] George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*. Citeseer, 2012.

[277] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8 (4):355 – 364, 2010.

[278] Yannis Tzitzikas, Nikos Manolis, and Panagiotis Papadakos. Faceted Exploration of RDF/S Datasets: A Survey. *Journal of Intelligent Information Systems*, 48(2):329–364, April 2017. ISSN 0925-9902. doi: 10.1007/s10844-016-0413-8. URL https://doi.org/10.1007/s10844-016-0413-8.

[279] Christina Unger, Philipp Cimiano, Vanessa López, and Enrico Motta, editors. *Proceedings of the 1st Workshop on Question Answering Over Linked Data (QALD-1)*, 2011. URL http://www.w3.org/TR/2011/NOTE-void-20110303/.

[280] Christina Unger, Philipp Cimiano, Vanessa López, Enrico Motta, Paul Buitelaar, and Richard Cyganiak, editors. *Proceedings of the Workshop on Interacting with Linked Data, Heraklion, Greece, May 28, 2012*, volume 913 of *CEUR Workshop Proceedings*, 2012. CEUR-WS.org. URL http://ceur-ws.org/Vol-913.

[281] Christina Unger, Corina Forascu, Vanessa López, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. Question answering over linked data (QALD-4). In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*, pages 1172–1180. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1180/CLEF2014wn-QA-UngerEt2014.pdf.

[282] Christina Unger, Corina Forascu, Vanessa López, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. Question answering over linked data (QALD-5). In Linda Cappellato, Nicola Ferro, Gareth J. F. Jones, and Eric SanJuan, editors, *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*, volume 1391 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL http://ceur-ws.org/Vol-1391/173-CR.pdf.

[283] Christina Unger, Axel-Cyrille Ngonga Ngomo, and Elena Cabrio. 6th open challenge on question answering over linked data (QALD-6). In Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange, editors, *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 641 of *Communications in Computer and Information Science*, pages 171–177. Springer, 2016. doi: 10.1007/978-3-319-46565-4_13. URL https://doi.org/10.1007/978-3-319-46565-4_13.

[284] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In *Proceedings of the Twenty-First European Conference on Artificial Intelligence*, ECAI'14, page 1113–1114, NLD, 2014. IOS Press. ISBN 9781614994183.

[285] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 457–471, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11964-9. doi: 10.1007/978-3-319-11964-9_29. URL https://link.springer.com/chapter/10.1007/978-3-319-11964-9_29.

[286] Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngomo Ngonga. Evaluating Entity Annotators Using GERBIL. In Fabien Gandon, Christophe Guéret, Serena Villata, John Breslin, Catherine Faron-Zucker, and Antoine Zimmermann, editors, *The Semantic Web: ESWC 2015 Satellite Events*, pages 159–164, Cham, 2015. Springer International Publishing. ISBN 978-3-319-25639-9. doi: 10.1007/978-3-319-25639-9_31. URL http://svn.aksw.org/papers/2015/ESWC_GERBIL_demo/public.pdf.

[287] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL – General Entity Annotation Benchmark Framework. In *24th WWW conference*, 2015.

[288] Ricardo Usbeck, Michael Röder, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Requirements to Modern Semantic Search Engine. In Axel-Cyrille Ngonga Ngomo and Petr Křemen, editors, *Knowledge Engineering and Semantic Web*, pages 328–343, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45880-9. doi: 10.1007/978-3-319-45880-9_25. URL https://link.springer.com/chapter/10.1007/978-3-319-45880-9_25.

[289] Ricardo Usbeck, Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. DIESEL – Distributed Search over Large Enterprise Data. In *ESWC, EU networking session*, 2016. URL http://svn.aksw.org/papers/2016/ESWC_2016_EU_DIESEL/public.pdf.

[290] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 7th open challenge on question answering over linked data (QALD-7). In Mauro Dragoni, Monika Solanki, and Eva Blomqvist, editors, *Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, volume 769 of *Communications in Computer and Information Science*, pages 59–69. Springer, 2017. doi: 10.1007/978-3-319-69146-6_6. URL https://doi.org/10.1007/978-3-319-69146-6_6.

[291] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Felix Conrads, Michael Röder, and Giulio Napolitano. 8th Challenge on Question Answering over Linked Data (QALD-8) (invited paper). In Key-Sun Choi, Luis Espinosa Anke, Thierry Declerck, Dagmar Gromann, Jin-Dong Kim, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, and Ricardo Usbeck, editors, *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018*, volume 2241 of *CEUR Workshop Proceedings*, pages 51–57. CEUR-WS.org, 2018. URL http://ceur-ws.org/Vol-2241/paper-05.pdf.

[292] Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrad, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger. Benchmarking Question Answering Systems. *Semantic Web*, 10 (2):293–304, January 2019. doi: 10.3233/SW-180312. URL http://www.semantic-web-journal.net/system/files/swj1578.pdf.

[293] Jóakim v. Kistowski, Jeremy A. Arnold, Karl Huppler, Klaus-Dieter Lange, John L. Henning, and Paul Cao. How to Build a Benchmark. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, ICPE '15, page 333–336, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450332484. doi: 10.1145/2668930.2688819. URL https://doi.org/10.1145/2668930.2688819.

[294] Patrick R.J. van der Laag and Shan-Hwei Nienhuys-Cheng. Completeness and properness of refinement operators in inductive logic programming. *The Journal of Logic Programming*, 34(3), 1998. ISSN 0743-1066. doi: https://doi.org/10.1016/S0743-1066(97)00077-0. URL https://www.sciencedirect.com/science/article/pii/S0743106697000770.

[295] Pierre-Yves Vandenbussche, Ghislain A. Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semantic Web*, 8(3):437–452, 2017. doi: 10.3233/SW-160213.

[296] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and Maintaining Links on the Web of Data. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2009*, pages 650–665, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04930-9. doi: 10.1007/978-3-642-04930-9_41. URL https://doi.org/10.1007/978-3-642-04930-9_41.

[297] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk – a Link Discovery Framework for the Web of Data. In *Proceedings of the 2nd Workshop on Linked Data on the Web (LDOW) in conjunction with the 18th International World Wide Web Conference (WWW)*, 2009. URL https://openreview.net/pdf?id=S1-aQfbuWr.

[298] Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

[299] Andreas Wagner, Peter Haase, Achim Rettinger, and Holger Lamm. Discovering Related Data Sources in Data-Portals. In Sarven Capadisli, Franck Cotton, Richard Cyganiak, Armin Haller, Alistair Hamilton, and Raphaël Troncy, editors, *Proceedings of the 1st International Workshop on Semantic Statistics (SemStats 2013) co-located with 12th International Semantic Web Conference*

(ISWC 2013), volume 1549 of *CEUR Workshop Proceedings*. CEUR-WS.org, October 2013. URL http://ceur-ws.org/Vol-1549/article-07.pdf.

[300] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking LDA: why priors matter. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1973–1981, 2009. URL http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.

[301] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998. doi: 10.1038/30918. URL https://doi.org/10.1038/30918.

[302] Matthias Wauer. Deliverable 1.1: Anforderungsanalyse. Project deliverable, OPAL – Open Data Portal, 2018. URL https://drive.google.com/open?id=1Ui45iCq63sHXNnc6xec14psmvXKLavDH. Last time accessed, December 15th, 2021, via http://projekt-opal.de/en/results/deliverables/.

[303] Matthias Wauer, Geraldo de Souza, Adrian Wilke, and Afshin Amini. Deliverable 2.1: Spezifikation der Crawler-Komponente. Project deliverable, OPAL – Open Data Portal, 2018. URL https://drive.google.com/open?id=1kIZGfMrnf1Bw2lvjRiBysD0yTv2C-QMG. Last time accessed, December 15th, 2021, via http://projekt-opal.de/en/results/deliverables/.

[304] Klaus Weltner. *Mathematik für Physiker 1*. Number 14 in Springer-Lehrbuch. Springer Berlin, Heidelberg, 2008. ISBN 978-3-540-74194-7. doi: 10.1007/978-3-540-74194-7. URL https://doi.org/10.1007/978-3-540-74194-7.

[305] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, Dec 1945. URL http://links.jstor.org/sici?sici=0099-4987%28194512%291%3A6%3C80%3AICBRM%3E2.0.CO%3B2-P.

[306] Mark D. Wilkinson, Michel Dumontier, Jan I. Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C. Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes,

Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3, 2016. doi: 10.1038/sdata.2016.18. URL https://doi.org/10.1038/sdata.2016.18.

[307] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. ACL '94, page 133–138, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL https://doi.org/10.3115/981732.981751.

[308] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 937–946, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584959. doi: 10.1145/1557019.1557121. URL https://doi.org/10.1145/1557019.1557121.

[309] Ying Zhang, Minh-Duc Pham, Oscar Corcho, and Jean-Paul Calbimonte. SRBench: A Streaming RDF/SPARQL Benchmark. In *International Semantic Web Conference (ISWC)*, volume 7649 of *Lecture Notes in Computer Science*, pages 641–657. Springer, 2012. doi: 10.1007/978-3-642-35176-1_40.

[310] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *33rd European Conference on IR Research, ECIR 2011*, pages 338–349, 2011. ISBN 978-3-642-20160-8. doi: 10.1007/978-3-642-20161-5_34.

[311] Xiaojuan Zhao, Yan Jia, Aiping Li, Rong Jiang, and Yichen Song. Multi-source knowledge fusion: a survey. *World Wide Web*, 23:2567–2592, July 2020. ISSN 1573-1413. doi: 10.1007/s11280-020-00811-0. URL https://doi.org/10.1007/s11280-020-00811-0.

[312] Matthäus Zloch, Maribel Acosta, Daniel Hienert, Stefan Dietze, and Stefan Conrad. A Software Framework and Datasets for the Analysis of Graph Measures on RDF Graphs. In Pascal Hitzler, Miriam Fernández, Krzysztof Janowicz, Amrapali Zaveri, Alasdair J.G. Gray, Vanessa Lopez, Armin Haller, and Karl Hammar, editors, *The Semantic Web*, pages 523–539, Cham, 2019. Springer International Publishing. ISBN 978-3-030-21348-0. doi: 10.1007/978-3-030-21348-0_34. URL https://doi.org/10.1007/978-3-030-21348-0_34.

