When Facts Expire : Hybrid Approaches for Temporal Validation of Facts in Multiple and Heterogeneous Knowledge Graphs

Thibaut Soulard

▶ **To cite this version:**

Thibaut Soulard. When Facts Expire : Hybrid Approaches for Temporal Validation of Facts in Multiple and Heterogeneous Knowledge Graphs. Artificial Intelligence [cs.AI]. Université Paris-Saclay, 2025. English. ⟨NNT : 2025UPASG105⟩. ⟨tel-05441524⟩

HAL Id: tel-05441524

https://theses.hal.science/tel-05441524v1

Submitted on 5 Jan 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

université PARIS-SACLAY

When Facts Expire: Hybrid Approaches for Temporal Validation of Facts in Multiple and Heterogeneous Knowledge Graphs

Quand les faits expirent : approches hybrides de validation temporelle de faits dans des graphes de connaissances multiples et hétérogènes

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée au **Laboratoire interdisciplinaire des sciences du numérique** (Université Paris-Saclay, CNRS)
sous la direction de **Fatiha SAÏS**, Professeure
et le co-encadrement de **Joe RAAD**, Maître de conférences

Thèse soutenue à Paris-Saclay, le 15 décembre 2025, par

Thibaut SOULARD

Composition du jury

Membres du jury avec voix délibérative

Wassila OUERDANE Professeure, CentraleSupélec, Université Paris-Saclay	Présidente
Marieke VAN ERP Chargé de recherche, équivalent HDR, KNAW Humanities Cluster	Rapporteure & Examinatrice
Oscar CORCHO Professeur, Universidad Politécnica de Madrid	Rapporteur & Examinateur
Catherine FARON Professeure, Université Côte d'Azur	Examinatrice
Thomas GUYET Chargé de recherche, HDR, Centre Inria de Lyon	Examinateur

THESE DE DOCTORAT
NNT : 2025UPASG105

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: When Facts Expire: Hybrid Approaches for Temporal Validation of Facts in Multiple and Heterogeneous Knowledge Graphs
Keywords: Knowledge Graph, Deep Learning, Temporality, Key Transfer, Ontology Alignment

Abstract: This doctoral thesis addresses the challenges of correcting and ensuring the completeness of large-scale Knowledge Graphs (KGs), focusing on the temporal validity of facts. KGs often exhibit inaccuracies and gaps, particularly for dynamic facts. Robust correction mechanisms are crucial, as fact veracity depends on the source, extraction, and validity period, making temporal information indispensable for validating relations such as the residence of a person or his employment.

To overcome these limitations, this research develops new frameworks and resources. A new framework improves entity alignment between KGs through the transfer and verification of precomputed keys. To bridge a gap in direct property and class alignment, KG Nexus, a LOD-Cloud catalog, was created. The thesis also explores symbolic temporal constraints (Allen's Algebra) to validate the temporal context of facts. Finally, a new deep learning framework is presented for temporal knowledge graph embedding tools, relying also on Allen's Algebra relations, resolving the need to represent every timestamp.

Collectively, these contributions facilitate more robust and user-friendly data validation in KGs, improving LOD Cloud utilization and cross-validation. This research advances KG correction and temporal reasoning, while also highlighting future needs in information validation within temporal knowledge graphs.

Titre: Quand les faits expirent : approches hybrides de validation temporelle de faits dans des graphes de connaissances multiples et hétérogènes.
Mots clés: Graphes de connaissances, Deep Learning, Temporalité, Transfert de clés, Alignement d'ontologie

Résumé: Cette thèse de doctorat aborde les défis de correction et d'exhaustivité des graphes de connaissances (KGs) à grande échelle, en se concentrant sur la validité temporelle des faits. Les KGs présentent souvent des inexactitudes et des lacunes, notamment pour les faits dynamiques. Des mécanismes de correction robustes sont cruciaux, car la véracité des faits dépend de la source, de l'extraction et de la période de validité, rendant l'information temporelle indispensable pour valider des relations comme la résidence d'une personne ou son emploi.

Pour surmonter ces limites, ces travaux de thèse développent de nouveaux cadres et ressources. Un nouveau cadre améliore l'alignement des entités entre KGs via le transfert et la vérification de clés précalculées. Pour combler une lacune en alignement direct de propriétés et de classes, KG Nexus, un catalogue LOD-Cloud, a été créé. La thèse explore également les contraintes temporelles symboliques (algèbre d'Allen) pour valider le contexte temporel des faits. Enfin, un nouveau cadre d'apprentissage profond est présenté pour les outils d'intégration de graphes de connaissances temporels, reposant aussi sur l'algèbre d'Allen, résolvant ainsi la nécessité de représenter chaque point temporel.

Collectivement, ces contributions facilitent une validation plus robuste et collégiale des données dans les KGs, améliorant l'utilisation du LOD Cloud et la validation croisée. Cette recherche fait progresser la correction des KGs et le raisonnement temporel, tout en soulignant les besoins futurs en validation d'informations dans les graphes de connaissances temporels.

Remerciements

I would like to first thank Fatiha and Joe for dedicating so much time throughout this PhD to guide me and allow me to grow both as a researcher and as a teacher. I am particularly grateful for the opportunities you gave me to assist with teaching and later to teach classes myself. Thank you for following me through all the peregrinations of these nearly four years. I will always be grateful for the many opportunities you offered me, including the trips you encouraged me to take, as well as for your advice, even during the moments when I have been stubborn and very focused on my ideas. I will always remember the moments shared with you during the multiple PFIA conferences we attended together, where you showed that you were just as kind in your personal interactions as you are in your professional life.

Thank you to Prof. Oscar Corcho and Dr. Marieke Van Erp for taking the time and the care to review this manuscript and for providing me with all the elements needed to improve it. I would also like to thank Prof. Wassila Ouerdane, Prof. Catherine Faron, and Dr. Thomas Guyet for agreeing to take part in my PhD defense and for the depth and quality of the discussions generated by your questions. I would like to express particular thanks to Dr. Thomas Guyet for the conversation we had in La Rochelle and for the email you sent after the defense, as your words were always kind and consistently helped me improve my work.

Nathalie, Emmanuel, Gianluca, Francesca, everyone at LaHDAK and again Joe and Fatiha, thank you for giving me my first opportunities to discover the world of research through several internships. Each of you showed me essential aspects of what it means to be a great researcher, through your kindness, your availability, and the time you dedicated to guiding me.

Alan and Jonathan, thank you for sharing this entire journey with me (and even longer with you Jonathan), from the very beginning to the very end, with many unexpected side quests along the way. Thank you for always being there, for your support, and for enduring my most delusional moments. From very early mornings with you Alan, far too early actually, to very late evenings at the Crous of Centrale or around the pool table with the both of you, often accompanied by the guards, these moments will remain unforgettable and timeless.

Olivier, thank you so much for all the coffee breaks and lunches we shared, during which you always gave me the opportunity to benefit from your more experienced perspective on my PhD and my academic career. Each time, you helped me put my difficulties into perspective, and your attentive listening was always there when I needed it the most. I very much look forward to seeing you defend your work and to discovering the perspective through which you will present it, as your way of approaching research has always been a reference I aspire to follow.

Camille we also started this path from the bachelor together and we have gone through a change of University or just name, Covid and finally a full PhD together without even realizing it. We have always supported each other and I was very happy to have seen you defend yours, within a few days

from each other the same way we did for each of our achievements in our academic path.

To the original PhD crew, Armita, Lucas, Alexandra, and Taha, thank you for showing me how to survive this path. Your advice has been invaluable, and in particular the answer you gave to my question in Saint Etienne, Armita and Lucas, now makes complete sense to me and showed me that this path was never meant to be easy.

Alexandre, Lilia, and Yacine, thank you for bringing so much energy and life to the lab during your internships and now throughout your PhDs, even if it is from further away for you Yacine. It is a real pleasure to see you following a similar path, and I very much look forward to attending each of your defenses and celebrating your achievements.

A heartfelt thank you to Michael and Frank for welcoming me so warmly in Amsterdam, allowing me to discover a different way of doing research and to integrate into a new team. I would also like to thank from the bottom of my heart all the PhD students with whom I shared so many great and timeless moments, Giacomo, Romy, Ritten, Thiviyan, Márk, Ruud, Inès, Yannick, Taraneh, Andrea, Andreas, Dimitrios, Nikos and many more, as well as the many postdocs and professors who provided inspiring examples throughout this experience.

To all my friends, Amira, Filipe, Lucas, Corentin, Michael, Calvin, Emma, taller Thibault, and so many others, thank you for your constant support. Each of you have contributed greatly to the person I am today, and your presence has always been a source of strength pushing me toward a better version of myself.

Gabi, meeting you was one of the greatest highlights of this PhD journey. You made this final year truly special, and your support during the writing phase, despite all the lows I went through, meant more than I can express. Amsterdam, Utrecht, and now Rotterdam have become home for me through you. I cannot wait to continue exploring Spain and the rest of the world together.

Mathieu, Anaïs, Christian, and especially Maman, thank you for always being there for me. Your support was essential, particularly during the many side quests that shaped this PhD, including the renovation of my apartment and the transition from sleeping on the floor to using an actual bed. Papa, even if we are more distant now, thank you for everything you have given me. Florian, Carole, Lola, and Ambre, thank you for always welcoming me with a smile each time I come to see you far in the north. To my grandparents, thank you for giving me your love, whether from the sky or still here with us. All of you gave me your unconditional love and have taught me the most important skill, perseverance.

Finally, I would like to always remember that I was able to do it. I hope that in the future I will look back on these moments of doubt and remember that, regardless of how low I felt, I was always able to get through them as best as I could, thanks to the help, trust, and love of the people around me.

Contents

Abstracts 3

Table of Contents 10

List of Figures 12

List of Tables 14

1 Introduction 15
- 1.1 Motivation . . . 15
- 1.2 Research Problem, Objectives, and Structure of the Thesis . . . 17

2 Fundamentals 21
- 2.1 Knowledge Graph . . . 22
  - 2.1.1 Ontology Layer . . . 22
  - 2.1.2 RDF Data Graph . . . 23
  - 2.1.3 Interaction Between Schema and Data Layers . . . 24
- 2.2 Temporal Knowledge Graphs . . . 25
  - 2.2.1 From Static to Temporal Knowledge Representation . . . 25
  - 2.2.2 Technical Implementation of Temporal Context . . . 26
  - 2.2.3 Representation of Temporal Relations . . . 29
  - 2.2.4 Challenges and Opportunities in Temporal Knowledge Graphs . . . 31
- 2.3 Static Embedded Representations . . . 32
  - 2.3.1 A simple translational model – TransE . . . 32
  - 2.3.2 Beyond TransE: Capturing Complex Patterns . . . 34
  - 2.3.3 Applications: Knowledge Graph Completion and Entity Alignment . . . 34
  - 2.3.4 Towards Temporally Aware Embeddings . . . 35
  - 2.3.5 Summary . . . 36
- 2.4 Linking Knowledge . . . 36
  - 2.4.1 Predefined Alignments . . . 36
  - 2.4.2 Linking of Entities . . . 37
  - 2.4.3 Ontology Alignments . . . 39
  - 2.4.4 Summary . . . 40
- 2.5 Synthesis . . . 40

3 Explainable Temporal Fact Validation Through Constraints Discovery 43
- 3.1 Introduction . . . 44
- 3.2 Related Work . . . 45
- 3.3 Background and Preliminaries . . . 47
  - 3.3.1 Temporal Knowledge Graphs . . . 47

3.3.2 Temporal Relations . . . 48
3.4 Approach Overview . . . 48
3.5 Temporal Constraint Discovery from KGs . . . 50
3.5.1 Entity Time Sequence . . . 50
3.5.2 Entity Time Sequence Comparison . . . 51
3.5.3 Simple Temporal Constraint . . . 52
3.5.4 Complex Temporal Constraints . . . 53
3.5.5 Temporal Constraints Generalisation . . . 56
3.5.6 Temporal Constraints for non-Functional Temporal Properties . . . 59
3.6 Constraint-based Temporal Fact Validation . . . 60
3.6.1 Single Constraint Application . . . 61
3.6.2 Overall Constraint Application . . . 62
3.7 Experimental Evaluation . . . 63
3.7.1 Datasets and Experimental Setup . . . 63
3.7.2 Key Experiments and Results . . . 64
3.7.3 Summary and Takeaways . . . 67
3.8 Conclusion . . . 68
3.9 Future Work . . . 69
3.9.1 Scalability and Explainability. . . . 69
3.9.2 Cross-Knowledge Graph Transfer . . . 70
3.9.3 Constraints Across Classes and Property Paths . . . 70
3.9.4 Advanced Temporal Reasoning. . . . 71
3.9.5 Benchmarking and Comparative Evaluation. . . . 71