List of Abbreviations

Notation	Description	Page List
20NG	A dataset based on the 20 Newsgroups corpus [10].	144–146, 150
A	Accessibility—one of the FAIR data principles.	42, 45, 47–49, 55, 78
API	Application Programming Interface.	38, 45, 52, 53, 78, 91–93, 96, 97, 103
ASCII	American Standard Code for Information Interchange.	10
AUC-ROC	area under receiver operating characteristic curve.	188, 192
BCS	Biased Class Selection.	64, 65, 73–76, 112, 284, 285
BIS	Biased Instance Selection.	65, 72–77, 112, 284, 285
CCS	Clustered Class Selection.	65, 73–76, 284, 285
CDF	Crawl delay fulfilment.	112
CKAN	Comprehensive Knowledge Archive Network.	91–93, 97, 98, 102, 103, 109–111, 113, 115, 116, 119
CPU	Central Processing Unit.	51, 83, 108, 111, 112
CR	Requirement for the data web crawler SQUIRREL.	87, 90, 92, 94
CSV	Comma Separated Value.	4, 196
D2KB	Disambiguation to knowledge base.	35
DCAT	Data Catalog Vocabulary.	93, 119

Notation	Description	Page List
DNS	Domain Name Service.	106
DOI	Digital Object Identifier.	2, 10
ELBO	Evidence Lower Bound.	20, 21, 272
F	Findable—one of the FAIR data principles.	42, 45, 47–49, 78
FAIR	FAIR data principles [306].	29, 31, 32, 40, 42, 43, 78, 82, 103, 116, 211
FIFO	first-in-first-out.	90
FN	false negative.	24, 25
FP	false positive.	24, 25
FS	An instance of LDSpider that uses a file sink.	110–114
FTP	File Transfer Protocol.	90
GTIN	Global Trade Item Number.	2
HTML	Hyper Text Markup Language.	2, 81, 83, 86, 91–93, 95–97, 103, 105, 115, 116
HTTP	Hypertext Transfer Protocol.	9, 10, 48, 83, 89, 90, 96, 116
HTTPS	Hyper Text Transfer Protocol Secure.	48
I	Interoperability—one of the FAIR data principles.	42, 45, 47, 78
ICC	International Chronostratigraphic Chart.	70, 71, 73, 74, 76, 77, 283, 285
IP	Internet Protocol.	90, 94

Notation	Description	Page List
IRI	Internationalized Resource Identifier.	9–14, 28, 34–39, 42, 45, 47, 49, 56, 58, 63, 66, 71, 78, 79, 84–86, 88–91, 93–96, 98, 100–103, 105–107, 131, 166, 173, 174, 176, 178–180, 188–190, 193–196, 263, 264, 271, 275, 277, 278
JSON-LD	JavaScript Object Notation for Linked Data.	4, 86, 91, 92, 96, 116, 119
KML	Keyhole Markup Language.	4
KPI	Key Performance Indicator.	40, 41, 45, 47–49, 53, 55, 107, 182, 183, 263, 269
LBS	Load-balancing strategy.	110, 112, 114
LDA	Latent Dirichlet Allocation.	15–18, 21, 22, 129, 161, 171, 180, 195
LGD	Linked Geo Data.	70, 71, 73, 74, 76, 283, 284
LOD	Linked Open Data.	4, 85, 182
MIME	Multipurpose Internet Mail Extensions.	91
NETL	Neural Embedding Topic Labelling.	130, 131
NPMI	Normalized Pointwise Mutual Information.	23, 125, 134, 140, 146, 147, 186, 187, 206

Notation	Description	Page List
NYT	A dataset based on a set of New York Times articles [10].	144–147, 150–153
OWL	Web Ontology Language.	6, 173, 178
PMI	Pointwise Mutual Information.	22, 23, 123, 125, 140
PROV-O	Provenance Ontology.	45, 94
QMpH	query mixes per hour.	70, 73, 76
QpS	queries per second.	70, 73, 76, 77
R	Reuse—one of the FAIR data principles.	42, 45, 47, 49, 51, 78
RAM	Random Access Memory.	51, 108, 111–114
RDF	Resource Description Framework.	2, 4–6, 8, 9, 11–13, 15, 28, 30, 31, 33, 34, 45, 56, 57, 60, 66, 71, 72, 75, 77, 80, 82–86, 91–93, 95, 96, 98, 100–108, 110, 111, 113, 115–122, 127–129, 131, 132, 153–157, 162, 164–166, 168, 171, 172, 174–178, 182–184, 187, 188, 193, 194, 209, 263, 268, 269, 277, 278, 291
RDFa	Resource Description Framework in Attributes.	4, 91, 92, 96–98, 102, 103, 105, 110, 113, 119
RDFS	Resource Description Framework Schema.	178

Notation	Description	Page List
RDR	Requested disallowed resources.	112
RFC	Request for Commons.	10, 89
RG	Research gap. The 4 research gaps that are tackled within this thesis are defined in Chapter 1.	7, 8, 171, 211, 212
RMSE	root mean squared error.	70, 73, 76, 77
RSS	Really Simple Syndication.	84
RTL-NYT	A dataset based on a set of New York Times articles [63].	144–146, 150
RTL-Wiki	A dataset based on a set of Wikipedia articles [63].	144–146, 150
SKOS	Simple Knowledge Organization System.	178
SPARQL	SPARQL Protocol And RDF Query Language.	4, 9, 15, 30, 33, 44, 50, 52, 56, 67, 70, 81, 84, 86, 91–93, 95, 96, 98, 102, 103, 105, 110, 111, 113, 115, 116, 121, 174
SRC	Spearman rank correlation.	70, 76
SWDF	Semantic Web Dog Food.	70, 71, 73–76
TN	true negative.	24, 25
TP	true positive.	24, 25
U	User requirement for the HOBBIT platform.	40–45, 47–49, 51, 53, 54
UCS	Uniform Class Selection.	64, 65, 73–77, 284, 285
UIS	Uniform Instance Selection.	65, 73–76, 284, 285
URI	Uniform Resource Identifier.	9, 10, 38, 89
URL	Uniform Resource Locator.	4, 49, 84, 95, 103, 110
URN	Uniform Resource Name.	10
VoID	Vocabulary of Interlinked Datasets.	86, 96, 178

Notation	Description	Page List
W3C	World Wide Web Consortium.	11, 12, 15
XHTML	Extensible Hypertext Markup Language.	84, 91, 97
XML	Extensible Markup Language.	4, 10, 11
XSD	XML Schema Datatypes.	45

List of RDF Namespaces

Notation	Description	Page List
cg	http://data.oregon.gov/resource/i3bn-rwu4/.	176, 177, 179
dbo	http://dbpedia.org/ontology/.	13–15, 185, 186, 188–190, 201–207
dbr	http://dbpedia.org/resource/.	13, 14, 36, 37, 185, 186, 188–190, 205, 206
dcat	http://www.w3.org/ns/dcat#.	95
ex	http://example.org/.	36, 37
ho	http://w3id.org/hobbit/vocab#.	45, 46, 48
niw	http://aksw.org/notInWiki.	36
owl	http://www.w3.org/2002/07/owl#.	38, 131, 173, 188, 267
prov	http://www.w3.org/ns/prov#.	94
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#.	12, 13, 46, 66, 131, 164, 178, 201
rdfs	http://www.w3.org/2000/01/rdf-schema#.	13, 14, 46, 48, 131, 188–190, 201

Notation	Description	Page List
sc	http://data.oregon.gov/resource/qhct-wumz/.	176, 177, 179
sq	http://w3id.org/dice-research/squirrel/vocab#.	94
wd	http://www.wikidata.org/entity/.	188–190
wdt	http://www.wikidata.org/prop/direct/.	188–190
wiki	http://en.wikipedia.org/wiki/.	11, 37
xsd	http://www.w3.org/2001/XMLSchema#.	13, 14, 46, 48, 94

List of Symbols

Notation	Description	Page List
$*$	A symbol that works like a wildcard, i.e., it is used for parameters or suffixes to express that any valid value could be inserted.	19, 64, 136, 137, 139, 143, 180, 278, 287
$\cdot$	The symbol used for the dot product operation.	181, 182
$\circledast$	An arbitrary binary operator of $\mathscr{O}$.	60, 281, 282
$\boxplus$	An operator that links and fuses two given RDF datasets.	191, 192
$\oplus$	An arbitrary binary operator of $\mathscr{O}$.	281, 282
†	This symbol marks a run of LEMMING that stopped early.	74
@N	A suffix for a metric that is measured for the N-th element in an ordered list.	xxiii, 192, 205, 207, 209
F	Placeholder for the file format in an IRI template.	102, 103
H	Placeholder for the host name in an IRI template.	102, 103
N	Placeholder for the resource ID in an IRI template.	102, 103
¶	The publication information for a chapter.	8, 27, 81, 117, 171
α	The hyper parameter that is used as prior for the Dirichlet distribution over topics.	16–21, 159, 181, 195
β	The hyper parameter that is used as prior for the Dirichlet distribution over words.	16–21, 195
χ	A value of the variational parameter X.	20, 21, 132
X	A variational parameter used to approximate Φ.	20, 21, 263
δ	The density of a graph.	58, 59, 63, 73
Δ	The difference of KPI values across several runs.	xxiii, 192, 205, 207, 209, 263
▲	The area under the curve that is created by the of the Δ method across several runs.	xxiii, 192, 206–209
ϵ	A small constant that is added to avoid the calculation of the logarithm of 0.	23, 124, 125, 140, 141