4 Benchmarking Temporal Validity in Knowledge Graphs 73
4.1 Introduction . . . 73
4.2 Related Work . . . 75
4.2.1 Temporal Knowledge Graph Reasoning Tasks . . . 75
4.2.2 Temporal Link Prediction Dataset Characteristics . . . 78
4.3 Benchmark Description . . . 82
4.4 Benchmark Construction . . . 84
4.4.1 Data extraction . . . 84
4.4.2 Graph Preprocessing . . . 85
4.4.3 Negative Sample Generation . . . 88
4.4.4 Semantic Information Integration . . . 89
4.4.5 Benchmark Statistics . . . 90
4.5 Temporal Fact Validation Task . . . 91
4.5.1 Classical framework for both static and temporal completion . . . 92
4.5.2 Preliminary Experimental Analysis of the Adaptation to Temporal Validation . . . 96
4.5.3 Comparative Evaluation of Symbolic and Adapted Frameworks . . . 101
4.6 Conclusion . . . 102
4.7 Future Work . . . 103
4.7.1 Overlapping negative sampling . . . 103

4.7.2 Multi-scale temporal granularity . . . 103
4.7.3 Uncertain and fuzzy temporal annotations . . . 104
4.7.4 Improving the symbolic approach through semantic generalization . . . 104
4.7.5 Diverse datasets . . . 104

5 Studying Key Transferability for Scalable Data Linking Between Knowledge Graphs 105
5.1 Introduction . . . 106
5.2 Related Work . . . 107
5.3 Definitions and Notions . . . 108
5.4 Key Transfer Methodology . . . 110
5.4.1 Property Alignment . . . 110
5.4.2 Key Quality Measures . . . 112
5.4.3 Key Transfer Approach . . . 113
5.5 Experimental Evaluation . . . 115
5.5.1 Graphs with Heterogeneous Vocabularies . . . 115
5.5.2 Graphs with Harmonized Vocabulary . . . 120
5.6 Conclusion . . . 123
5.7 Future Work . . . 125
5.7.1 Comparative evaluation of key discovery methods . . . 125
5.7.2 Key discovery with native relative exception control . . . 125
5.7.3 Confidence-aware alignment handling . . . 125
5.7.4 Neuro-symbolic framework . . . 126
5.7.5 Temporally aware alignment . . . 126

6 KG Nexus: Bridging the Web of Data 127
6.1 Introduction . . . 127
6.2 Related Work . . . 129
6.2.1 Vocabulary and Ontology Repositories . . . 129
6.2.2 Link Repositories for Entity Alignment . . . 130
6.2.3 Centralised Entity Naming Services . . . 130
6.2.4 Integrated Domain-Specific Knowledge Graphs . . . 130
6.2.5 Infrastructure-Level Platforms for RDF Data . . . 130
6.3 Creation of the KG Nexus . . . 131
6.3.1 Vocabulary Reduction, Harmonisation and Augmentation . . . 133
6.3.2 Transformer-based Ontology Alignment . . . 135
6.3.3 Equivalence Relationships Computation . . . 136
6.3.4 Bring your own Alignment Tool . . . 138
6.3.5 Knowledge Graph Incorporation . . . 138
6.3.6 Key Incorporation . . . 139
6.4 Data analytics . . . 140
6.4.1 Raw information of the vocabularies . . . 140
6.4.2 Alignments Statistics . . . 141
6.4.3 Incorporation of Knowledge Graphs and Key Discovery . . . 141

6.5 Use Cases . . . 142
6.5.1 Knowledge Graph Enrichment . . . 142
6.5.2 Schema Consistency Verification . . . 143
6.5.3 Concept Shift Analysis . . . 143
6.5.4 Federated Query Optimization . . . 143
6.5.5 Key Quality Profiling and Benchmarking . . . 143
6.5.6 Automatic Schema Bootstrapping for New KGs . . . 144
6.6 Conclusion . . . 144
6.7 Future Work . . . 145
6.7.1 Multilingual Support . . . 145
6.7.2 Exploration through Dashboards . . . 145
6.7.3 Transversal Applications . . . 145

7 Conclusion and Perspectives **147**
7.1 Conclusion . . . 147
7.2 Active Collaboration: Deep Temporal Reasoning with Allen Algebra . . . 149
7.3 General Perspectives . . . 151
7.3.1 From Temporal Reasoning to Hybrid Intelligence . . . 151
7.3.2 Integrating Human Expertise in the Validation Loop . . . 151
7.3.3 Toward Sustainable and Open Knowledge Infrastructures . . . 152
7.4 Final Note . . . 153

A Résumé en Français **155**

B List of Publications **158**

Bibliography **161**

List of Figures

2.1 Wikidata temporal representation . . . 28
2.2 Allen's interval algebra: seven distinct, exhaustive, and qualitative temporal relations. . . 30
2.3 Extended Allen's Algebra . . . 31
2.4 Example of training samples for TransE. . . . 33

3.1 Examples of Allen's time interval relations . . . 48
3.2 Approach workflow . . . 49
3.3 An example of two time-sequences S and S' containing 9 and 3 time intervals respectively. The arrows refer to inter-sequence comparisons. . . . 50
3.4 Simple temporal constraints . . . 53
3.5 Complex Temporal Constraints Tree (CT^2) . . . 54
3.6 Joint Complex Constraints . . . 55
3.7 Disjoint Complex Constraints . . . 57
3.8 Example of value specialization . . . 59
3.9 Constraint based Temporal Fact Validation framework . . . 60
3.10 Insertion strategies for a new interval (blue) . . . 61
3.11 Symbolic combination strategy framework. . . . 62
3.12 Machine learning combination strategy framework. . . . 63

4.1 Number of timestamps per year in the reduced temporal scope (TR). . . . 85
4.2 Number of timestamps per year in the full temporal scope (FD), logarithmic scale on the y-axis. . . . 86
4.3 Percentage of entities retained per degree threshold N. . . . 87
4.4 Schematic of negative sampling by temporal corruption. . . . 88
4.5 Radar plots for the reduced temporal scope (TR), comparing day-level (TR-D) and year-level (TR-Y) settings through logarithmic scale. . . . 92
4.6 Radar plots for the full temporal scope (FD), comparing day-level (FD-D) and year-level (FD-Y) settings through logarithmic scale. . . . 93
4.7 Examples of predicted intervals (numbered intervals) compared to the expected interval (G). . . . 95
4.8 Accuracy of Timplex model with respect to the threshold value for the different settings of Extra Small. . . . 98
4.9 Accuracy of TNT Complex model with respect to the threshold value for the different settings of Extra Small. . . . 99
4.10 Accuracy of the symbolic framework with respect to the threshold value for the different settings of Extra Small. . . . 101

5.1 Key transfer workflow. . . . 114

6.1 Overall workflow of KG Nexus creation. 131
6.2 Schema of KG Nexus vocabulary representation (LOV-RHA) 135
6.3 Similarity and ontology alignment representation in KG Nexus 137
6.4 KG and Keys incorporation in KG Nexus 139

7.1 Example of transformation from an entity view to a relational view 150

List of Tables

3.1 Inter-Comparison matrix $M_{\triangleright}$ for the sequences S and S' from Figure 3.3. 52
3.2 Intra-Comparison matrix $M_{\triangleleft}$ for the sequences S and S' from Figure 3.3. 52
3.3 Description of the three datasets extracted from Wikidata 64
3.4 Constraints decision strategies 64
3.5 Constraints value specialization 65
3.6 Interval insertion strategies comparison 66
3.7 Thresholds comparison 67

4.1 Structural and temporal characteristics of TKG datasets 78
4.2 Entity, relation, and temporal statistics for TKG datasets (temporally contextualized edges only). 79
4.3 Statistics of the benchmark positive samples: counts of facts, unique entities (with average degree), relations (with average use), and timestamps (with average use) 91
4.4 Values of each metric with respect to the example in Figure 4.7. 95
4.5 Accuracy for the best threshold on the set with the accuracy metric (OOM means out of memory and OOT means out of time) 102

5.1 Class $Person$/$Human$ in DBpedia/Wikidata 116
5.2 Key discovery results for the Person/Human datasets using different relative exception rates (SK $\rightarrow$ Source Key, RK $\rightarrow$ Rewritten Key) 117
5.3 Discriminability of transferred keys according to the maximum relative exception rate. 118
5.4 Execution time in minutes for each scenario and processing step. (SK $\rightarrow$ Source Key, RK $\rightarrow$ Rewritten Key) 119
5.5 Example of non-monotonic keys. 119
5.6 Characteristics of the 8 classes in DBpedia and YAGO. 120
5.7 Execution time for source key discovery, source key verification, and transferred key verification for each scenario (i.e., from DBpedia, from YAGO, and from intersection). . 121
5.8 Number and average size of source keys and transferred keys per dataset and scenario (i.e., from DBpedia, from YAGO, and from intersection). 122
5.9 Data linking performance as a function of the maximum relative exception threshold {0%, 0.5%, 2%, 5%}. *D* and *Y* refer to DBpedia and YAGO as source graphs, respectively. *I* refers to the classical intersection-based approach. 124

6.1 Number of predicates per type of Entity. 133
6.2 Number of predicates describing properties 134
6.3 Number of predicates describing Classes 134
6.4 Top five vocabularies with respect to the number of Classes or Properties. 140
6.5 Top 5 languages with respect to the number of Vocabularies or Description. 141
6.6 Number of keys identified per dataset using the SAKey algorithm. 142

Chapter 1

Introduction

1.1 . Motivation

Knowledge does not merely accumulate; it re-emerges. As facts are revised, entities evolve, and interpretations shift, what we consider true today may differ from what was established yesterday. Knowledge is therefore not a static repository but a living, self-correcting process that continuously adapts to new contexts and evidence. Yet, most computational systems still encode it as immutable statements, detached from the temporal dynamics that shape them. This dissonance between the fluidity of knowledge and the rigidity of its representation limits our capacity to reason about the world as it unfolds. Capturing the temporal dimension of knowledge is thus essential for building systems that mirror both the persistence and transformation of meaning over time.

Over the past decade, artificial intelligence has increasingly relied on data-driven models to represent and manipulate knowledge. Large Language Models (LLMs) [Brown et al.(2020)] exemplify this evolution. Trained on massive corpora of text, they are capable of producing coherent and contextually rich responses to almost any query. Their generative flexibility and accessibility have transformed the perception of AI, expanding it from a technical field to a widely used societal tool. However, their success has also exposed fundamental weaknesses that question their ability to serve as reliable reasoning systems.

A first limitation concerns the temporal scope of their knowledge. The information contained within an LLM is bounded by the time at which its training data were collected. As the world evolves, the model's internal representation becomes outdated, causing it to generate answers that are no longer valid. To mitigate this, some models integrate retrieval mechanisms that allow them to query the web during inference. Yet, these solutions remain fragile: many websites or companies such as Cloudflare[1] now restrict or block automated access, and the information retrieved is often unstructured, redundant, or unverifiable.

A second and deeper limitation emerges from the growing contamination of training

[1] https://blog.cloudflare.com/declaring-your-aindependence-block-ai-bots-scrapers-and-crawlers-with-a-single-click/

data by model-generated content. As LLM outputs proliferate across the web, newer models increasingly learn from data that originate from previous generations of similar models. This recursive process reinforces existing biases and factual inaccuracies, producing a self-referential cycle in which models learn from their own approximations rather than from external reality. In such conditions, even models trained on ever-larger datasets risk converging toward synthetic coherence rather than factual reliability.

This situation revives the importance of a family of resources that once stood at the heart of AI research: *Knowledge Graphs* (KGs). Emerging from the Semantic Web vision [Berners-Lee and Hendler(2001)], they were conceived to organize information through explicit entities, relations, and ontologies, thereby providing data with formal and interoperable meaning that enables reasoning across heterogeneous sources. Following the breakthrough of Google's Knowledge Graph in 2012[2], other major technology companies such as Amazon, Microsoft, and Meta developed their own large-scale KGs for search, recommendation, and conversational systems. Nevertheless, despite their potential, the broader adoption of KGs progressed slowly [Hogan et al.(2021), Paulheim(2016)]. Their construction required specialized expertise in ontology engineering, logical modeling, and data curation, while maintaining consistency across evolving datasets proved technically and organizationally challenging [Kontokostas et al.(2014), de Groot et al.(2021)]. Consequently, their use remained confined to academic and large industrial environments for nearly a decade, awaiting more accessible tooling and standardized workflows[3]. In parallel, the rise of deep learning and large-scale statistical methods shifted attention toward approaches that prioritized scalability over structure, temporarily overshadowing symbolic paradigms before the recent resurgence of interest in explainable, knowledge-grounded, and temporally aware AI.