Notation	Description	Page List
ε	An error function returning the sum of the differences of the invariant expression values of the original graphs and a generated graph.	66, 73–76, 284, 285
η	The number of times a token with a particular word type has been assigned to a particular topic over all documents.	19, 20
γ	A value of the variational parameter Γ.	20, 21, 132
Γ	A variational parameter used to approximate Θ.	20, 21, 264
κ	Exponent proposed by Aletras et al. [10] to give higher values more weight.	125, 141–143, 145
λ	An (invariant) arithmetic expression.	60–63, 65, 66, 73, 74, 264, 281–283
$\lambda_\emptyset$	The empty expression.	60, 61, 63
Λ	A set of (invariant) arithmetic expressions.	56, 57, 63, 65, 66, 264
Λ_{max}	The set of the best performing invariant expressions.	65, 66
μ	The average value an expression returns for a set of graphs.	65, 66
ν	The number of nodes of a graph.	56, 57, 63, 70, 71, 98, 106, 264
ν_R	The number of nodes of a knowledge graph that are either blank nodes or IRI resources of that graph.	56, 57, 63, 70, 71
ω_o	A function that returns a set of class sets from Ω.	64
ω_s	A function that returns a set of class sets from Ω for a given property and a set of classes for the subject.	64
Ω	A set of constraints for a graph.	59, 64, 65, 264
φ	The result of a confirmation measure for a single subset pair S.	135, 142
ϕ	A topic's distribution over words.	16–20, 123, 180, 181, 265

Notation	Description	Page List
Φ	The set of word distributions (ϕ) of a topic model.	xxii, xxiii, 17–20, 22, 129, 159, 161, 181, 198, 200, 201, 263, 290, 291
π	a single path of a certain length between two single nodes in a knowledge graph.	184–186
Π	A set of paths in a knowledge graph. Depending on the number of additional arguments, it is either a set of paths between two given nodes or a set of corroborative paths for a given property.	184–187, 265
Π'	A set of paths between sets of instances of a given set of classes. If a restriction $\vec{q}$ is given in addition, the set contains only typed paths that fulfill the restriction.	184–187
ψ	The number of times a particular word type occurs in a certain document.	21, 132, 180, 193
Ψ	The digamma function, i.e., the first derivative of the logarithm of the gamma function [126].	21
ϱ	The number of topics in a topic model.	17–22, 129, 155, 156, 158–162, 181, 196–200, 204, 205, 207–209, 289, 290
ρ	A refinement operator.	56, 57, 60–63, 281
ς	A distribution over node degrees.	59, 65, 72
σ	The standard deviation an expression has for a set of graphs.	65, 66
τ	The number of triples.	100–102, 106, 265
τ_e	The number of external triples of a graph.	101
τ_i	The number of internal triples of a graph.	101, 102
θ	A document's distribution over topics.	16–20, 157, 158, 180–182, 266

Notation	Description	Page List
ϑ	The topic log odds for a document to measure the agreement between a topic model and human judgments.	xxii, 157, 158, 167
Θ	The set of topic distributions (θ) of a topic model.	17–20, 264
Υ	The set of node types.	98, 99, 266, 271
Υ_g	The set of node types to be generated.	98, 99
ξ	A value of the variational parameter Ξ.	20, 21, 132
Ξ	A variational parameter used to approximate Z.	20, 21, 266
ζ	The number of tokens that are assigned to a particular topic in a particular document.	19, 20, 156, 181
a	An aggregation function.	142, 143, 146, 148, 150, 152, 266, 287, 288
$\mathfrak{A}$	The global set of aggregation functions.	135, 146, 150, 287, 288
a_g	The geometric mean.	142, 148
a_h	The harmonic mean.	142, 148
a_x	The maximum as aggregation function.	142, 148
a_m	The median.	142, 148, 150
a_n	The minimum as aggregation function.	142, 148
a_q	The quadratic mean.	142, 148
a_a	The arithmetic mean.	142, 143, 146, 148, 150, 152, 287, 288
$\mathfrak{a}$	A difference function that calculates the difference of an expression value in comparison to the average and normalizes the result by the expression's standard deviation.	65, 66
A1	An example annotation system.	36, 37
A2	An example annotation system.	36, 37
$\mathbb{A}(\mathfrak{F})$	The space of all arithmetic expressions over a given finite set of real-valued functions.	60, 61
$\mathcal{A}$	A measure to find a good number of topics for an LDA model proposed by Arun et al. [19].	xxii, xxiii, 22, 159, 160, 162, 198–201, 290, 291

Notation	Description	Page List
b	A replacement function which replaces resources in triples of the target graph with resources of the source graph if these resources are connected with an `owl:sameAs` link.	189, 190
$\mathfrak{b}$	The ratio of the normalized number of tokens assigned to a topic in one corpus divided by the normalized number assigned to the same topic in another corpus.	156, 164
$\mathfrak{B}$	The global set of blank nodes as defined in Definition 2.4.	12, 13, 15, 58, 267
B	A set of blank nodes (a subset of $\mathfrak{B}$).	12, 58
BL_T	The topical aspect used by Kunze and Auer [155] as part of their RDF search engine.	193, 194, 196, 199, 201, 205–207
c	An RDF class.	14, 66
$\mathfrak{c}$	A mapping function that derives for a given node of an RDF graph the set of all classes the node belongs to.	13, 14, 58, 59, 63–66, 184, 271
cos	The cosine function.	125, 130, 142, 143, 146, 147, 150–152, 181, 268, 275, 287, 288
$\mathfrak{C}$	The global set of classes.	13, 14, 267
C	A set of classes (a subset of $\mathfrak{C}$).	13, 14, 58, 59, 64–66, 72, 73, 184, 185
$\mathcal{C}$	A topic coherence.	xxii, xxiii, 123–126, 135, 136, 142, 143, 146, 147, 149, 151–153, 155, 158–161, 169, 267, 268, 287–289
$\mathscr{C}$	The global set of topic coherences.	135
$\mathcal{C}_{\text{any-any}}$	The any-any coherence.	142, 288
$\mathcal{C}_{\text{cen}}$	The Centroid coherence.	125, 143, 288

Notation	Description	Page List
$\mathcal{C}_{\text{cos}}$	The Cosine coherence.	125, 143, 146, 147, 151, 288
$\mathcal{C}_{\text{Dice}}$	The Dice coherence.	143, 288
$\mathcal{C}_F$	Fitelson's coherence.	126, 143, 287
$\mathcal{C}_{\text{Jac}}$	The Jaccard coherence.	143, 288
$\mathcal{C}_{\text{NPMI}}$	The NPMI coherence.	125, 126, 142, 146, 147, 149, 151, 287
$\mathcal{C}_O$	Olsson's coherence.	126, 143, 287
$\mathcal{C}_{\text{one-all}}$	The one-all coherence.	142, 288
$\mathcal{C}_{\text{one-any}}$	The one-any coherence.	142, 146, 147, 151, 288
$\mathcal{C}_P$	The proposed coherence which relies only on a direct confirmation measure.	xxii, xxiii, 146, 147, 149, 151, 153, 155, 158–160, 169, 287, 289
$\mathcal{C}_S$	Shogenji's coherence.	126, 143, 287
$\mathcal{C}_{\text{UCI}}$	The UCI coherence.	123–126, 136, 142, 146, 147, 149, 151, 287
$\mathcal{C}_{\text{UMass}}$	The UMass coherence.	124, 142, 146, 147, 151, 287
$\mathcal{C}_{V2}$	The proposed vector-based coherence.	xxii, 146, 147, 149, 151–153, 155, 158–161, 169, 287
Dice	The Dice function.	142, 143, 268, 275, 288
d	A document.	15–22, 156, 157, 180, 181, 193
$\mathfrak{d}$	A function that retrieves the domain of a given property.	14, 184, 186, 187
d	The degree of an RDF resource in a graph.	59–61, 66, 100–102, 106, 268
$\bar{\mathrm{d}}$	The average degree of RDF resources.	100–102, 106
Dir	The Dirichlet distribution.	16, 17, 20

Notation	Description	Page List
D	A corpus (i.e., a set of documents).	xxii, xxiii, 15–22, 129, 156, 159, 161, 164, 193, 198, 200, 201, 269, 273, 280, 290, 291
D_{RDF}	A corpus generated from a given set of RDF datasets.	156, 164
D_{Wiki}	A corpus generated from the English Wikipedia.	156, 164
e	An edge of a graph.	95, 100
$\mathfrak{e}$	A function that maps a given element to a vector in the embedding space of the embedding model to which this function belongs to.	130, 269
$\mathfrak{e}^d_{d2v}$	A function that maps a given word phrase to a vector in the embedding space of a doc2vec embedding model.	130
$\mathfrak{e}^w_{d2v}$	A function that maps a given word to a vector in the embedding space of a doc2vec embedding model.	130
$\mathfrak{e}^w_{w2v}$	A function that maps a given word to a vector in the embedding space of a word2vec embedding model.	130
e	A KPI to measure the performance of f.	182, 183, 192
E	A set of edges.	95, 99, 100
$\mathbb{E}$	The expected value of a probability distribution.	20
f	A function that takes an RDF dataset as input.	182, 183, 192, 269
f	A frequency count.	131, 132, 178–180
$\mathfrak{f}$	A real-valued function.	60, 61, 63
F1	The F1-measure as defined in Equation 2.35.	24, 25, 182
F1_{mac}	The macro F1-measure as defined in Equation 2.41.	25
F1_{mic}	The micro F1-measure as defined in Equation 2.38.	25, 68
FN	The count of false negatives.	24, 25
FP	The count of false positives.	24, 25

Notation	Description	Page List
$\mathfrak{F}$	A set of real-valued functions.	xxv, 60, 61, 72, 266
$\mathfrak{s}$	A measure for the specificity of an expression if it is used as an invariant for a set of knowledge graphs.	63
$\mathcal{G}$	A knowledge graph.	12–14, 36, 56, 58, 59, 62–66, 73–76, 100–102, 176, 180, 181, 184, 186–192, 194, 203–206, 270, 277, 284, 285
$\mathbb{G}$	A directed graph.	95
$\mathcal{G}_{\text{ex}}$	An example knowledge graph shown in Figure 2.1.	13, 14, 36
$\dot{\mathcal{G}}$	A generated knowledge graph.	63–66, 73–76, 284, 285
$\mathfrak{G}$	A set of graph generators.	62, 72
$\tilde{\mathcal{G}}$	A negative example graph.	62
$\mathcal{G}_Q$	The query dataset for which linking candidates should be retrieved.	176, 180, 181, 187, 188, 191, 192, 203–206
$\mathscr{G}$	A set of knowledge graphs.	56–59, 62–66, 72, 73, 176, 178, 191, 192, 194, 270
$\tilde{\mathscr{G}}$	A set of negative example graphs.	62, 63, 72
$\mathcal{G}_{\mathbf{S}}$	The source knowledge graph in Link Discovery.	188–190
$\mathcal{G}_{\mathbf{T}}$	The target knowledge graph in Link Discovery.	188–190
h	A measure for the variance of an expression for a set of knowledge graphs.	62, 270
h'	An extension of h that takes the length of the given expression into account.	62, 63, 277
$\mathfrak{h}_1$	The distribution of singular values of the matrix M_1 as proposed by Arun et al. [19].	22
$\mathfrak{h}_2$	A vector based on the length vector of the documents and the matrix M_2 as proposed by Arun et al. [19].	22

Notation	Description	Page List
i	This letter is used to represent an index to enumerate elements of a set or similar.	15–17, 19–22, 58–60, 62, 64–66, 95, 98–100, 123–127, 132, 136, 137, 140–142, 152, 156–158, 164, 180, 181, 184–186, 188–190, 192–194, 277, 279, 280
$\mathfrak{i}$	A mapping function that derives for a given set of classes all nodes that are instances of all the given classes (the inverse of $\mathfrak{c}$).	14, 63, 65, 186
$\mathfrak{I}$	The global set of IRI resources as defined in Definition 2.2.	12, 13, 15, 58, 275
j	This letter is used to represent an index to enumerate elements of a set or similar.	16, 17, 19–21, 58, 59, 62, 64–66, 72, 95, 98, 100–102, 123–126, 136, 141, 156–158, 181, 184, 186–189, 192, 194, 279, 280
Jac	The Jaccard function.	140, 142, 143, 150, 268, 272, 275, 288
k	This letter is used to represent an index to enumerate elements of a set or similar.	19–21, 58, 65, 66, 95, 100, 102, 126, 156, 157, 181, 184–186, 190
$\mathfrak{k}$	An element of the connectivity matrix $\mathcal{K}$.	98
KL	The Kullback-Leibler divergence.	22
$\mathcal{K}$	The $\lvert\Upsilon\rvert \times \lvert\Upsilon\rvert$ connectivity matrix.	xxv, 98, 99, 271
l	A vector with the lengths of the single documents of a corpus.	22
ℓ	A distribution over literal values.	59, 66, 72
$\mathfrak{l}$	The length of a path.	184–187

Notation	Description	Page List
$\mathcal{L}$	The Likelihood of something. In this thesis, it is mainly used for the ELBO.	20
$\mathbb{L}$	A link set, i.e., a set of pairs of resources that are linked via a given relation.	188–190
$\mathfrak{L}$	The global set of literals as defined in Definition 2.3.	12, 15, 58, 272
L	A set of literals (a subset of $\mathfrak{L}$).	12, 15, 58, 59, 66, 274, 277
m	A confirmation measure.	126, 140–143, 146, 149, 150, 152, 272, 273, 287, 288
$\tilde{m}$	An indirect confirmation measure.	142, 143, 146, 150, 152, 287, 288
m_c	Conditional confirmation measure.	140, 150
m_d	Difference confirmation measure.	140, 142, 146, 149, 150, 288
m_f	Fitelson's confirmation measure.	126, 141, 143, 146, 149, 150, 287
m_{Jac}	Jaccard confirmation measure.	140, 150
$m_{\mathbb{P}}$	A simple confirmation measure that solely relies on the joint probability.	140, 150
m_l	Likelihood confirmation measure.	140, 141, 150
m_{lc}	Log-conditional confirmation measure.	140, 142, 146, 150, 287
$m_{l\text{Jac}}$	Log-jaccard confirmation measure.	140, 150
m_{ll}	Log-likelihood confirmation measure.	140, 141, 150
m_{lr}	Log-ratio confirmation measure.	140, 142, 146, 149, 150, 287
m_{ls}	Logarithmic variant of Shogenji's confirmation measure.	141, 143, 150, 287
m_{nlr}	Normalized log-ratio confirmation measure.	140, 142, 143, 146, 149, 150, 152, 287, 288

Notation	Description	Page List
m_o	Olsson's confirmation measure.	141, 143, 150, 287
m_r	Ratio confirmation measure.	140, 150
$\mathfrak{M}$	The global set of confirmation measures.	134, 135, 146, 150, 287, 288
M	The corpus matrix of order $\|D\| \times \|\mathbb{V}\|$.	22, 270
$\mathfrak{n}$	The number of datasets in an evaluation.	25
n	The number of top words of a topic.	123, 124, 126, 141, 142, 146, 152, 153
NPMI	The normalized pointwise mutual information measure as defined in Equation 2.31.	23, 186, 187
NPMI_ϵ	The normalized pointwise mutual information measure for probabilities based on counts as defined in Equation 2.32.	23, 125
N	The rank of an element within a ranking.	xxiii, 192, 205–207, 209, 263
o	The object of a triple as defined in Definition 2.6.	12, 14, 58, 59, 64, 65, 101, 184, 185, 187, 204, 264
O	This symbol marks the big O notation.	190, 192
$\mathscr{O}$	A set of binary arithmetic operations.	60–62, 263, 281
p	A property as defined in Definition 2.5.	12, 14, 15, 58, 59, 64–66, 72, 73, 101, 184–189, 204
PMI	The pointwise mutual information measure as defined in Equation 2.29.	23
PMI_ϵ	The pointwise mutual information measure for probabilities based on counts as defined in Equation 2.30.	23, 123, 124
Pr	The precision value as defined in Equation 2.33.	24, 25
Pr_{mac}	The macro precision value as defined in Equation 2.39.	25
Pr_{mic}	The micro precision value as defined in Equation 2.36.	25, 68

Notation	Description	Page List
$\mathbb{P}$	A probability.	xxii, xxiii, 17–20, 22, 23, 58, 59, 64, 65, 73, 124–126, 129, 140, 150, 159, 161, 181, 186, 198, 200, 201, 272, 290, 291
$\mathcal{P}$	The method of probability estimation that defines the way how probabilities are derived from the underlying data.	134, 135, 137, 139, 142, 146, 148, 150, 274, 287, 288
$\boldsymbol{\mathcal{P}}$	The space of all probability estimations.	134, 135, 146, 287, 288
$\mathcal{P}_{bd}$	The boolean document probability estimation.	137, 139, 142, 146, 148, 287, 288
$\mathcal{P}_{bp}$	The boolean paragraph probability estimation.	137, 139, 148
$\mathcal{P}_{bs}$	The boolean sentence probability estimation.	137, 139, 148
$\mathcal{P}_{cw}$	The context window probability estimation. The number attached to the subscript indicates the size of the window.	139, 146, 148, 288
$\mathcal{P}_{sw}$	The sliding window probability estimation. The number attached to the subscript indicates the size of the window.	137, 139, 142, 146, 148, 150, 287
$\mathbf{P}$	A set of probabilities.	135
$\mathfrak{P}$	The global set of properties as defined in Definition 2.6.	12, 274
P	A set of properties (a subset of $\mathfrak{P}$).	12, 15, 59, 66, 101, 102, 274
P_L	A set of datatype properties.	15, 59, 66
P_R	A set of object properties.	15
q	A distribution introduced by the Variational Bayesian approach.	20
$\mathfrak{q}$	A property which is used as a restriction for a set of paths.	xxix, 184–187, 205, 206, 265, 275

Notation	Description	Page List
$\vec{\mathfrak{q}}$	An ordered set of properties that are used to restrict a set of paths.	xxix, 184–187, 265
$\mathfrak{Q}$	A set of restrictions.	187
$\mathfrak{r}$	A function that retrieves the range of a given property.	14, 15, 184, 186, 187
r	An IRI resource.	101, 102
r	A rounding function, which results the next integer value.	132, 180
Re	The recall value as defined in Equation 2.34.	24, 25
Re_{mac}	The macro recall value as defined in Equation 2.40.	25
Re_{mic}	The micro recall value as defined in Equation 2.37.	25, 68, 110, 112, 113
R_e	The set of external IRI resources, i.e., resources belonging to a different RDF graph.	100–102
R_i	The set of internal IRI resources.	100–102
R	A set of IRI resources (a subset of $\mathfrak{I}$).	12, 15, 56–59, 63, 70, 71, 100–102, 188, 190, 264, 274, 275, 277, 278
sim	A function that measures the similarity of two given things.	130, 142, 181, 275
sim_{cos}	The cosine similarity function that calculates the similarity of two given vectors based on the cosine of the angle between the two vectors.	130, 142, 181
sim_{Dice}	The Dice similarity function that calculates the similarity of two given vectors.	142
sim_{Jac}	The Jaccard similarity function that calculates the similarity of two given vectors.	142
sim_{lt}	A function to measure the similarity between a label candidate and the top words of a topic.	130
s	The subject of a triple as defined in Definition 2.6.	12, 14, 58, 59, 64, 65, 101, 184, 185, 187, 204, 264
$\mathfrak{S}$	A set of seed nodes.	xxvii, 95, 97, 99, 100

Notation	Description	Page List
S	A segmentation.	xxi, 126, 127, 135–138, 140–143, 146, 148–150, 152, 153, 264, 276, 287, 288
$\mathcal{S}$	The set of all segmentation schemes.	134, 135, 146, 150, 287, 288
S_{one}^{all}	The all-one segmentation.	137, 138, 143, 148, 287
S_{any}^{any}	The any-any segmentation.	xxi, 136–138, 142, 148, 150, 152, 153
S_{all}^{one}	The one-all segmentation.	xxi, 136–138, 142, 146, 148, 150, 152, 153, 287, 288
S_{any}^{one}	The one-any segmentation.	xxi, 126, 136–138, 142, 143, 146, 148, 152, 153, 287, 288
S_{one}^{one}	The one-one segmentation.	xxi, 136, 138, 142, 143, 146, 148, 149, 152, 287, 288
S_{pre}^{one}	The one-preceding segmentation.	xxi, 136, 138, 142, 146, 148, 149, 152, 153, 287, 288
S_{set}^{one}	The one-set segmentation.	xxi, 137, 138, 143, 148, 149, 152, 288
S_{suc}^{one}	The one-succeding segmentation.	xxi, 136, 138, 148, 149, 152, 153
S_{set}^{set}	The set-set segmentation.	137, 143, 148, 149, 152, 287