Today, Knowledge Graphs are re-emerging as indispensable assets in the pursuit of trustworthy AI [Bhuyan et al.(2024)]. Their structured, explainable, and temporally aware nature provides exactly what data-driven models lack: grounding, traceability, and verifiable semantics. By explicitly encoding when and how facts hold true, KGs and in particular Temporal Knowledge Graphs offer a principled way to contextualize information and maintain its validity over time. Unlike web-scale retrieval, they can be curated, versioned, and updated through transparent processes that ensure continuity and reliability. As such, Knowledge Graphs are not only valuable for explainable reasoning but also represent a promising foundation for a sustainable and auditable memory infrastructure. They can complement neural models by serving as stable repositories of verified information, anchoring learning systems in structured and evolving representations of knowledge.

However, the renewed interest in Knowledge Graphs also reveals the depth of the challenges that remain to make them fully operational at scale. As these resources evolve from static repositories to dynamic reasoning infrastructures, several research questions persist. The first relates to **temporal consistency**: most current graphs encode facts without explicit temporal boundaries, which hinders the ability to reason about when

[2] https://blog.google/products/search/introducing-knowledge-graph-things-not/
[3] https://aws.amazon.com/neptune/knowledge-graphs-on-aws/

statements are valid or how they relate chronologically. Integrating temporal semantics requires discovering, representing, and validating time-dependent dependencies directly within the graph.

The second challenge concerns **heterogeneity and interoperability**. The Web of Data comprises thousands of datasets that differ in vocabulary, granularity, and modeling choices. Despite the adoption of standards such as RDF and OWL, linking across heterogeneous sources remains complex and computationally demanding. Crucially, entity alignments and ontology mappings are often learned or engineered for one pair of datasets, which makes them brittle outside that pair and difficult to share as reusable knowledge. Interoperability therefore calls for methods that represent alignment knowledge in a form that abstracts from any single pair, supports propagation to new datasets, and preserves semantic precision and provenance.

Finally, **sustainability and reusability** emerge as critical concerns for the long-term reliability of the KG ecosystem. Many outputs of research, such as temporal constraints, linking rules, or benchmarks, are produced in isolation and rarely preserved after publication. Ensuring cumulative progress demands infrastructures that make these assets shareable, versioned, and reproducible, in line with the FAIR principles [Wilkinson et al.(2016)] that emphasize Findability, Accessibility, Interoperability, and Reusability.

Together, these dimensions: temporality, interoperability, and sustainability define the conceptual boundaries of this thesis. They frame the search for methods that contribute in making Knowledge Graphs both interpretable and operational at scale, reconnecting the symbolic legacy of structured knowledge with the adaptive power of modern machine learning.

1.2 . Research Problem, Objectives, and Structure of the Thesis

The reflections outlined above converge toward a central research question that captures the scientific and methodological ambitions of this work:

How can reasoning approaches validate the temporal consistency of information across multiple and heterogeneous Knowledge Graphs?

This question articulates the core challenge at the intersection of knowledge representation, reasoning, and sustainability. Addressing it requires designing systems that can identify when a fact remains valid, explain why it does, and update or refute it when temporal inconsistencies arise. More broadly, it calls for frameworks that integrate symbolic reasoning with statistical methods to ensure that knowledge remains both interpretable and adaptable within an ever-evolving informational landscape.

To tackle this overarching problem, the thesis explores three interrelated research directions that collectively define its scientific program.

The first direction concerns the representation and validation of temporal knowledge. While Knowledge Graphs explicitly encode relations among entities, they rarely include

the temporal conditions under which these relations hold. Capturing the evolution of entities and events therefore requires mechanisms capable of discovering, representing, and validating temporal dependencies within the data. This work investigates how temporal constraints can be automatically derived from existing graphs and used to assess the plausibility and consistency of facts over time.

The second research direction focuses on the evaluation and transfer of reasoning knowledge across graphs. Assessing temporal reasoning requires appropriate benchmarks that isolate temporal validity rather than measuring predictive accuracy alone. Furthermore, the ability to reuse and transfer reasoning artifacts, such as discovered constraints or structural keys (a set of properties whose joint values uniquely identify an entity), across multiple graphs is essential for scalability and generalization. This work therefore explores both the construction of benchmarks that test temporal soundness and the design of mechanisms that enable the reuse of reasoning knowledge across heterogeneous datasets.

The third direction relates to sustainability and interoperability of knowledge resources. As the community continues to produce new datasets, benchmarks, and alignments, ensuring that these assets are maintained, shared, and built upon becomes a pressing concern. Many valuable outputs of research remain isolated and quickly outdated. To promote cumulative progress, this thesis emphasizes the creation of open and FAIR-compliant infrastructures that preserve, document, and disseminate reusable knowledge artifacts as part of the broader Semantic Web ecosystem.

These three directions are instantiated through four complementary contributions, each developed in a dedicated chapter.

- Chapter 3 introduces an explainable framework for temporal fact validation, grounded in the automatic discovery of temporal constraints between entity properties. By extending Allen's interval algebra from intervals to sequences, it models complex temporal dependencies such as `before(graduatedAsDoctor, isSurgeonIn)` and uses them to validate facts in web-scale temporal Knowledge Graphs.

- Chapter 4 presents the first benchmark dedicated to temporal validity, designed to evaluated reasoning models on calibration, granularity, and temporal coherence rather than link prediction. It establishes the first standardized evaluation resource for temporal reasoning in Knowledge Graphs.

- Chapter 5 investigates the transfer of keys as a strategy to improve scalability and reuse in data linking. Keys mined in a source graph are projected to a target graph through property alignments, then evaluated by a verification phase that prunes candidates whose discriminative power to distinguish entities falls below a confidence threshold. This pipeline eliminates redundant searches for keys and converts structural knowledge into reusable assets.

- Chapter 6 introduces KG Nexus, a curated and extensible infrastructure that consolidates vocabularies, alignments, and transferable keys. It operationalizes the concept of alignment as a reusable resource and contributes to sustainable and transparent interoperability across the Web of Data.

These contributions are preceded by Chapter 2, which provides the conceptual and methodological foundations necessary for understanding and positioning the rest of the manuscript. It reviews the scientific background of temporal reasoning, knowledge linking, and hybrid approaches to Knowledge Graph validation. Each subsequent chapter then builds upon these elements to address one aspect of the overarching research problem, from temporal reasoning and evaluation to transferability and the creation of sustainable infrastructures.

The thesis concludes in Chapter 7 with a synthesis that revisits these contributions in light of the three overarching research directions. It highlights the complementarity between symbolic reasoning and data-driven methods, showing how their integration supports more robust and interpretable reasoning frameworks. The conclusion also reflects on the broader perspectives opened by this work, such as the potential of large language models to interact with temporal and structured knowledge, the value of human expertise in the validation and refinement of reasoning processes, and the importance of FAIR and community-driven infrastructures for sustaining shared knowledge resources. Together, these perspectives outline a vision for future research in which Knowledge Graphs serve as reliable, explainable, and continuously evolving foundations for hybrid artificial intelligence.

Chapter 2

Fundamentals

In this chapter, we introduce the fundamental notions that underpin the contributions of this dissertation, following an organization consistent with the subsequent chapters. We begin by defining Knowledge Graphs (KGs), distinguishing between their schema and data layers, and outlining the role of ontologies in providing semantic structure. This foundation is necessary for understanding how knowledge can be consistently represented and manipulated across heterogeneous sources.

We then extend this discussion to Temporal Knowledge Graphs (TKGs), where facts are annotated with temporal information indicating when they hold true. Temporal representation is essential for the contributions of the first part of the thesis, which address temporal constraints and benchmarks for evaluating temporal reasoning.

Next, we review deep learning approaches for Knowledge Graphs, focusing on embedding models that project symbolic facts into continuous spaces. These methods illustrate how statistical learning can be leveraged for tasks such as completion and alignment, and they provide the technical background for the temporal benchmark presented in the second part of the thesis.

Finally, we turn to the challenge of cross-graph linking. We first discuss entity identification using IRIs and the owl:sameAs predicate, then move to ontology alignment and key-based identification. These mechanisms establish the necessary background for the latter part of the thesis, which develops methods for key transfer and proposes KG Nexus as a resource to support large-scale interoperability.

The chapter concludes with a synthesis that highlights the two central challenges guiding this dissertation: the integration of temporality into knowledge representation, and the alignment of heterogeneous graphs through scalable and explainable mechanisms.

2.1 . Knowledge Graph

Before presenting the more advanced notions used throughout this dissertation, we first establish a precise definition of what is meant by a *Knowledge Graph*. The following definition formalizes its two fundamental components, the ontology as defined by OWL (Web Ontology Language)[1] and the RDF data graph[2], which will be examined in detail in the next subsections.

Definition 1 (RDF Knowledge Graph) *A Knowledge Graph is defined as a pair* $(\mathcal{O}, \mathcal{G})$*, where:*

- $\mathcal{O} = (\mathcal{C}, \mathcal{P})$ *is an ontology represented in OWL, consisting of:*
  - *a set of classes* $\mathcal{C}$*;*
  - *a set of properties* $\mathcal{P}$*, which may be either* `owl:ObjectProperty` *(whose domain and range are classes) or* `owl:DatatypeProperty` *(whose domain is a class and whose range is a datatype such as* `xsd:date`*,* `xsd:string`*, or* `xsd:integer`*);*
  - *optionally, a set of logical axioms and constraints (e.g., subclass relations, disjointness, cardinalities) that enrich the schema with semantics and support automated reasoning.*
- $\mathcal{G}$ *is a finite set of RDF triples, each representing a statement of the form* $\langle subject,\ property,\ object \rangle$ *that instantiate the ontology* $\mathcal{O}$ *by describing concrete entities and their relationships, thus forming the RDF data graph.*

This formalization distinguishes between two conceptual layers: the schema layer*, encoded by the ontology* $\mathcal{O}$*, which defines the vocabulary and structural rules, and the* data layer*, represented by the RDF graph* $\mathcal{G}$*, which contains factual assertions about entities. This separation is fundamental for ensuring semantic interoperability and will be detailed in the following subsections.*

2.1.1 . Ontology Layer

Ontologies form the backbone of the schema layer in Knowledge Graphs, as they provide the semantic structure and logical rules that govern how knowledge is modeled and interpreted. Unlike simple vocabularies, which mainly list reusable terms, ontologies enable richer reasoning by introducing formal constraints and relationships.

Definition 2 (Ontology) *An ontology is a formal and structured specification of a conceptualization, defined in terms of:*

- *classes and properties (as in lightweight vocabularies);*

[1] https://www.w3.org/2001/sw/wiki/OWL
[2] https://www.w3.org/TR/rdf12-concepts/

- *logical axioms (e.g., disjointness, transitivity, symmetry);*
- *restrictions and constraints (e.g., domain/range, cardinality);*
- *class hierarchies and property hierarchies;*
- *support for automated reasoning.*

Ontologies are typically expressed in OWL, which allows advanced reasoning tasks such as classification, consistency checking, and inference over subclass relations.

Example. Consider a simple ontology with the classes `Person` and `City`, and the property `birthPlace`. We may specify that:

$$\langle birthPlace,\ rdf{:}type,\ owl:ObjectProperty\rangle$$
$$\langle birthPlace,\ rdfs{:}domain,\ Person\rangle$$
$$\langle birthPlace,\ rdfs{:}range,\ City\rangle$$

This ontology expresses the rule that the subject of a `birthPlace` relation must be a person, and its object must be a city.

2.1.2 . RDF Data Graph

While the ontology provides the schema layer of a Knowledge Graph, the *data layer* is instantiated as an RDF graph containing factual assertions about entities.

Definition 3 (RDF Data Graph) *An RDF data graph $\mathcal{G}$ is a finite set of triples of the form $\langle s,\ p,\ o\rangle$, where:*

- *$s \in \mathcal{I}$ is the subject, identified by an Internationalized Resource Identifier (IRI);*
- *$p \in \mathcal{P}$ is a property (predicate);*
- *$o \in \mathcal{I} \cup \mathcal{L}$ is the object, which may be an IRI or a literal value.*

Here, $\mathcal{I}$ denotes the set of all IRIs and $\mathcal{L}$ the set of literals, such as strings, dates, or numbers. IRIs generalize URIs by supporting the full Unicode character set, which ensures global and unambiguous identification of resources across datasets.