Notation	Description	Page List
$\mathscr{S}$	A set of RDF statements for which the veracity is known.	187, 204, 206, 277
$\mathscr{S}^-$	A set of RDF statements that are known to be false.	187
$\mathscr{S}^+$	A set of RDF statements that are known to be true.	187, 204
tf-idf	The "term frequency, inversed document frequency" function.	175, 193, 194, 196, 199, 201, 205–208
T	The set of triples.	12, 58, 59, 100, 187, 189, 277
T'_e	The set of generated external triples, i.e., triples that have an external resource as object.	101, 102
T'	The set of generated triples that only contain IRI resources as subjects and objects.	101
T'_i	The set of generated internal triples, i.e., triples that have an internal resource as object.	101, 102
T_L	The set of triples that have only literals as objects.	58, 59
T_R	The set of triples that have only blank nodes or IRI resources as objects.	58, 59
TN	The count of true negatives.	25
TP	The count of true positives.	24, 25
u	The weight of the length for the result of h'.	62, 63, 72
u	A node type.	98
$\mathfrak{u}$	The id of a topic that has been chosen by a user as intruder topic.	157, 158
u	A function that returns 1 if the given RDF vocabulary $\mathfrak{v}$ is used in the given dataset $\mathcal{G}$ or 0 otherwise.	194
$\mathfrak{U}$	The bag of user answers.	157, 158
$\vec{\mathrm{v}}$	A context vector of a subset of $\overline{W}$.	125, 141, 142, 277
$\vec{\mathrm{v}}_c$	The centroid vector of the context vectors of a subset of $\overline{W}$.	125
v	A single value of a context vector of a subset of $\overline{W}$.	125
$\vec{\mathrm{v}}'$	The context vector of W'.	141, 142, 277
v'_i	The i-th element of the context vector $\vec{\mathrm{v}}'$.	142
$\vec{\mathrm{v}}''$	The context vector of W''.	141, 142, 277
v''_i	The i-th element of the context vector $\vec{\mathrm{v}}''$.	142

Notation	Description	Page List
v	A node in a graph.	xxvii, 13, 14, 58, 64–66, 95, 99, 100, 184–186, 188–190, 278
v_ϵ	The entrance node of a graph.	95
$\mathfrak{v}$	An RDF vocabulary.	194, 277
$\mathbb{V}$	The vocabulary of a corpus.	15, 16, 19–22, 193, 273
V	A set of nodes.	12–14, 58, 62, 65, 66, 95, 97–101, 190, 278
V_m	The set of marked nodes in a graph.	xxvii, 99, 100
V_R	The set of nodes of a knowledge graph that comprises the blank nodes and IRI resources of that graph but not the its literals.	58
V_Q	A queue for graph nodes.	100
V_ν	The set of nodes of the synthetic Data Web.	97–99
V_u	The set of unmarked nodes in a graph.	99
$\mathfrak{V}$	The union of RDF vocabularies.	194
$\mathcal{V}_{*L}$	A variant of TAPIOCA's document generation that logarithmic on unique counts.	180
$\mathcal{V}_{*U}$	A variant of TAPIOCA's document generation that relies on unique counts.	180
$\mathcal{V}_A$	A variant of TAPIOCA's document generation that relies on class and property counts.	179
$\mathcal{V}_{AL}$	A variant of TAPIOCA's document generation that relies on logarithmic class and property counts.	xxii, 180, 195–197, 204–209
$\mathcal{V}_{AU}$	A variant of TAPIOCA's document generation that relies on unique class and property counts.	180, 195, 196
$\mathcal{V}_C$	A variant of TAPIOCA's document generation that relies on class counts.	179
$\mathcal{V}_{CL}$	A variant of TAPIOCA's document generation that relies on logarithmic class counts.	180, 195–197, 204–208
$\mathcal{V}_{CU}$	A variant of TAPIOCA's document generation that relies on unique class counts.	180, 195, 196
$\mathcal{V}_P$	A variant of TAPIOCA's document generation that relies on property counts.	179

Notation	Description	Page List
$\mathcal{V}_{PL}$	A variant of TAPIOCA's document generation that relies on logarithmic property counts.	xxii, 180, 195–200, 204–209
$\mathcal{V}_{PU}$	A variant of TAPIOCA's document generation that relies on unique property counts.	180, 195, 196
$\mathfrak{w}$	The weight of a node when sampling a node for an edge.	65
w	A word token, i.e., the single occurrence of a word type.	15–19, 181
w	A word type, i.e., a (natural language) word from a vocabulary.	15, 17, 19–21, 123–126, 130–132, 136, 137, 141, 180, 181, 193, 279
W_L	The set of words of a topic label candidate.	130, 131
$\mathbb{W}_i$	The set of all subsets of a given word set that do not contain the word type w_i.	126, 279
$\mathbb{W}_{i,j}$	The j-th element of $\mathbb{W}_i$, i.e., the j-th set of all subsets of a given word set that do not contain the word type w_i.	126
W	A set of words.	123–127, 130, 131, 135–137, 140–142, 153, 161, 163, 277, 279
W'	The set of words whose occurrence might be supported by W''.	127, 136, 137, 140–142, 277, 279
W''	The set of supporting words, i.e., words that are tested whether they support the occurrence of W'.	127, 136, 137, 140–142, 153, 277, 279
$\overline{W}$	The set of top words of a topic.	123–126, 130, 131, 135–137, 141, 142, 161, 163, 277, 279
$\overline{W}_{\mathrm{ex}}$	A set of top words of an example topic.	124–126
x	The id of the real intruder topic.	157, 158
x	A random value of $\mathcal{X}$.	22, 23

Notation	Description	Page List
$\mathcal{X}$	A random variable.	22, 23, 279
y	A random value of $\mathcal{Y}$.	22, 23
$\mathcal{Y}$	A random variable.	22, 23, 280
z	The topic assignment of a word, i.e., the index of a topic that created this word.	17–20, 181, 280
$\mathfrak{z}$	The veracity score that is calculated for a given triple.	187
Z	The set of all topic indexes of a corpus D.	17–22, 181, 266, 280
$\tilde{Z}_{i,j}$	The set of all topic assignments except the assignment $z_{i,j}$.	19, 20

Appendix A

A.1 Expression Transformation

This leads to unbalanced trees since the left subtree of a node might be a complex expression while the right node is always a single function. We name this a left deep tree. Hence, ρ is complete if it is possible that every binary expression tree over $\mathscr{O}$ can be represented as such a left deep tree.

As described in Section 3.5.2, our refinement operator ρ is complete since it is able to generate Taylor series.[1] These series can approximate other expressions [304]. However, a lot of expressions can be directly transformed into a left-deep tree. Let λ_1 and λ_2 be two expressions which are connected by a binary operation $\circledast$ which can be any of the binary operations in $\mathscr{O}$. Their binary expression tree is shown in Figure A.1a. If λ_2 is a single function the tree already has the form of a left-deep tree. If λ_2 comprises more than a single function, it can be represented by its two subexpressions $\lambda_{2,1}$ and $\lambda_{2,2}$ that are connected by the operator $\oplus \in \mathscr{O}$ as follows:

$$\lambda_2 = \lambda_{2,1} \oplus \lambda_{2,2} \,. \tag{A.1}$$

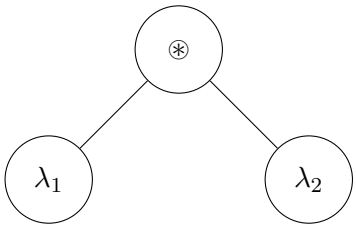
As shown in Figure A.1b, the binary expression tree of the two connected expressions λ_1 and λ_2 is not a left-deep tree:

$$\lambda_1 \circledast \lambda_2 = \lambda_1 \circledast (\lambda_{2,1} \oplus \lambda_{2,2}) \,. \tag{A.2}$$

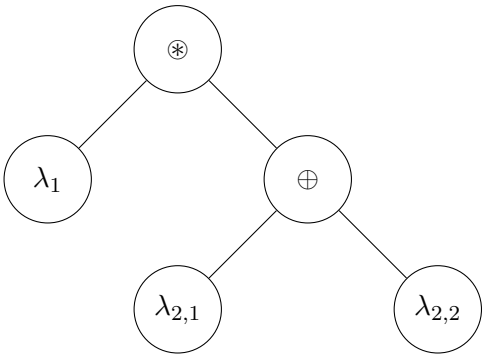
In a left-deep tree, the left operator has to be executed before the right operator. However, because of the parenthesis in the expression, this does not hold in our example.

Since $|\mathscr{O}| = 4$, there are 16 different combinations that the two operators $\circledast$ and $\oplus$ can have in the equation above. We demonstrate for 14 of the 16 combinations how

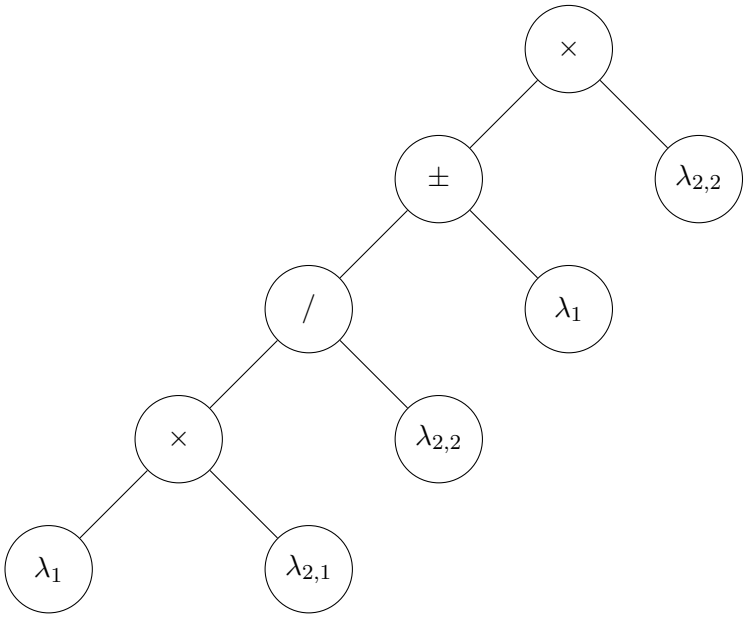
[1] Our definition of completeness covers global completeness. We do not claim that our refinement operator offers local completeness [195].