Example. Using the ontology introduced above, an RDF data graph may contain the following triples:

$$\langle Alice,\ birthPlace,\ Paris\rangle$$
$$\langle Paris,\ locatedIn,\ France\rangle$$
$$\langle Alice,\ birthDate,\ \text{"}1990-05-12\text{"}\hat{}\hat{}xsd:date\rangle$$
$$\langle Rex,\ isPetOf,\ Alice\rangle$$

Here, the triples assert that Alice was born in Paris on 12 May 1990 and owns a pet named `:Rex` while Paris is located in France.

To move beyond individual triples, it is often useful to describe the *local context* of an entity within the graph. This motivates the following definition.

Definition 4 (RDF Description of an Entity) *Given an entity $i \in \mathcal{I}$ in an RDF graph $\mathcal{G}$, its RDF description is the set:*

$$D(i) = \{(p, v) \mid \langle i, p, v\rangle \in \mathcal{G} \text{ or } \langle v, p, i\rangle \in \mathcal{G}\}.$$

This includes both outgoing and incoming object property statements involving i. We define:

- $P(i)$ *as the set of properties p for which there exists a value v such that $(p, v) \in D(i)$; for instance, in the example above:*

$$P(Alice) = \{birthPlace, birthDate, isPetOf\}$$

- $V(i, p)$ *as the set of values v associated with property p in $D(i)$; for example:*

$$V(Alice, birthPlace) = \{Paris,\ Rex\}$$

This entity-centered formalism is particularly useful for tasks such as entity alignment, similarity computation, and fact validation. It will be reused throughout this thesis as a basis for defining similarity functions, temporal constraints, and machine learning features.

2.1.3 . Interaction Between Schema and Data Layers

The true power of Knowledge Graphs emerges from the interaction between the schema layer (ontology) and the data layer (RDF graph). The ontology specifies the classes, properties, and logical constraints, while the RDF graph provides concrete instances of these concepts. By combining them, it becomes possible to infer new facts, validate existing ones, and enrich the graph with additional information.

Example. Using the ontology and RDF graph defined above, we know that the property $birthPlace$ has range $City$. From the triple $(Alice,\ birthPlace,\ Paris)$, we can automatically infer that $Paris$ is an instance of $City$, even if this triple is not explicitly present in the RDF graph. Similarly, if the ontology specifies that every $Person$ is a subclass of $Mammal$, then from the assertion $\langle Alice,\ rdf{:}type,\ Person\rangle$ we can deduce that $Alice$ is also a $Mammal$. On the other hand, from our very simple ontology we can not infer any implicit information for the entity Rex.

These examples illustrate how the combination of schema-level knowledge and data-level assertions supports automated reasoning. This interaction increases the amount of usable information, reduces redundancy, and improves semantic interoperability across heterogeneous sources. It is precisely this interplay that makes Knowledge Graphs a versatile framework for both symbolic reasoning and data-driven learning.

2.2 . Temporal Knowledge Graphs

Knowledge in the real world evolves continuously, which implies that the validity of many facts is inherently bounded in time. In application domains such as public administration, transportation infrastructures, and encyclopedic knowledge bases, roles change, regulations are updated, and factual descriptions must be interpreted with respect to their period of validity. It therefore becomes necessary to represent not only whether a statement is true, but also when it holds, in order to enable correct interpretation and reasoning over evolving data. Traditional Knowledge Graphs, however, represent facts as immutable triples, which makes them ill suited to capture such temporal evolution and may result in outdated or misleading inferences. To address this limitation, the notion of *Temporal Knowledge Graphs* has been introduced, extending factual statements with explicit temporal validity information. In this section, we provide the foundations necessary to understand Temporal Knowledge Graphs. We first describe how static triples can be augmented with temporal annotations to represent time dependent facts. We then introduce formal models for expressing temporal relations, with a particular focus on Allen's interval algebra. Finally, we discuss the main challenges and opportunities that arise when integrating temporal information into Knowledge Graphs, thereby motivating the temporal validation and reasoning approaches developed in the subsequent chapters.

2.2.1 . From Static to Temporal Knowledge Representation

Having introduced the motivation for Temporal Knowledge Graphs, we now examine how they extend the classical triple-based model. In a static KG, facts are represented as $\langle s,\ p,\ o\rangle$ and are implicitly assumed to hold universally, without reference to time. This assumption is often too restrictive, as many real-world facts are only valid during specific periods. Temporal Knowledge Graphs overcome this limitation by augmenting each triple with a temporal component, thereby turning it into a quadruple $\langle s,\ p,\ o,\ \tau\rangle$. This simple yet fundamental extension allows KGs to represent evolving states of the world in a time-aware manner.

Definition 5 (Temporal Knowledge Graph) *A **Temporal Knowledge Graph** is an extension of a standard Knowledge Graph in which each triple is augmented with temporal information. Formally, a Temporal Knowledge Graph is a pair $(\mathcal{O}, \mathcal{G})$, where:*

- *$\mathcal{O} = (\mathcal{C}, \mathcal{P})$ is an ontology represented in OWL, consisting of:*
  - *a set of classes $\mathcal{C}$;*
  - *a set of properties $\mathcal{P}$;*
- *$\mathcal{G}$ is a finite set of **RDF quadruples** that instantiate the ontology $\mathcal{O}$ by describing concrete entities, their relationships, and their temporal scope:*

$$\langle s,\ p,\ o,\ \tau\rangle$$

where s is an entity, p is a relation, o is either a literal or an entity and τ is a temporal expression indicating the time at which the fact holds. Depending on the temporal model used, τ can be:

- *a **point in time**, representing a single time point.*

$$\langle Alice,\ met,\ Bob,\ 2025{-}06{-}12\rangle$$

- *a **temporal interval**, specifying a validity period, such that $\tau = [\tau.start, \tau.end]$ or abbreviated for the rest of this thesis $[\tau.s, \tau.e]$ with $\tau.s < \tau.e$.*

$$\langle Alice,\ livedIn,\ Paris,\ [1990, 2015]\rangle$$

Example. With this temporal extension, we can represent both punctual events, such as $Alice$ meeting Bob on a specific day, and long-term situations, such as $Alice$ living in $Paris$ between 1990 and 2015. Without the temporal component, a static graph would either omit this information or represent it incorrectly: stating only $\langle Alice,\ livedIn,\ Paris\rangle$ would wrongly suggest that $Alice$ still lives in $Paris$. The temporal dimension thus prevents outdated or misleading representations and enables reasoning over the history and evolution of entities.

In practice, Temporal Knowledge Graphs are often only partially temporal. Whether a fact requires temporal qualification depends on the context in which the graph is constructed. For instance, in a graph describing events of the recent human century, it is unnecessary to add temporal information to the statement that the highest summit in the world is Mount Everest. By contrast, in a geological dataset, the same information might be annotated with a temporal scope, since elevations and landforms can vary over different geological timescales. Moreover, even when temporal information is provided, its granularity may differ both across datasets and within a single graph. Some facts may be recorded at the level of a year (e.g., $\langle Paris,\ elected,\ "mostAttractiveCity",\ 2024\rangle$, others with precise timestamps (e.g., $\langle Alice,\ met,\ Bob,\ 2025\text{-}06\text{-}12\rangle$), and still others as broader intervals (e.g., $\langle Alice,\ livedIn,\ Paris,\ [1990, 2015]\rangle$). This heterogeneity mirrors the diversity of real-world information and creates additional challenges for reasoning, alignment, and integration across graphs.

This temporal annotation enables Knowledge Graphs to capture the evolution of facts and to reason over the changing state of the world. It also opens the door to tasks such as temporal consistency checking, forecasting, and alignment of facts across time.

2.2.2 . Technical Implementation of Temporal Context

Based upon the survey [Wu et al.(2024)], we will now review the most common technical implementation of temporal context in real-world Knowledge Graphs. Different communities have adopted distinct modeling practices, which vary in their reliance on ontologies, qualifiers, or reification strategies. We focus here on four representative approaches: the Time Ontology in OWL with reification, named graph, Wikidata's statement-based model, and the use of RDF-star as in YAGO.

Time Ontology in OWL - Reification

A standard way to represent temporal elements in the Semantic Web is the Time Ontology in OWL[3] [Hobbs and Pan(2004)]. This ontology provides classes and properties to describe instants, intervals, and calendrical elements. To represent a temporally qualified fact, it is typically necessary to introduce an intermediate event entity. Using the example from the previous section, Listing 1 illustrates how an event in which `:Alice` and `:Bob` participate can be associated with a temporal description.

```
@prefix time: <https://www.w3.org/TR/owl-time/>.
@prefix greg: <https://www.w3.org/TR/owl-time/#greg:>.

:Alice :participateIn :Event.
:Bob :participateIn :Event.

:Event a time:Instant;
    time:inDateTime     :EventTime;
    time:inXSDDateTimeStamp     "2025-06-12"^^xsd:date.

:EventTime a time:DateTimeDescription ;
  time:unitType     time:unitDay ;
  time:day          "---12"^^xsd:gDay ;
  time:dayOfWeek    time:Friday ;
  time:dayOfYear    157 ;
  time:week         23 ;
  time:month        "--06"^^xsd:gMonth ;
  time:monthOfYear  greg:June ;
  time:timeZone     <https://www.timeanddate.com/time/zones/gmt> ;
  time:year         "2025"^^xsd:gYear .
```

Listing 1: Example of usage of the Time ontology

This model offers two levels of granularity: either a direct timestamp using $time{:}inSXDDateTimeStamp$, or a more detailed breakdown using a $time{:}DateTimeDescription$. It also supports the association of multiple entities with the same event. However, despite its expressiveness, this ontology is not widely adopted in large-scale general KGs, partly due to its verbosity and modeling overhead.

Named Graphs

Another classical approach to represent temporal context is through the use of *Named Graphs*, a feature already present in the RDF specification. In the work of Tappolet and

[3] www.w3.org/TR/owl-time/

Bernstein [Tappolet and Bernstein(2009)], each fact or triple f is placed inside a graph g that contains only the set of facts f' sharing the same temporal context (i.e., the same time point or temporal interval). The graph g is then associated with its temporal scope, which can be queried using standard SPARQL queries.

Although this approach is rarely adopted in large, general-purpose Knowledge Graphs, it finds close analogues in datasets designed for Temporal Knowledge Graph learning. For instance, the ICEWS dataset [García-Durán et al.(2018)] is represented in a TSV format with quadruples $\langle s, p, o, \tau \rangle$, where the fourth column τ can be interpreted as denoting a temporal named graph anchored at the corresponding time point. This style of representation is also commonly described as a *snapshot-based* approach, in which multiple subgraphs capture facts that appear or disappear across different temporal contexts.

Such representations are particularly well-suited to Deep Learning approaches, as they provide discrete time slices that can be readily used for temporal prediction tasks. We will return to this perspective in Chapter 4, where we discuss benchmarks and datasets for learning over Temporal Knowledge Graphs.

Wikidata through statements

One of the largest general-purpose Knowledge Graphs, Wikidata [Vrandečić and Krötzsch(2014)], follows a different design. Instead of reifying every temporal statement as an event, Wikidata preserves the direct link between subject and object using the wdt namespace, while also creating a reified statement entity in the p namespace. This statement acts as a container for qualifiers, which provide contextual information about the triple. For temporal data, qualifiers such as $P580$ (start time), $P582$ (end time), and $P585$ (point in time) are commonly used under the pq namespace. Other qualifiers can capture additional context, such as the source or precision of the information. The original value can be retrieved through the ps namespace. An illustrative example of this modeling pattern is shown in Figure 2.1.

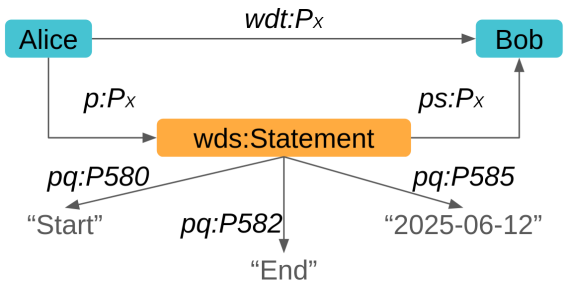


Figure 2.1: Wikidata temporal representation

This design strikes a balance: the graph can be queried in a straightforward way as if it were non-contextualized, while still allowing richer information to be accessed by navigating through the statement entity. Its main limitation is the lack of strict ontological semantics, as qualifiers are modeled in an ad-hoc but practical fashion.