(a) Binary expression tree with two expressions.



(b) λ_2 represented by its subexpressions and its binary arithmetic operation.



(c) The left-deep tree for the case that $\circledast = \times$ and $\oplus \in \{+, -\}$ (expressed by $\pm$) as shown in equation A.13.

Figure A.1.: A graphical representation of binary expression trees.

the expression can be transformed to create a left-deep tree. In the following, we use $\pm$ in cases in which plus or minus could be used:

$$\lambda_1 + (\lambda_{2,1} + \lambda_{2,2}) = (\lambda_1 + \lambda_{2,1}) + \lambda_{2,2}\,, \tag{A.3}$$

$$\lambda_1 - (\lambda_{2,1} - \lambda_{2,2}) = (\lambda_1 - \lambda_{2,1}) + \lambda_{2,2}\,, \tag{A.4}$$

$$\lambda_1 \times (\lambda_{2,1} \times \lambda_{2,2}) = (\lambda_1 \times \lambda_{2,1}) \times \lambda_{2,2}\,, \tag{A.5}$$

$$\lambda_1 / (\lambda_{2,1} / \lambda_{2,2}) = (\lambda_1 / \lambda_{2,1}) \times \lambda_{2,2}\,, \tag{A.6}$$

$$\lambda_1 + (\lambda_{2,1} - \lambda_{2,2}) = (\lambda_1 + \lambda_{2,1}) - \lambda_{2,2}\,, \tag{A.7}$$

$$\lambda_1 - (\lambda_{2,1} + \lambda_{2,2}) = (\lambda_1 - \lambda_{2,1}) - \lambda_{2,2}\,, \tag{A.8}$$

$$\lambda_1 \times (\lambda_{2,1} / \lambda_{2,2}) = (\lambda_1 \times \lambda_{2,1}) / \lambda_{2,2}\,, \tag{A.9}$$

$$\lambda_1 / (\lambda_{2,1} \times \lambda_{2,2}) = (\lambda_1 / \lambda_{2,1}) / \lambda_{2,2}\,, \tag{A.10}$$

$$\begin{aligned} \lambda_1 \pm (\lambda_{2,1} \times \lambda_{2,2}) &= ((\lambda_1 \times \lambda_{2,2}) / \lambda_{2,2}) \pm (\lambda_{2,1} \times \lambda_{2,2}) \\ &= ((\lambda_1 / \lambda_{2,2}) \pm \lambda_{2,1}) \times \lambda_{2,2}\,, \end{aligned} \tag{A.11}$$

$$\begin{aligned} \lambda_1 \pm (\lambda_{2,1} / \lambda_{2,2}) &= ((\lambda_1 \times \lambda_{2,2}) / \lambda_{2,2}) \pm (\lambda_{2,1} / \lambda_{2,2}) \\ &= ((\lambda_1 \times \lambda_{2,2}) \pm \lambda_{2,1}) / \lambda_{2,2}\,, \end{aligned} \tag{A.12}$$

$$\begin{aligned} \lambda_1 \times (\lambda_{2,1} \pm \lambda_{2,2}) &= (\lambda_1 \times \lambda_{2,1}) \pm (\lambda_1 \times \lambda_{2,2}) \\ &= ((\lambda_1 \times \lambda_{2,1}) \times (\lambda_{2,2} / \lambda_{2,2})) \pm (\lambda_1 \times \lambda_{2,2}) \\ &= (((\lambda_1 \times \lambda_{2,1}) / \lambda_{2,2}) \pm \lambda_1) \times \lambda_{2,2})\,. \end{aligned} \tag{A.13}$$

A.2 Lemming Error Plots

Figures A.2 and A.3 show the error scores for the LGD and ICC dataset, respectively. The diagrams show the reduction of the error scores throughout the experiment described in Section 3.6.3. Each diagram comprises three curves for the three runs.

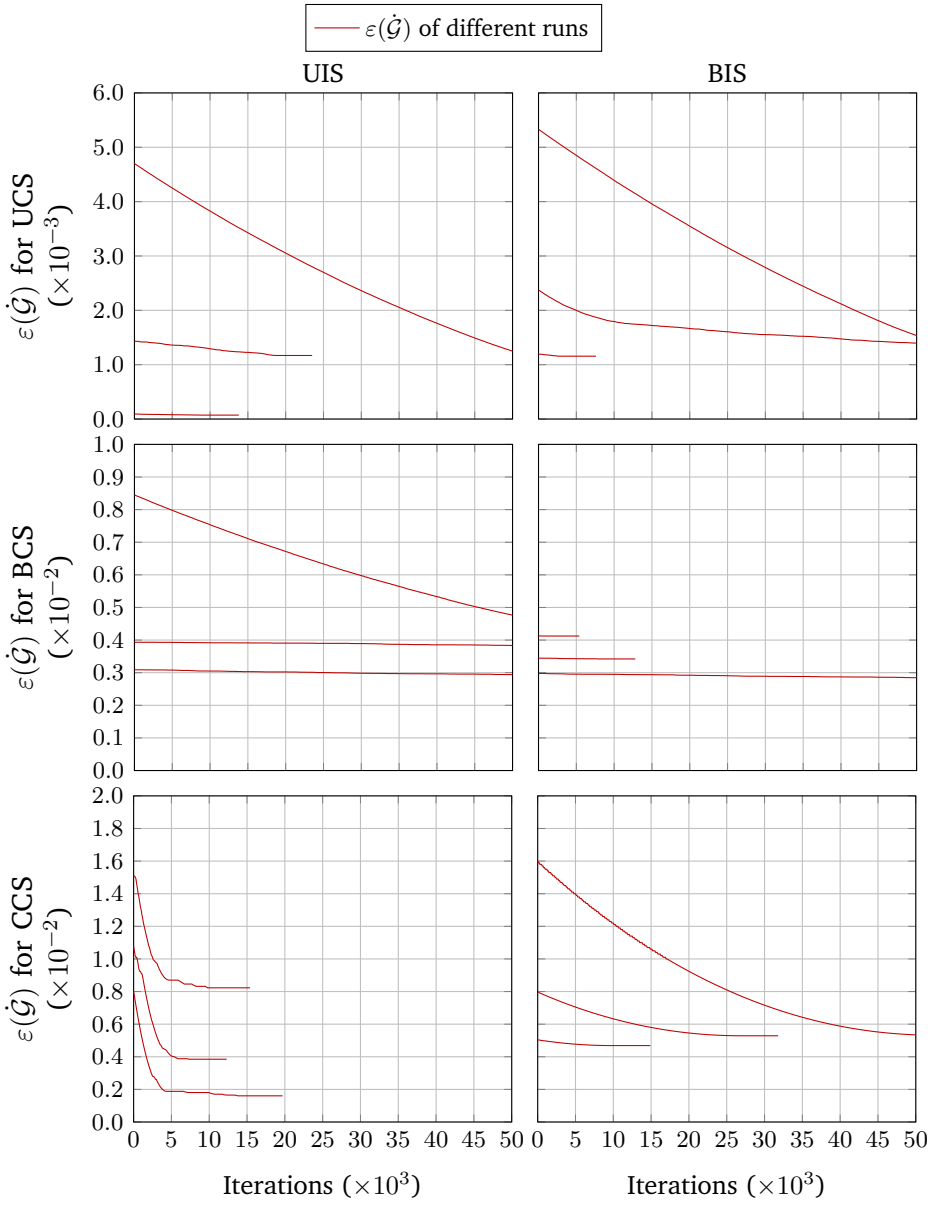


Figure A.2.: The course of error values during the amendment phase for the LGD dataset and all variants of LEMMING. The three rows are the class selection variants while the two columns show the instance selection variants.

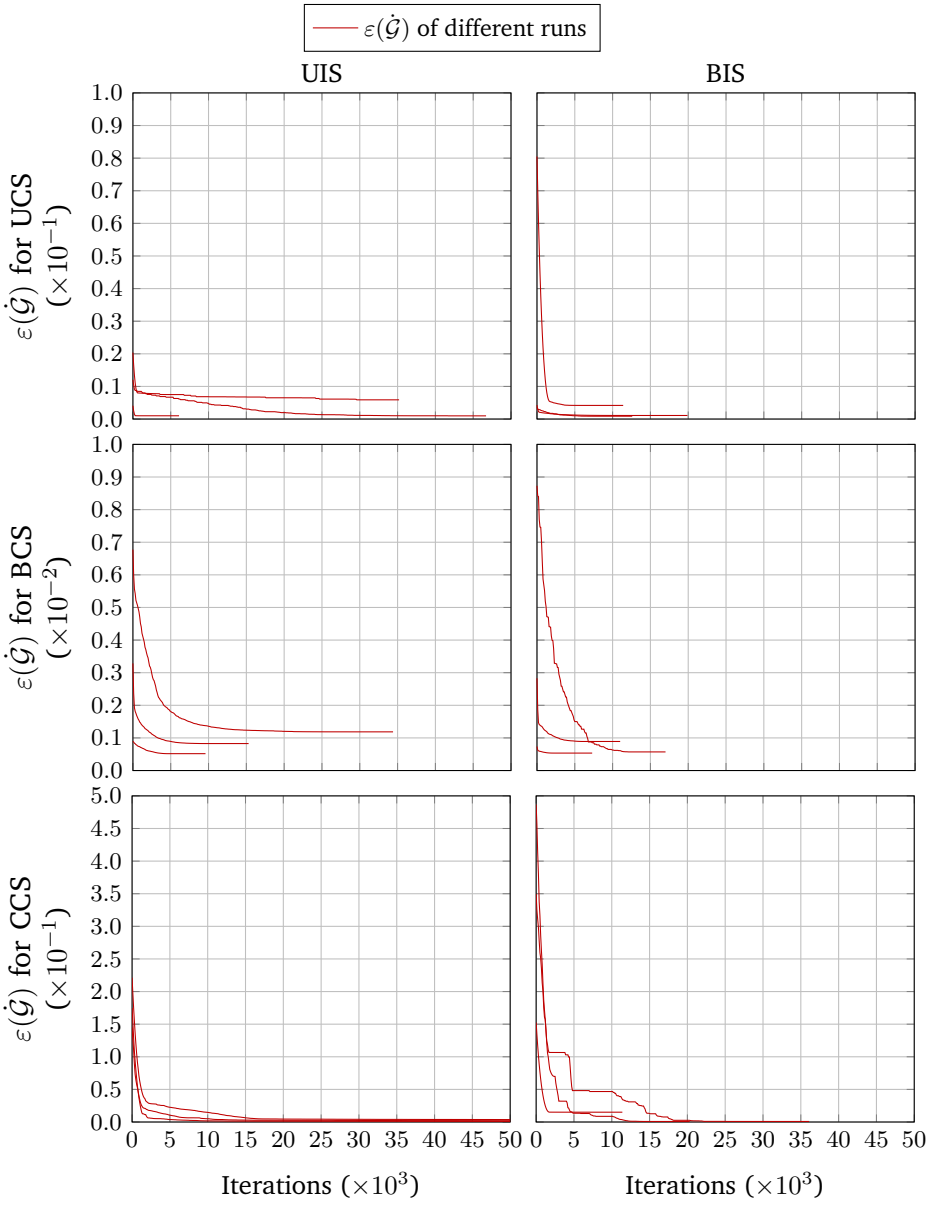


Figure A.3.: The course of error values during the amendment phase for the ICC dataset and all variants of LEMMING. The three rows are the class selection variants while the two columns show the instance selection variants.