Yago 4.5 with RDF-star

Another general-purpose KG, YAGO [Suchanek et al.(2024)], has adopted yet another representation strategy based on RDF-star[4]. This format allows RDF triples to be used as subjects of other triples, thereby enabling a concise reification mechanism. Temporal information can then be attached directly to a triple using properties such as $schema{:}startDate$ and $schema{:}endDate$:

$$\begin{array}{llll} \langle\langle\ \langle s,\ p,\ o\rangle, & schema{:}startDate, & \tau.s\rangle \\ \langle\langle\ \langle s,\ p,\ o\rangle, & schema{:}endDate, & \tau.e\rangle \end{array}$$

This representation avoids the need for additional event entities or verbose qualifier structures, while remaining closer to the RDF data model. Its limitation, however, is that RDF-star is still an emerging standard, and tool support to query the information remains incomplete compared to classical RDF.

Synthesis

In summary, these four approaches illustrate the spectrum of strategies available for modeling temporal context in Knowledge Graphs. The *Time Ontology in OWL* provides the most expressive and fine-grained representation, but its verbosity and modeling overhead limit its adoption in large-scale graphs. *Named Graphs* offer a conceptually simple and standards-compliant solution by grouping facts that share the same temporal scope, and they are closely related to the snapshot-based representations commonly used in temporal learning benchmarks; however, they are rarely employed in mainstream general-purpose KGs. *Wikidata's statement-based model* represents a pragmatic compromise: it retains the usability of classical triple queries while allowing qualifiers to enrich facts with temporal and other contextual information, albeit at the cost of formal ontological rigor. Finally, *RDF-star*, as used in YAGO, offers a lightweight and elegant mechanism to annotate triples directly, but its adoption is still limited due to the relative novelty of the standard and incomplete tool support.

Taken together, these approaches highlight a central trade-off between expressiveness and scalability in temporal modeling. As a result, no single approach has yet emerged as a universal standard, and current implementations remain fragmented across communities. This heterogeneity is one of the motivations for the contributions presented in this dissertation, where temporal reasoning must be developed in a way that is robust to multiple underlying representations.

2.2.3 . Representation of Temporal Relations

A key advantage of Temporal Knowledge Graphs is their ability to reason not only about individual facts, but also about the temporal relationships between them. For this purpose, *Allen's interval algebra* [Allen(1983)] provides a well-established framework for expressing qualitative relations between time intervals. Each interval is defined by its

[4] https://www.w3.org/2021/12/rdf-star.html

start and end points, and the algebra enumerates all the possible configurations of how two intervals may relate to one another.

Definition 6 (Allen's Interval Algebra) *Allen's interval algebra defines thirteen atomic relations that can hold between two temporal intervals. With the exception of* equals*, which is symmetric and thus its own inverse, each relation has a corresponding inverse (for example,* before *is the inverse of* after*). The complete set of relations is illustrated in Figure 2.2.*

These relations are distinct*, meaning that at most one relation can apply to any given pair of intervals;* exhaustive*, in that every pair of intervals must satisfy exactly one relation; and* qualitative*, since they capture relative ordering without reference to numeric duration.*

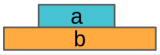
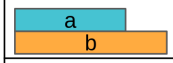
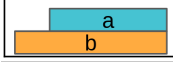
	a ***before*** b	b ***after*** a
	a ***equals*** b	b ***equals*** a
	a ***meets*** b	b ***is met by*** a
	a ***overlaps*** b	b ***is overlapped by*** a
	a ***during*** b	b ***contains*** a
	a ***starts*** b	b ***is started by*** a
	a ***finishes*** b	b ***is finished by*** a

Figure 2.2: Allen's interval algebra: seven distinct, exhaustive, and qualitative temporal relations.

Example. Consider two intervals representing Alice's residence in Paris $(1990, 2015)$ and her residence in London $(2015, 2020)$. The first interval *meets* the second, since it ends exactly at the point where the other begins. Capturing such relations enables temporal reasoning tasks such as consistency checking (e.g., detecting impossible overlaps) or the alignment of facts across different datasets. These possibilities will be further investigated in Chapter 3.

While Allen's original algebra focuses on relations between two intervals, many real-world facts also involve time points information, such as specific dates or events. To address this, this algebra has been extended to incorporate relations between intervals and time points [Ligozat(1991)].

Definition 7 (Extended Allen's Algebra) *This Extended Allen's algebra defines 5 supplementary relations between a time point and a temporal interval. These 5 relations are oriented with respect to the position of the time point yielding 10 unique relations. This orientation is shown through the usage of a dot* $[.]$ *to indicate where the temporal point is located. For instance if* a *is a time point and* b *a temporal interval then for any relations we would have* $a \ .R \ b$ *and* $b \ .R \ a$*. A complete list of examples are provided in Figure 2.3.*

Like the original algebra, the extension remains distinct*, exhaustive, and* qualitative*.*

	a ***.before*** b	b ***after.*** a
	a ***.begins*** b	b ***begun-by.*** a
	a ***.during*** b	b ***contains.*** a
	a ***.ends*** b	b ***ended-by.*** a
	a ***.after*** b	b ***before.*** a

Figure 2.3: Extended Allen's Algebra

Beyond the original or the extend algebra, a trivial algebra can also be defined for pairs of time points using simple operators: $<$ (*before*), $>$ (*after*), and $=$ (*equals*).

Building on the previous subsection 2.2.2, it is worth noting that the Time Ontology in OWL provides properties for several of Allen's original relations, such as $time : interval\ Before$. It also supports the point-based relations *before* and *after*, although no direct operator for *equals* is provided. By contrast, no standard OWL-Time properties exist for the extended point–interval algebra, leaving its representation dependent on ad-hoc modeling.

2.2.4 . Challenges and Opportunities in Temporal Knowledge Graphs

The introduction of temporal information into Knowledge Graphs brings both **challenges** and **opportunities**.

On the challenges side, facts may overlap or even conflict in their temporal scopes, making consistency checking and inference considerably more complex. Temporal data is also often sparse, irregular, or uncertain, particularly when extracted automatically from heterogeneous sources. For example, the birth of an individual may be recorded as a precise timestamp in one dataset but only as a year in another, introducing heterogeneity in temporal granularity. In addition, there is no consensus on how temporal context should be represented: different Knowledge Graphs adopt different formats, ranging from OWL-Time ontologies to qualifier-based models or RDF-star reification. This lack of standardization requires both users and tools to adapt to multiple modeling conventions, complicating interoperability and system design. More generally, representing changing entities, evolving relationships, and transient states requires rethinking not only storage models but also reasoning and learning strategies that can accommodate incomplete or noisy temporal evidence.

At the same time, Temporal Knowledge Graphs provide substantial **advantages**. They enable a faithful and fine-grained representation of reality by explicitly anchoring facts in time. They support advanced reasoning tasks such as temporal consistency checking,

event forecasting, and chronology-based inference. They also improve cross-graph **alignment**, since facts that occur during incompatible temporal intervals can be excluded as candidates for equivalence. Finally, they enhance **knowledge representation** in domains where change is intrinsic, such as history, medicine, or news.

These dual aspects explain why the design of temporal models, benchmarks, and reasoning frameworks has become a central focus in recent KG research. Addressing the challenges while leveraging the opportunities is crucial for building robust, context-aware applications. This motivation underpins the contributions of this dissertation, in particular the development of temporal constraints (Chapter 3) and benchmark resources (Chapter 4).

2.3 . Static Embedded Representations

Having examined how temporal information can be represented in Knowledge Graphs, we now turn to a complementary line of research that has become central in recent years: embedding-based representations. While the symbolic formalism of Knowledge Graphs, as defined in Definition 1, provides a human-readable and logically precise structure, it is not directly compatible with machine learning algorithms. Symbolic triples, composed of discrete identifiers and logical constructs, lack the differentiable properties required for optimization and learning in neural models. To overcome this limitation, Knowledge Graph Embedding (KGE) methods, as described in the survey [Dai et al.(2020)], aim to map entities and relations into continuous vector spaces, preserving as much semantic information as possible while enabling inference, generalization, and scalability. In this section, we present the foundations of KGE in the static context, before later extending the discussion to temporal embeddings.

Definition 8 (Knowledge Graph Embedding) *Given a Knowledge Graph $\mathcal{G}$ composed of triples $\langle s,\ p,\ o\rangle$, a Knowledge Graph Embedding is a mapping of entities $s,\ o \in \mathcal{I}$ and relations $p \in \mathcal{P}$ into an embedded representation that can take the form of a vector in the continuous space $\mathbb{R}^d$ but also in different spaces such as the complex $\mathbb{C}^d$. This embedded representation has to capture the structural and semantic relationship of the original graph. The embedding is typically learned by optimizing a scoring function that distinguishes valid triples from corrupted ones.*

2.3.1 . A simple translational model – TransE

A foundational approach in KGE is **TransE** [Bordes et al.(2013)], which introduces a translational model in vector space. In this geometric representation, each triple $\langle s,\ p,\ o\rangle$ is modeled such that the embedding vectors satisfy:

$$\vec{s} + \vec{p} \approx \vec{o}.$$

This captures the semantics of a relation as a translation from the subject to the object. For instance, taking the example in Figure 2.4, the model is trained to minimize the distance between $\overrightarrow{Alice} + \overrightarrow{bornIn} \approx \overrightarrow{Paris}$ for a true fact, while maximizing the distance for

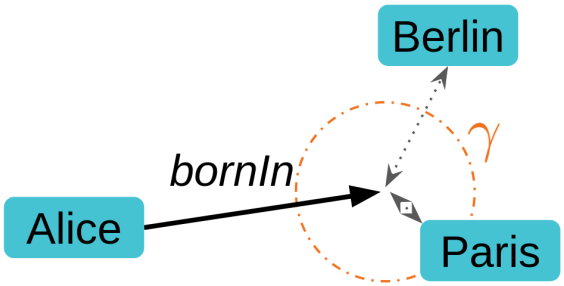


Figure 2.4: Example of training samples for TransE.

a false triple such as $\overrightarrow{Alice} + \overrightarrow{bornIn} \not\approx \overrightarrow{Berlin}$. The simplicity and efficiency of this idea made TransE a landmark model in the field.

Training a Knowledge Graph Embedding Model

Building on Definition 8, TransE and related models are trained using a **margin-based ranking loss**, which enforces that valid triples are scored higher than negative ones. A margin parameter γ (illustrated in orange in Figure 2.4) specifies the minimum distance by which negative samples must be separated from the end point of the translation.

Definition 9 (Margin-based ranking loss) *Given a positive triple $\langle s,\ p,\ o\rangle$ within the graph G and a negative triple $\langle s',\ p,\ o'\rangle$ where one component is corrupted, defined in G', the loss function is typically defined as:*

$$\mathcal{L} = \sum_{\langle s,p,o\rangle \in \mathcal{G}} \sum_{\langle s,p,o'\rangle \in \mathcal{G}'} \left[\gamma + f(s,p,o) - f(s',p,o')\right]_+ ,$$

where f is a scoring function, usually the L2 norm, γ is the margin hyperparameter, and $[x]_+ = \max(0,x)$ ensures non-negativity during optimization.

A key element of this function is the use of **negative examples** during training. However, Knowledge Graphs typically contain no explicit false facts, as storing them would be impractical. Furthermore, under the open-world assumption, the absence of a triple does not imply its falsehood. However in practice, the likelihood of a randomly sampled triple being true is close to zero, so KGE models usually adopt a *corruption strategy*, replacing one element of a true triple (often the object o) to generate a negative sample. Recent work focuses on improving negative sampling, either by selecting semantically close but incorrect entities (e.g., corrupting `Paris` into `Berlin`) or by ensuring that negative samples are not trivially valid in the KG.

2.3.2 . Beyond TransE: Capturing Complex Patterns

Although TransE is effective and intuitive, it struggles with complex relational structures such as one-to-many, many-to-one, and many-to-many relations. Consider the fact $\langle Alice,\ citizenOf,\ France\rangle$: Alice is one of millions of citizens of France, and the model must embed all of them close to each other. At the same time, these entities may participate in other relations, such as `bornIn`, which would require them to be embedded differently. This tension highlights the limitations of a purely translational model.

To address this, several extensions have been proposed:

- **TransH** [Wang et al.(2014)] introduces relation-specific hyperplanes to allow different entity projections per relation.
- **TransR** [Lin et al.(2015)] and **TransD** [Ji et al.(2015)] project entities and relations into separate or dynamically constructed vector spaces.
- **RotatE** [Sun et al.(2019)] represents relations as rotations in the complex space, capturing symmetry and inversion.

Other modeling paradigms have also been explored, including tensor-based approaches:

- **RESCAL** [Nickel et al.(2011)] and **DistMult** [Yang et al.(2014)], which rely on matrix factorization with different optimization strategies.
- **HolE** [Nickel et al.(2016)], based on circular correlation to increase expressivity.
- **ComplEx** [Trouillon et al.(2016)], which embeds entities and relations in the complex space to capture asymmetric relations while retaining the efficiency of DistMult.