A.3 Detailed Correlation Results

Tables A.1 and A.1 show the detailed results of the new coherence measures and coherence measures that have been proposed by related work. The description of the latter can be found ins Section 5.3.1. The experiment description can be found in Section 5.3.2. The discussion of the most important results can be found in Section 5.3.3.

Table A.1.: Correlations and rankings (in paranthesis) of the coherence measures $\mathcal{C}_{V2}$, $\mathcal{C}_P$ and the measures listed in Section 5.3.1 except the measures proposed by Rosner et al. [233] and Aletras et al. [10].

Name	$\mathcal{C}_{V2}$	$\mathcal{C}_P$	$\mathcal{C}_{\text{UMass}}$	$\mathcal{C}_{\text{UCI}}$	$\mathcal{C}_{\text{NPMI}}$	$\mathcal{C}_S$	$\mathcal{C}_O$	$\mathcal{C}_F$
$\mathcal{S}$	S^{one}_{all}	S^{one}_{pre}	S^{one}_{pre}	S^{one}_{one}	S^{one}_{one}	S^{all}_{one}	S^{set}_{set}	S^{one}_{any}
$\boldsymbol{\mathcal{P}}$	$\mathcal{P}_{sw(110)}$	$\mathcal{P}_{sw(70)}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{sw(10)}$	$\mathcal{P}_{sw(10)}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{sw(300)}$	$\mathcal{P}_{sw(300)}$
$\mathfrak{M}$	$\tilde{m}_{\text{cos},m_{nlr},1}$	m_f	m_{lc}	m_{lr}	m_{nlr}	m_{ls}	m_o	m_f
$\mathfrak{A}$	a_a	a_a	a_a	a_a	a_a	a_a	$*$	a_a
20NG	0.832 (691.5)	0.825 (1 810.0)	0.555 (95 887.0)	0.747 (17 055.0)	0.809 (4 107.0)	-0.540 (447 307.0)	0.185 (220 157.5)	0.729 (21 238.0)
Genomics	0.726 (3 147.5)	0.721 (2 896.0)	0.461 (47 398.0)	0.602 (20 427.5)	0.671 (9 804.0)	-0.351 (422 975.0)	NaN (501 449.0)	0.123 (238 312.0)
NYT	0.820 (11.5)	0.757 (2 215.0)	0.519 (50 798.0)	0.751 (2 641.0)	0.798 (267.0)	-0.334 (389 079.0)	NaN (290 627.5)	0.657 (13 485.0)
RTL-NYT	0.736 (151.5)	0.720 (886.0)	0.099 (316 872.0)	0.544 (60 306.0)	0.659 (14 433.0)	0.487 (148 803.0)	0.236 (218 296.5)	0.629 (24 643.0)
RTL-Wiki	0.684 (144.5)	0.645 (3 901.0)	0.336 (164 631.0)	0.548 (28 928.0)	0.609 (9 725.0)	0.321 (106 464.0)	0.156 (266 499.5)	0.608 (10 085.0)
Movie	0.542 (11 963.5)	0.533 (16 355.0)	0.143 (345 189.0)	0.447 (63 203.0)	0.452 (60 316.0)	0.351 (146 733.0)	0.285 (109 348.5)	0.494 (35 381.0)
Avg. rank	2 685.0	4 677.2	170 129.2	32 093.4	16 442.0	276 893.5	267 729.8	57 190.7
Std. dev.	4 287.6	5 304.1	120 432.8	22 373.6	20 132.6	144 546.0	118 983.1	81 406.1

Table A.2.: Correlations and rankings (in paranthesis) of the coherence measures proposed by Rosner et al. [233] and Aletras et al. [10].

Name	$\mathcal{C}_{\text{one-any}}$	$\mathcal{C}_{\text{any-any}}$	$\mathcal{C}_{\text{one-all}}$	$\mathcal{C}_{\text{cos}}$	$\mathcal{C}_{\text{Dice}}$	$\mathcal{C}_{\text{Jac}}$	$\mathcal{C}_{\text{cen}}$
$\mathcal{S}$	S^{one}_{all}	S^{one}_{pre}	S^{one}_{any}	S^{one}_{one}	S^{one}_{one}	S^{one}_{one}	S^{one}_{set}
$\boldsymbol{\mathcal{P}}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{bd}$	$\mathcal{P}_{cw(5)}$	$\mathcal{P}_{cw(5)}$	$\mathcal{P}_{cw(5)}$	$\mathcal{P}_{cw(5)}$
$\mathfrak{M}$	m_d	m_d	m_d	$\tilde{m}_{\text{cos},m_{nlr},1}$	$\tilde{m}_{\text{Dice},m_{nlr},2}$	$\tilde{m}_{\text{Jac},m_{nlr},2}$	$\tilde{m}_{\text{cos},m_{nlr},1}$
$\mathfrak{A}$	a_a	a_a	a_a	a_a	a_a	a_a	a_a
20NG	0.822 (2 090.0)	0.765 (12 681.0)	0.599 (76 170.0)	0.790 (7 555.0)	0.740 (18 817.0)	0.738 (21 209.0)	0.765 (12 579.0)
Genomics	0.452 (50 037.0)	0.357 (89 345.0)	-0.042 (341 280.5)	0.640 (14 587.5)	0.641 (14 479.5)	0.633 (20 499.0)	0.654 (12 671.0)
NYT	0.612 (22 738.0)	0.575 (31 523.0)	0.447 (91 691.0)	0.733 (4 082.5)	0.650 (14 881.0)	0.659 (8 417.0)	0.724 (5 009.0)
RTL-NYT	0.438 (114 219.0)	0.406 (130 845.0)	0.335 (178 236.0)	0.630 (23 859.0)	0.683 (7 749.5)	0.665 (44 337.0)	0.615 (30 191.0)
RTL-Wiki	0.499 (52 244.0)	0.459 (74 832.0)	0.438 (90 708.0)	0.579 (18 203.0)	0.565 (22 587.0)	0.535 (23 730.0)	0.571 (20 812.0)
Movie	0.454 (59 269.0)	0.408 (85 740.0)	0.489 (37 356.0)	0.465 (51 831.5)	0.474 (46 326.0)	0.458 (28 700.0)	0.467 (50 491.0)
Avg. rank	50 099.5	70 827.7	135 906.9	20 019.8	20 806.7	24 482.0	21 958.8
Std. dev.	34 769.6	38 964.2	101 035.3	15 646.6	12 274.6	10 779.9	14 970.6

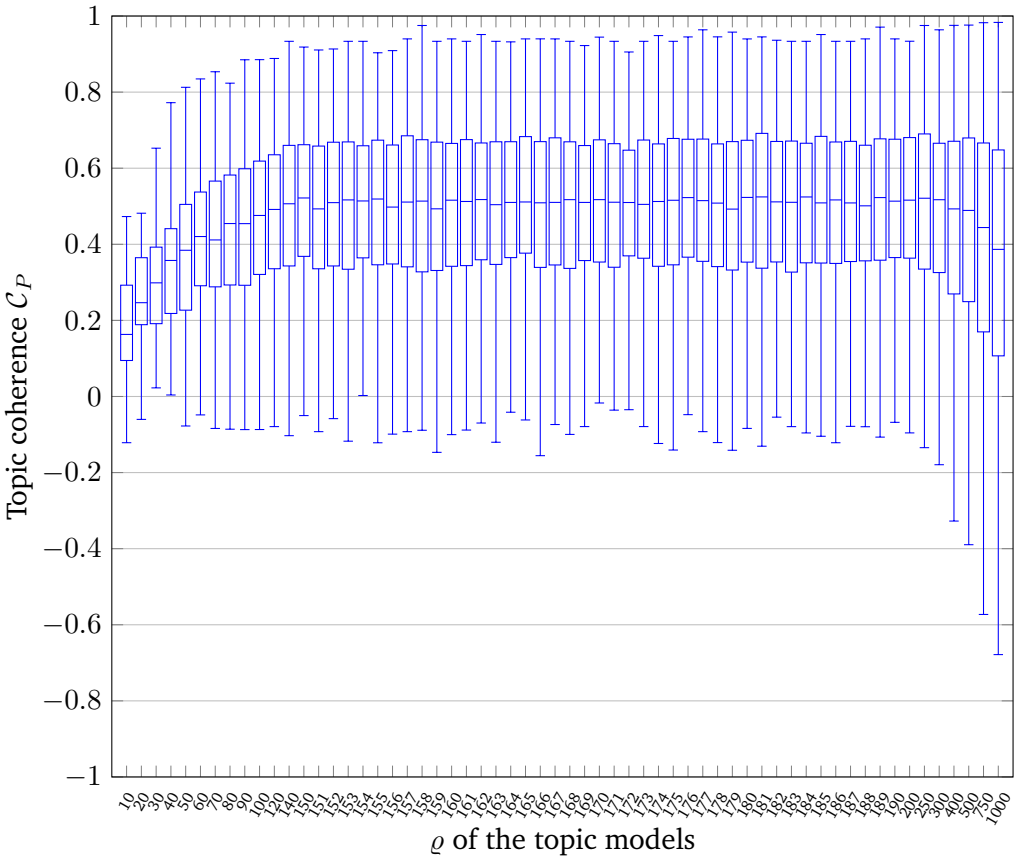


Figure A.4.: Box plots of the topic coherence values ($\mathcal{C}_P$) of different topic models with different numbers of topics.

A.4 Detailed Measure Comparison

In an additional experiment, we compare the coherence-based evaluation of topic models with the measures proposed by Griffiths et al. [110] and Arun et al. [19], which are explained in detail in Section 2.2.3. First, we create a subset of the Wikipedia corpus comprising 337 754 documents with 236 million word tokens and 4.4 million different word types. In a first step, we generate topic models with $\varrho = \{50, 100, 150, 200, 250, 300, 400\}$ and measure their coherence. We measure the coherence of the generated models using the $\mathcal{C}_P$ measure. Then, we generate additional models with different numbers of topics in areas in which the coherence seems to be high. This holds especially for the range $[150, 200]$. In addition, we use the measures proposed by Griffiths et al. [110] and Arun et al. [19] for the same models to compare the results. For each number of topics, we generate three models and report the average across these models.

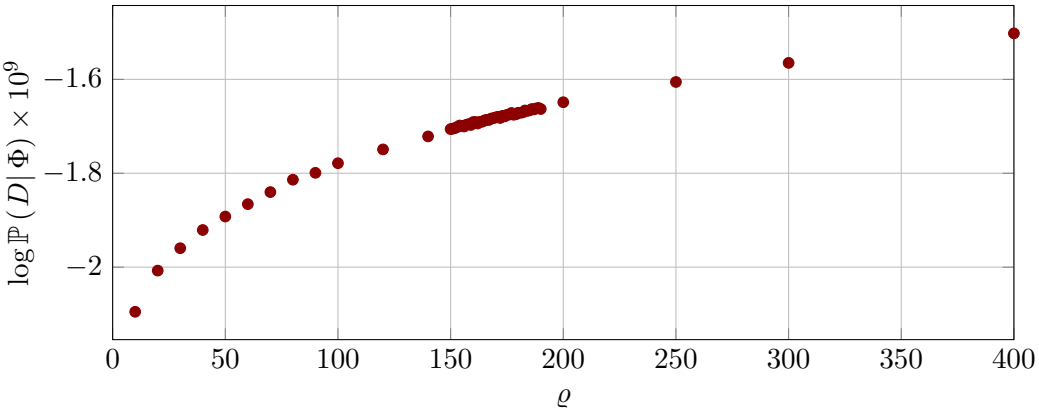


Figure A.5.: Values of $\log(\mathbb{P}(D|\Phi))$ calculated for the generated models.

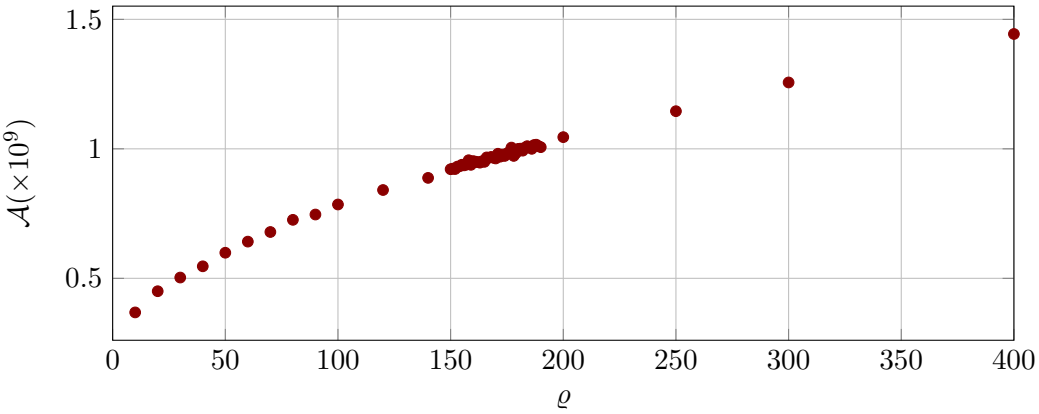


Figure A.6.: Average values of the measure $\mathcal{A}$ calculated for the generated models.

Figure A.4 shows the results of the coherence measure as box plots. The average coherence value increases with the number of topics until it reaches a plateau with around 150 topics. All models with less than 400 topics seem to have a similar coherence value. From 400 topics on, the average coherence value of the models drops. To ensure that this observation is not solely based on a single model, we added models with 500, 750 and 1000 topics.[2] They confirm the observation. Figures A.5 and A.6 show the values of $\mathbb{P}(D|\Phi)$ and $\mathcal{A}$, respectively. Both values increase with the number of topics. Hence, the probability $\mathbb{P}(D|\Phi)$ would choose the topic model with 400 topics as best model. The $\mathcal{A}$ measure instead would choose the topic model with 10 topics. These observations confirm the results from our experiment in Section 5.4.3.

A.5 Questionnaire

This section shows the 10 questions of the LODCat questionnaire of the third experiment described in Section 5.4. All questions in the questionnaire have the same structure. An example of these questions is the following:

Given the dataset 0ca, which of the following 4 topics does not fit to the dataset?

In the online questionnaire, the name of the dataset in the question is linked to the RDF file to give the user access to the dataset. The question is followed by the list of topics. The topics have been ordered randomly when creating the questions. We list the topics in this order, but mark the meaning of the topics, i.e, whether their are 1st, 2nd, 3rd or the intruder topic (I). Each topic comes with a title and 10 top words.

1. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/0ca881a1fc5817a3575d2ff6191cc622%3Ftype=hdt.n3`
   - I **College basketball**
     state, basketball, conference, university, ncaa, kentucky, carolina, tournament, man, college
   - 2nd **Philosophy**
     language, study, social, theory, university, work, culture, history, press, philosophy
   - 3rd **Information**
     use, system, software, user, datum, computer, include, information, support, service

[2] We do not report the values of $\mathbb{P}(D|\Phi)$ and $\mathcal{A}$ for these models since the evaluation of the large models consumed too much RAM.

1st **University**
university, college, research, science, institute, professor, award, work, study, society

2. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/0df3609cdd77b40d2a06ea88d09f656b%3Ftype=hdt.n3`
   1st **Nausea**
   disease, patient, may, cause, treatment, use, cancer, blood, symptom, include
   I **Train**
   railway, station, line, train, service, locomotive, rail, passenger, bus, transport
   2nd **Philosophy**
   language, study, social, theory, university, work, culture, history, press, philosophy
   3rd **Football**
   player, footballer, football, expatriate, fc, people, bear, play, birth, club

3. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/6f3341506ee83bd28b787c8e2f6858c1%3Ftype=hdt.n3`
   1st **Information**
   use, system, software, user, datum, computer, include, information, support, service
   2nd **Physicist**
   prize, physics, award, design, mitchell, research, science, physicist, quantum, brain
   3rd **University**
   university, college, research, science, institute, professor, award, work, study, society
   I **Sweden**
   danish, norwegian, norway, denmark, club, swedish, sweden, copenhagen, sc, people

4. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/c89db4032b3809c7ef9e57e5485ca2f8%3Ftype=hdt.n3`
   I **Genus**
   species, genus, describe, family, find, name, fish, genera, plant, america

3rd **Sir**
john, london, son, king, william, british, sir, english, die, death
2nd **Building**
building, house, new, park, street, area, south, build, town, centre
1st **University**
university, college, research, science, institute, professor, award, work, study, society

5. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/df7e479638564102139dc1ff359ebc15%3Ftype=hdt.n3`
2nd **University**
university, college, research, science, institute, professor, award, work, study, society
1st **Information**
use, system, software, user, datum, computer, include, information, support, service
I **Flower**
plant, bird, flower, species, leaf, long, flora, grow, name, tree
3rd **Atlantic**
album, song, release, band, music, chart, record, single, track, records

6. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/113e514ef0a94b738ce37350525cae0d%3Ftype=hdt.n3`
2nd **Portugal**
airport, international, brazil, portuguese, são, romanian, portugal, brazilian, language, romania
1st **China**
chinese, china, singapore, li, wang, shanghai, chen, beijing, hong, zhang
I **Season**
game, team, season, win, first, league, play, point, second, record

7. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/1d6c4d4a80a507331801dda33f631097%3Ftype=hdt.n3`
1st **Upland**
river, island, area, mountain, park, north, south, forest, water, land
I **Govinda (actor)**
film, indian, tamil, singh, role, kumar, actor, hindi, award, telugu

2nd **Portugal**
airport, international, brazil, portuguese, são, romanian, portugal, brazilian, language, romania

8. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/b056341b71e7ecb9ed12f144c085dc60%3Ftype=hdt.n3`
   I **Painting**
   art, museum, work, artist, painting, gallery, exhibition, painter, collection, new
   1st **Germany**
   german, der, die, und, germany, flag, berlin, austrian, vienna, work
   2nd **Philosophy**
   language, study, social, theory, university, work, culture, history, press, philosophy

9. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/79d5ea832dd19745b7707644af618ef1%3Ftype=hdt.n3`
   2nd **Portugal**
   airport, international, brazil, portuguese, são, romanian, portugal, brazilian, language, romania
   I **American hockey league**
   hockey, player, season, team, ice, toronto, nhl, league, new, play
   1st **Business**
   company, bank, business, million, market, financial, industry, year, group, tax

10. Dataset `https://hobbitdata.informatik.uni-leipzig.de/lodcat/lodalot/evaluation/rdf/979c3e5c7aa2f1937a6939ec9e16b965%3Ftype=hdt.n3`
    1st **Upland**
    river, island, area, mountain, park, north, south, forest, water, land
    2nd **Australia**
    australia, australian, south, sydney, wales, melbourne, victoria, new, queensland, western
    I **Uttar pradesh**
    india, indian, state, delhi, pradesh, bangladesh, bengal, tamil, national, nepal