Finally, graph-based approaches such as **Relational Graph Convolutional Networks (RGCN)** [Schlichtkrull et al.(2018a)] extend graph neural networks to handle multi-relational graphs. By propagating and aggregating information over neighborhoods, RGCNs enable relational reasoning that goes beyond the scope of simple triples.

2.3.3 . Applications: Knowledge Graph Completion and Entity Alignment

The main application of embedding models is **Knowledge Graph Completion (KGC)**, where the task is to predict missing facts by scoring candidate triples. For example, given an incomplete triple such as $\langle s,\ p,\ ?\rangle$, the model ranks potential object entities based on their vector compatibility, effectively suggesting likely completions.

In addition, embeddings are increasingly used for **entity alignment**. By embedding entities from multiple Knowledge Graphs into a shared or comparable space, similarities can be computed to identify alignment candidates.

2.3.4 . Towards Temporally Aware Embeddings

Building on these static models, we now consider how temporal context can be incorporated into embedding methods. Two recent surveys provide comprehensive overviews of this emerging field [Cai et al.(2022), Wang et al.(2023)]. A central observation is that many classical static models have been extended to handle the fourth component of a fact, turning the triple $\langle s,\ p,\ o\rangle$ into a quadruple $\langle s,\ p,\ o,\ \tau\rangle$.

Examples of Temporal Extensions

One of the earliest adaptations is *TTransE* [Leblay and Chekol([n. d.])], which extends the translational principle of TransE by incorporating temporal information into the scoring function. Other families of models include *hybrid approaches*, such as HyTE [Dasgupta et al.(2018)], which project entity and relation embeddings onto temporal hyperplanes, and *recurrent architectures*, such as TA-TransE [García-Durán et al.(2018)] or RE-Net [Jin et al.(2020)], which model sequences of temporal facts with recurrent or attention-based networks. More recent proposals explore *continuous-time models* (e.g., TNTComplEx [Lacroix et al.(2020)], TimePlex [Jain et al.(2020)]) that represent temporal validity directly in continuous vector spaces. Together, these approaches illustrate the diversity of strategies for integrating temporal information, from simple time-aware modifications of existing models to dedicated architectures that treat time as a first-class dimension.

Main Challenges

Despite these advances, several challenges remain. First, temporal information is often *sparse* and *heterogeneous*, making it difficult to train embeddings that generalize across different granularities (years, months, days, or timestamps). Second, most methods are tailored to specific benchmark datasets, which limits their transferability to real-world graphs where temporal annotations are incomplete or noisy. Third, incorporating time generally increases model complexity, raising concerns about scalability in very large graphs. Finally, these approaches remain predominantly statistical, leaving little room for explicit symbolic reasoning such as Allen's interval algebra.

Outlook

Taken together, temporal embeddings offer powerful tools for tasks such as temporal link prediction and event forecasting. Yet their limited interpretability and dependency on benchmark-specific assumptions highlight the need for hybrid approaches that combine the strengths of symbolic reasoning with the flexibility of embeddings. This perspective motivates the contributions developed later in this dissertation, where temporal validation is studied at the intersection of symbolic and statistical paradigms.

2.3.5 . Summary

In this section, we reviewed the foundations of Knowledge Graph Embeddings (KGE), from translational models such as TransE to more expressive extensions, and highlighted their applications in completion and entity alignment. We then outlined how these methods have been adapted to the temporal setting, giving rise to Temporal KGE (TKGE). Despite promising advances, TKGE models remain limited by sparse and heterogeneous temporal data, reliance on benchmark-specific assumptions, and limited interpretability.

Beyond these modeling issues, current benchmarks only address classical tasks such as link prediction or forecasting, leaving aside the novel challenge of **temporal validation** that we will introduce in Chapter 3. Yet, a shared resource is essential to compare competing approaches fairly and to foster progress on this task. Chapter 4 will therefore introduces a dedicated benchmark designed for temporal validation, filling this gap and providing the basis for systematic evaluation of future models.

2.4 . Linking Knowledge

Alongside temporal reasoning and embedding-based representations, a further central challenge for Knowledge Graphs is their *heterogeneity*. In practice, knowledge is distributed across multiple datasets, each with its own scope, modeling conventions, and identifiers. To obtain a coherent and interoperable view of the Web of Data, it is therefore necessary to establish reliable links across these heterogeneous sources.

At the heart of this problem lies the question of *entity identification*. As described earlier, each entity in a Knowledge Graph is uniquely denoted by an Internationalized Resource Identifier (IRI). While IRIs ensure unambiguous references within a single KG and facilitate efficient retrieval, they are not sufficient to recognize the same entity across different graphs. This limitation becomes particularly problematic in the context of the Linked Open Data (LOD) Cloud, where knowledge is fragmented across both specialized and general-purpose resources. For example, detailed biomedical data may reside in domain-specific databases, while general biographical information may be represented in Wikidata or DBpedia.

2.4.1 . Predefined Alignments

From the very beginning of the Semantic Web, the challenge of connecting distributed datasets has been recognized. Tim Berners-Lee's vision of the Web of Data emphasized openness and linkage, and the OWL ontology language therefore introduced the predicate `owl:sameAs` to assert that two identifiers denote the same real-world entity. Similar constructs have since been proposed in major vocabularies, such as `schema:sameAs` in Schema.org.

Although widely adopted, these predefined alignments come with important limitations. Their creation often relies on manual curation to ensure quality, which is not scalable to today's large Knowledge Graphs. Moreover, `owl:sameAs` links usually lack transparency or justification, providing no explanation of why two entities should be con-

sidered equivalent. Empirical studies [Hogan et al.(2012), Raad et al.(2018)] have also revealed that many such links are erroneous or only partially valid, leading to mismatches, ambiguities, and inconsistencies across datasets.

Beyond entities, the schema layer of a Knowledge Graph also requires alignment. While community guidelines encourage reusing existing vocabularies, in practice many new ontologies are created to represent domain-specific data. As a result, it becomes necessary to establish correspondences between classes and properties across different ontologies. This problem is made more complex when ontologies differ in their level of granularity or precision, requiring versioning and alignment mechanisms. Historically, such schema alignments were performed manually, but initiatives such as the Ontology Alignment Evaluation Initiative (OAEI)[5] have spurred the development of automatic matching frameworks. However, these alignments are typically not maintained or distributed as reusable resources after evaluation, limiting their practical impact.

2.4.2 . Linking of Entities

To overcome the limitations of predefined alignments, several frameworks have been proposed for entity linking, ranging from symbolic approaches to deep learning models. We begin here with symbolic methods based on the notion of keys.

Symbolic Alignments through Keys

In relational databases, *keys* have long been used to perform joins between tables. Since database schemas are explicitly designed with keys, unique identification is guaranteed by construction. By contrast, Knowledge Graphs were not originally designed with this constraint in mind. As a result, it is generally impossible to define a single universal set of properties that uniquely identifies entities across all graphs. Instead, keys in Knowledge Graphs are understood only in the context of a particular class.

Definition 10 (Key in a Knowledge Graph) *Let $\mathcal{G}$ be an RDF graph, and let E be the set of entities described in $\mathcal{G}$. A set of properties $\{p_1, p_2, \ldots, p_n\}$ is a **key** for a class C if, for any two instances $(e_1, e_2) \in E \times E$ of class C, the following holds:*

$$\forall i \in \{1, \ldots, n\},\ V(e_1, p_i) = V(e_2, p_i) \Rightarrow e_1 = e_2$$

In other words, if two instances share the same values for all properties in the key set, they are considered to represent the same real-world entity.

Building on this definition, frameworks such as SAKey [Symeonidou et al.(2014)] exemplify the use of keys for Knowledge Graphs. SAKey discovers candidate keys in an unsupervised manner by identifying minimal sets of properties that uniquely describe entities within a dataset. For example, the combination of `name`, `dateOfBirth`, `placeOfBirth`, and `hasParent` may serve as a unique identifier for individuals in genealogical datasets.

[5] https://oaei.ontologymatching.org/

Once discovered, such keys can be used not only for **intra-KG consistency checks**, but also for **inter-KG entity linking**. If two entities in different graphs share identical values for the properties comprising a key, they can be automatically aligned, with a structured justification based on their descriptions. This approach enables scalable and explainable linking of entities across large Knowledge Graphs, going beyond the black-box nature of statistical embeddings and the limited semantics of predefined `owl:sameAs` links.

Deep Learning Alignments through Embeddings

In contrast to symbolic methods, entity linking can also be tackled through embedding-based approaches as described in the survey [Sun et al.(2020)]. The core idea is to represent entities from multiple Knowledge Graphs in a shared or comparable vector space, such that equivalent entities are placed close to each other. This builds directly on the embedding techniques introduced in Section 2.3.

To achieve this, it is typically necessary to rely on a small set of *seed alignments*, which act as anchors between the two graphs. These seeds may come from manually curated correspondences, overlapping identifiers, or high-confidence matches. Using these anchors, a mapping function between the embedding spaces can be learned, either through linear transformations, adversarial training, or joint embedding of the graphs into a single space. Once the mapping is established, candidate alignments are identified by measuring the similarity of entity embeddings across the graphs.

Several frameworks exemplify this approach. MTransE [Chen et al.(2017)] extends the TransE model to learn transformations across multiple graphs. BootEA [Sun et al.(2018)] improves performance through bootstrapping, progressively expanding the set of alignments using high-confidence predictions. More recent methods leverage graph neural networks (GNNs), such as RREA [Mao et al.(2020)], which exploit structural neighborhoods to refine entity embeddings for alignment.

Embedding-based alignment offers several advantages. It scales to very large graphs, integrates seamlessly with downstream tasks such as completion, and does not require explicit modeling of symbolic constraints. However, it also suffers from limitations: it depends heavily on the availability and quality of seed alignments, it may struggle with heterogeneous modeling choices across graphs, and it generally lacks the transparency of symbolic methods such as keys.

In summary, deep learning approaches provide a powerful but less interpretable alternative to symbolic alignment. They highlight a central trade-off in entity linking: embeddings excel at scalability and adaptability, while symbolic methods offer more structured and explainable justifications for equivalence.

2.4.3 . Ontology Alignments

As we described in the previous section, symbolic or even hybrid approaches can heavily rely on the ontology description of the data to perform an entity alignment. Yet, apart from a few widely adopted vocabularies such as RDF(S)[6] or `schema.org`, ontologies are often created independently for specific domains. As a result, an alignment is required at the schema level before heterogeneous Knowledge Graphs can be effectively integrated. For example, one KG may describe individuals using $foaf{:}Person$ while another employs $schema{:}Person$; similarly, relations such as $birthPlace$ and $placeOfBirth$ may differ syntactically but denote the same concept.

Ontology alignment techniques address this problem by discovering **equivalence** or **subsumption mappings** between classes and properties in different schemas. These mappings are essential for **semantic interoperability**, enabling data from heterogeneous sources to be merged, queried, and reasoned over in a consistent way. As shown in the survey [Portisch et al.(2024)], a wide variety of alignment strategies exist:

- *Symbolic methods*, which exploit label similarity, lexical resources, or logical constraints. [Jiménez-Ruiz and Cuenca Grau(2011)]
- *Statistical approaches*, which rely on distributional similarity in instance data or usage patterns.
- *Neural and transformer-based models*, which leverage contextual embeddings to capture semantic similarity at the schema level.

The Ontology Alignment Evaluation Initiative (OAEI)[7] provides a long-standing forum for evaluating such methods. It organizes annual challenges across a range of tracks, including benchmarks for large biomedical ontologies (e.g., FMA–NCI–SNOMED), cross-lingual alignments, and knowledge graph tracks based on DBpedia and Wikidata. These evaluations have played a key role in advancing the state of the art and in highlighting the trade-offs between precision, recall, and scalability of different approaches.

However, as with instance-level linking, ontology-level alignments (e.g., `owl:equivalentClass`, `owl:equivalentProperty`) often suffer from noise, ambiguity, or modeling mismatches. Moreover, OAEI evaluations typically focus on one-off matching tasks rather than producing alignments that are maintained and reused as community resources. For this reason, in this thesis we explore complementary methods that combine ontology alignment with **key discovery** and **contextual validation**, in order to ensure robust linking at both the schema and data levels.

[6] RDFS: `https://www.w3.org/TR/rdf-schema/`
[7] `https://oaei.ontologymatching.org`

2.4.4 . Summary

In this section, we reviewed the central challenge of *linking knowledge* across heterogeneous Knowledge Graphs. At the entity level, predefined alignments such as `owl:sameAs` and `schema:sameAs` were introduced to assert equivalence, but they remain difficult to scale and often contain erroneous or ambiguous links. At the schema level, ontology alignments are required to reconcile differences in classes and properties, as exemplified by initiatives such as the OAEI, which has stimulated the development of a wide range of symbolic, statistical, and learning-based systems. Finally, we examined two complementary strategies for entity alignment: symbolic approaches based on keys, which provide explainable justifications for equivalence, and embedding-based methods, which offer scalability but at the cost of transparency.

Taken together, these approaches illustrate the trade-off between **scalability**, **accuracy**, and **explainability** in linking heterogeneous Knowledge Graphs. While substantial progress has been made, current methods often suffer from limited robustness and fragmentation across different communities. This motivates the contributions presented in Chapter 5, where we investigate whether keys can be transferred across heterogeneous Knowledge Graphs. We then describe in Chapter 6 the construction of a resource to store pre-computed ontology alignments and discovered keys, with the goal of accelerating and systematizing the process of linking knowledge.

2.5 . Synthesis

In summary, this chapter has reviewed the core foundations that underpin the contributions of this dissertation. We began with the definition of Knowledge Graphs, emphasizing the interplay between schema-level ontologies and data-level assertions. This dual perspective provides the necessary basis for reasoning, interoperability, and alignment across heterogeneous sources.

We then examined how temporal information can be represented in Knowledge Graphs. Several strategies were reviewed, ranging from highly expressive ontological models such as the Time Ontology in OWL, to pragmatic solutions such as Wikidata's statement model, RDF-star reification, and snapshot-based named graphs. Each approach embodies a trade-off between expressiveness, scalability, and tool support. This heterogeneity directly motivates the temporal reasoning methods and constraint discovery that will be introduced in Chapter 3, as well as the construction of a dedicated temporal benchmark in Chapter 4.

The chapter also surveyed embedding-based representations of Knowledge Graphs, showing how symbolic triples are mapped into continuous spaces for completion, alignment, and prediction tasks. While these models have been extended to handle temporal information, they remain limited by sparse and heterogeneous annotations, benchmark-specific assumptions, and a lack of interpretability. These observations provide the context for the benchmark resources of Chapter 4, which are designed to evaluate temporal validation more systematically.

Finally, we turned to the challenge of linking heterogeneous graphs. Predefined equivalence links and ontology alignments offer partial solutions, but often suffer from noise and lack of transparency. Symbolic approaches based on keys provide explainable justifications for entity equivalence, while embedding-based alignments offer scalability at the cost of interpretability. The tension between these strategies motivates the development of transferable keys (Chapter 5) and the creation of *KG Nexus*, a reusable resource that consolidates alignment-relevant knowledge (Chapter 6).

Taken together, these foundations highlight two central challenges addressed in this dissertation: integrating temporality into knowledge representation and reasoning, and ensuring robust and explainable linking across heterogeneous graphs. The following chapters tackle these challenges by proposing new methods and resources for temporal constraints, temporal validation benchmarks, transferable keys, and scalable alignment support.

Chapter 3

Explainable Temporal Fact Validation Through Constraints Discovery

*The work presented in this chapter has been in collaboration with **Fatiha Saïs, and Joe Raad**. Part of this work has been published in :*

SOULARD Thibaut, SAÏS Fatiha, and RAAD Joe.
Explainable Temporal Fact Validation Through Constraints Discovery in Knowledge Graphs,
in **ESWC 2025**.

SOULARD Thibaut, SAÏS Fatiha, and RAAD Joe.
Validation temporelle explicable de faits par la découverte de contraintes temporelles complexes dans les graphes de connaissances,
in **IC at PFIA 2024**.

Building upon the definitions provided in chapter 2, we present here one of the central research contributions of this thesis: the validation of temporal facts. This problem does not merely concern whether a fact appears in a knowledge graph, but rather whether it holds true within the time interval that is asserted. Addressing this challenge requires more than background definitions: it demands the discovery of patterns, the formulation of temporal constraints, and the development of methods that are both effective and explainable. The present chapter proposes a new method for temporal validation that leverages temporal constraints automatically extracted from the knowledge graph.

3.1 . Introduction

The spread of misinformation poses significant threats to political, economic, and scientific domains. With the increasing deployment of generative AI, the need for reliable mechanisms to validate or refute facts has become more pressing than ever. Knowledge Graphs (KGs) [Hogan et al.(2021)] offer a valuable means for fact verification by structuring information into machine-readable statements. They have become fundamental resources for AI applications, supporting tasks such as question answering [Saxena et al.(2020)], recommender systems [Liu and Duan(2021)], and image recognition [Chen et al.(2020)], while also providing rich semantic and contextual information [Peng et al.(2023)].

Yet, despite the large number of KGs published in recent years, only a few explicitly associate a temporal component with their time-dependent facts. This temporal dimension, present for example in Wikidata[1], which is one of the largest publicly available KGs, is essential for ensuring factual accuracy. Many statements are only valid within specific intervals or at precise points in time. For instance, the fact `(Obama, headOfState, USA)` is only true between `2009/01/20` and `2017/01/19`. With the growing use of KGs in fact-checking applications [Tchechmedjiev et al.(2019)], ensuring the temporal validity of such facts is a key challenge. However, the task of validating facts against their temporal constraints remains largely underexplored in current research.

In this chapter, we address the challenge of temporal fact validation by introducing an explainable framework that leverages automatically discovered temporal constraints. Knowledge Graphs (KGs), often constructed through automatic or semi-automatic processes, are frequently subject to incompleteness and errors, which complicates the task of assessing the veracity of temporal facts. Our approach assumes that the validity of a temporal fact depends on its coherence with the temporal history of the related entity. For instance, an individual cannot hold the position of surgeon without having previously graduated in medicine. Such dependencies can be expressed as temporal constraints between properties, such as `before(graduatedAsDoctor,isSurgeonIn)`. Since such constraints are rarely encoded in KGs, our method automatically discovers them by extending Allen's interval algebra [Allen(1983)] from intervals to sequences. Once discovered, these constraints can be used to check the temporal consistency of candidate facts with what is already known in the KG. They are then applied either symbolically, to provide explicit and interpretable justifications for validation or rejection, or as features integrated into machine learning models, thus combining explainability with scalability in large-scale scenarios.

[1] https://www.wikidata.org/

3.2 . Related Work

Fact validation, also known as veracity assessment or truth discovery, is a complex task that requires integrating and evaluating information from multiple sources while accounting for factors such as source reliability, majority consensus, and contextual cues [Aïmeur et al.(2023)]. In this section, we review the main families of AI-based frameworks designed for this task. We begin by characterizing the different types of input data that such frameworks must handle. We then examine the corresponding methodologies, ranging from purely textual approaches to structured Knowledge Graph (KG) reasoning. Finally, we highlight how temporality and temporal constraints have been introduced into these frameworks.

Textual Fact Validation

At the most unstructured level, facts are expressed as free natural language text that may contain multiple claims. For instance, the statement *"Macron, the president of Spain, has met the head of the European Union"* encodes at least two distinct claims: that Macron is president of Spain, and that he has met the EU head. In such settings, frameworks must operate solely on the textual surface form without prior disambiguation.

Early approaches relied on classical language models, while recent advances exploit Large Language Models (LLMs). For example, the QABriefer framework [Fan et al.(2020)] decomposes a claim into a series of questions such as *"Who is Macron?"*, *"Who is the President of Spain?"*, or *"What is the EU?"*. The answers to these questions, retrieved via search engines, are then aggregated into a summary brief that can be presented to the user to ease the burden of fact checking. More recent work adopts multi-agent LLM architectures, where different agents specialize in subtasks such as question generation, evidence retrieval, reasoning, and adjudication. In the framework of [Ma et al.(2025)], agents iteratively decompose, retrieve, and evaluate claims, while "judge" agents arbitrate disagreements until consensus is reached. These designs enable deeper reasoning and more robust evidence integration than single-pass systems, but they also raise challenges of scalability and reproducibility, as they depend heavily on retrieval quality and the opaque reasoning of LLMs.

Semi-structured Fact Validation

A more semi-structured level is exemplified by ClaimsKG [Tchechmedjiev et al.(2019)], where claims extracted from text are enriched with metadata such as the source, time, and identified entities (e.g., `:Macron`, `:Spain`). This additional context supports entity linking [Fernández et al.(2007)] and improves evidence retrieval. For example, disambiguating the mention *"head of the European Union"* to Ursula von der Leyen enables frameworks such as DeFacto [Gerber et al.(2015)] or FactKG [Kim et al.(2023)] to ground claims in external knowledge bases, thereby increasing reliability and interpretability. Nevertheless, such frameworks remain primarily focused on static claims and have limited ability to incorporate temporal dynamics.

Structured Fact Validation in Knowledge Graphs

In fully structured settings, facts are expressed as disambiguated triples, such as `(:Macron, headOfState, :Spain)` or `(:Macron, met, :headOfEU)`. Classical KG-based methods can directly exploit the graph structure. Flow- and path-based algorithms [Ciampaglia et al.(2015), Shiralkar et al.(2017), Syed et al.(2019)] evaluate the plausibility of a triple by analyzing supporting paths and relational constraints. The thesis of Valentina Beretta [Beretta(2018)] further demonstrates the effectiveness of ontology-based rules for validation. More recently, hybrid approaches such as KGValidator [Boylan et al.(2024)] combine structured and textual reasoning by verbalizing triples into natural language, embedding them, and then applying LLMs to integrate KG and textual evidence. These strategies illustrate how fact validation frameworks increasingly blur the line between unstructured, semi-structured, and structured settings, combining their strengths to improve both scalability and explainability. Yet, they typically ignore temporal validity, which is critical for many real-world facts.

Incorporating Temporal Information

Several approaches have introduced temporal signals into fact validation. DeFacto [Gerber et al.(2015)] and BiLSTM-based models [Allein et al.(2021)] account for time annotations in data, but primarily operate on static claims. In reality, atomic claims are rarely universally true; they depend on temporal intervals (e.g., `(Obama, headOfState, USA)`).

In the KG community, temporal reasoning has been primarily explored through Temporal Knowledge Graph Completion (TKGC) as described in 2.3.4. These models, which we detail further in Chapter 4, focus on *link prediction*, inferring missing facts by predicting relationships or entities in a temporal context. Techniques include translation-based embeddings, tensor decomposition, Graph Convolutional Networks (GCNs), and recurrent architectures such as LSTMs and GRUs [Wang et al.(2023), Cai et al.(2024)]. While effective for predicting links, these models do not validate whether existing facts are correct in their stated temporal contexts.

A related line of work, *temporal context prediction*, seeks to determine the temporal interval of a fact given its components [Ling et al.(2021), Islakoglu et al.(2024), Jain et al.(2020)]. These models move closer to reasoning about temporal information, but they still stop short of addressing *temporal fact validation*, which requires verifying the correctness of a fact against discovered temporal constraints.

Temporal Constraint Discovery

Our work explicitly tackles temporal fact validation by discovering and applying temporal constraints. The closest approach, TemporalFC [Qudus et al.(2023)], focuses on validating corrupted facts but is limited in several respects: it excludes literals, disregards semantic information from entity classes, and relies on timestamps rather than intervals, thereby missing real-world temporal variability.

More generally, *temporal constraint discovery* has begun to attract attention in the KG community, as illustrated by PaTeCon [Chen et al.(2023a)] and TLogic [Liu et al.(2022)]. In parallel, research in relational data profiling [Abedjan et al.(2019)] has explored ordering relations using simple comparison operators (e.g., $\leq, <$) [Consonni et al.(2019), Xiao et al.(2022)]. While valuable, these methods fail to capture the full range of interval-specific operators required for robust temporal reasoning. To the best of our knowledge, no existing approach achieves the expressiveness of our temporal constraints, nor do they integrate such constraints into an explainable and scalable framework for temporal fact validation.

Synthesis

Taken together, prior work has laid important foundations in fact validation, temporal KG completion, and data profiling. However, none provide a comprehensive and explainable framework for validating temporal facts. Our contribution is to fill this gap by automatically discovering temporal constraints from KGs and leveraging them for fact validation, thereby combining the interpretability of symbolic reasoning with the adaptability of machine learning.

3.3 . Background and Preliminaries

Our framework for temporal fact validation relies on Allen's interval algebra [Allen(1983)] to represent about relationships between time intervals. Since the formal definition of Allen's algebra has already been introduced in the subsection 6, we restrict ourselves here to the notations and refinements that are specific to our approach. In particular, we recall the definition of temporal knowledge graphs and explain how we adapt Allen's relations to facilitate constraint discovery when the temporal information is represented as sequences of intervals.

3.3.1 . Temporal Knowledge Graphs

We focus on temporal knowledge graphs (TKGs), i.e., knowledge graphs where facts are explicitly associated with temporal information. Formally, each fact is represented as a quadruple $\langle s, p, o, \tau \rangle$ as defined in Definition 5. However we reduce this definition to specific case where τ denotes a time interval, and $(I.s; I.e)$ represents its start and end date respectively. To ensure uniformity, we treat any timestamp τ as a degenerate interval $(I.s = t; I.e = t + \epsilon)$, with ϵ being an insignificant duration determined by the temporal granularity (e.g., centuries, years, days, minutes). Throughout the chapter, we will systematically treat temporal information in terms of intervals.

3.3.2 . Temporal Relations

Allen's interval algebra as we introduced in the section 2.2.3 defines thirteen atomic binary relations between intervals. For the purpose of temporal constraint discovery, we restrict this set to seven relations: *before*, *meets*, *equals*, *overlaps*, *during*, *starts*, and *finishes*. The remaining six relations are symmetric counterparts of these (e.g., *after* is the inverse of *before*), and thus do not add expressive power in our setting. This simplification reduces redundancy and improves computational efficiency without compromising the generality of the discovered constraints.

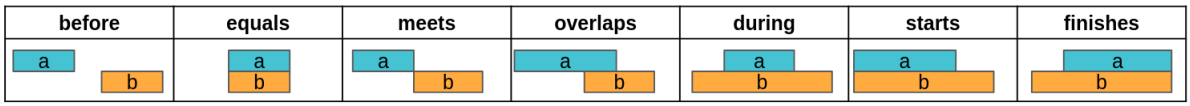


Figure 3.1: Examples of Allen's time interval relations

To enhance clarity, the selected relations are further organized into two main categories, according to whether the intervals under comparison overlap in time:

- **Disjoint Relations (DR):** representing non-overlapping temporal relationships, namely $DR = \{\texttt{before}, \texttt{meets}\}$.
- **Joint Relations (JR):** representing relationships where intervals overlap at least partially, specifically $JR = \{\texttt{equals}, \texttt{overlaps}, \texttt{during}, \texttt{starts}, \texttt{finishes}\}$.

Given two intervals I and I', we may either identify the exact Allen relation between them or use the generalized notation $DR(I, I')$ (resp. $JR(I, I')$) to indicate that a disjoint (resp. joint) relation holds. This generalization is particularly useful when constraints must be applied across heterogeneous entities, where the precise atomic relation may vary but the disjoint/intersecting distinction still captures the intended temporal dependency.

3.4 . Approach Overview

The core idea of our framework is to validate temporal facts by checking their consistency with temporal constraints automatically discovered from the temporal knowledge graph (**TKG**). Rather than assuming that temporal constraints are manually specified by domain experts, we extract them directly from the graph, generalize them across entity classes, and then use them to decide whether a new fact is valid in its asserted temporal context.

In Figure 3.2 we present the general pipeline of our approach of temporal fact validation. It operates in three main steps:

(1) Temporal Constraint Discovery. We begin by extracting temporal constraints from the temporal information associated with each entity. To this end, we consider all

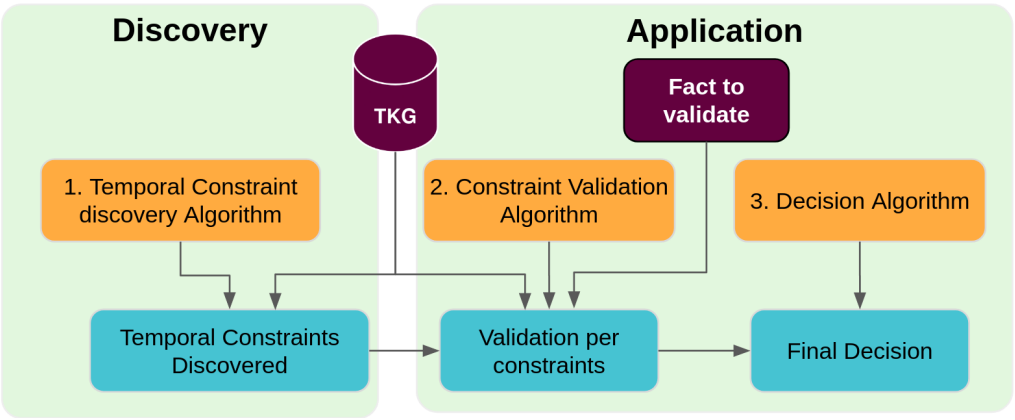


Figure 3.2: Approach workflow

possible pairs of time-dependent properties of the entity in order to derive constraints. These may be simple, relying on a single Allen relation, or more complex, combining multiple relations across sequences. The extracted constraints are then generalized across all entities of the same class by applying both error and generalization thresholds.

(2) Constraint-Based Validation. Given a candidate temporal fact, we verify whether its temporal information is consistent with the temporal constraints discovered in the KG. This is achieved by integrating the fact's temporal information into the timeline of the entity serving as its subject. The verification can be performed with respect to individual constraints or by considering multiple constraints simultaneously.

(3) Decision Strategy. To reach a final verdict on the validity of a fact, we combine the outcomes of individual constraints. We investigate two complementary strategies:

- a *symbolic strategy*, based on majority voting over constraints, which ensures full explainability by explicitly listing the rules involved in the decision;
- an *ML-based strategy*, where a classifier learns to weigh constraints differently depending on their predictive utility, thus improving adaptability to large-scale and noisy settings.

Our approach is designed to strike a balance between two often conflicting requirements: *transparency*, ensured by symbolic reasoning, and *robustness*, supported by machine learning. The following sections detail each step: temporal constraint discovery (Section 3.5), validation mechanisms (Section 3.6), and experimental evaluation (Section 3.7).

3.5 . Temporal Constraint Discovery from KGs

Our approach starts by discovering all possible temporal constraints for each entity e within a $\mathcal{TKG}$. These constraints concern pairs of time-dependent properties associated with e, whether e acts as a subject or an object. The discovered constraints can be either simple (involving a single temporal relation) or complex (combining multiple temporal relations). Initially, these constraints are discovered for a single entity, before being generalised across all entities within the same class. The following sections outline our approach for temporal constraint discovery. We begin by defining the concepts of entity time sequences (section 3.5.1) and the relevant intra- and inter-sequence comparisons (section 3.5.2). Next, we define the simple and complex temporal constraints that we consider and detail the method used for their discovery at the entity level (section 3.5.3). Finally, we describe how the discovered temporal constraints can be generalised to the broader class of entities (section 3.5.5).

3.5.1 . Entity Time Sequence

For each entity and every time-dependent property associated with that entity in the $\mathcal{TKG}$, we construct an entity time sequence containing the intervals during which the property holds for the entity (see Definition 11).

Definition 11 (Entity Time Sequence) *A time sequence of an entity x for a property p is the ordered set of intervals S of the quadruples $\{q_1, \ldots, q_n\}$, in the form of $\langle x,\ p,\ y_k,\ I_k \rangle$ or $\langle y_k,\ p,\ x,\ I_k \rangle$ with $k \in [1..n]$ and I_1 having the earliest start date and I_n having the last start date in the time sequence.*

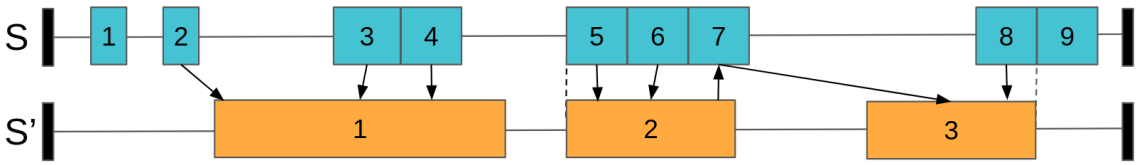


Figure 3.3: An example of two time-sequences S and S' containing 9 and 3 time intervals respectively. The arrows refer to inter-sequence comparisons.

Figure 3.3 presents the entity time sequences S and S' of two properties `PlaysIn` and `WorksAt` respectively for a single entity. Each element in the sequence represents a time interval. To be able to compare time intervals within the same entity time sequence and between different time sequences, we restrict in the following subsections our approach to properties that are temporally functional (see Definition 12). In the last subsection (subsection 3.5.6), we propose an extension that allows our approach to also take into account non-functional temporal properties.

Definition 12 (Temporally Functional Property) *A property p is temporally functional, if for every entity it does not exist a pair of overlapping intervals in its corresponding time sequence. We denote $\mathcal{FP}$ the set of temporally functional properties in $\mathcal{TKG}$. A property p is in $\mathcal{FP}$ iff:*

$$\forall x \in \mathcal{E},\ \forall y, y' \in \mathcal{E} \cup \mathcal{L},\ \forall I, I' \in \mathcal{T}, \langle x, p, y, I\rangle \wedge \langle x, p, y', I'\rangle \wedge DR(I, I')$$

For instance, `presidentOf` is temporally functional, since a person cannot be the `presidentOf` of two different countries in overlapping time intervals. For the remainder of this work, the temporal functionality of a property is inferred directly from its observed usage in the data, without relying on predefined ontological constraints or external domain knowledge.

3.5.2 . Entity Time Sequence Comparison

In our approach, we generate temporal sequences for all temporally functional properties associated with a given entity in the $\mathcal{TKG}$. The goal is to generate temporal constraints by comparing intervals across these sequences. To ensure the quality of the generated constraints and maintain computational efficiency, we focus on inter-sequence comparisons that are relevant, as presented below.

Definition 13 (Relevant inter-sequence comparisons) *Given an entity e and a pair of its time sequences S and S' corresponding to two temporally functional properties P and P', two intervals I from S and I' from S' are considered relevant for comparison if they share a sub-interval or if no other interval lies between them. This definition can be written under the following formula :*

$$\begin{array}{rl} & I \cap_t I' \neq \emptyset \\ \vee & \left(\begin{array}{ll} & I.s < I'.e \\ \wedge & (\nexists I'' \in S \setminus \{I\}, (I''.s \geq I.e \wedge I''.s \leq I'.s) \\ \wedge & (\nexists I'' \in S' \setminus \{I'\}, (I''.e \leq I'.e \wedge I''.e \geq I.s) \end{array}\right) \\ \vee & \left(\begin{array}{ll} & I.s < I'.e \\ \wedge & (\nexists I'' \in S \setminus \{I\}, (I''.e \geq I'.s \wedge I''.e \leq I.s) \\ \wedge & (\nexists I'' \in S' \setminus \{I'\}, (I''.s \leq I.s \wedge I''.s \geq I'.e) \end{array}\right) \end{array}$$

We denote by $\Omega(S, S')$ the set of relevant inter-comparisons between S and S'.

The results of these relevant inter-sequence comparisons are captured in a matrix $M_{\triangleright}$, representing the number of interval pairs across S and S' that satisfy a temporal relation r. The two rows are to account for non-symmetric relations (“before”, “during”, “starts” and "finishes"). In the example shown in Figure 3.3, these relevant inter-sequence comparisons are illustrated with arrows and summarised in the matrix $M_{\triangleright}$ in Table 3.1. For instance, the comparisons $\texttt{starts}(S.I_5, S'.I_2)$ and $\texttt{finishes}(S.I_8, S'.I_3)$ respectively correspond to the only "starts" and "finishes" relations in $M_{\triangleright}$ (represented with "1" in Table 3.1).

	before	equals	meets	overlaps	during	starts	finishes
o($S.I$, $S'.I'$)	2	0	0	1	3	1	1
o($S'.I'$, $S.I$)	0	0	0	1	0	0	0

Table 3.1: Inter-Comparison matrix $M_{\triangleright}$ for the sequences S and S' from Figure 3.3.

For discovering complex temporal constraints, intra-sequence comparisons are also needed. To improve efficiency, we also limit comparisons to relevant intra-sequence intervals, defined as follows:

Definition 14 (Relevant intra-sequence comparisons) *For a given time sequence S of an entity e and a property P, relevant intra-sequence comparisons consists of pairs of successive intervals.*

In Figure 3.3, the sequence S has four relevant intra-sequence comparisons with `meets(3,4)`, `meets(5,6)`, `meets(6,7)`, and `meets(8,9)`. On the other hand, S' does not have any relevant comparisons. These comparisons are captured in the matrix $M_{\triangleleft}$ 3.2. We only keep track of the "meets" relation as all non-"meets" comparisons default to the “before” relation.

	S	S'
meets	4	0

Table 3.2: Intra-Comparison matrix $M_{\triangleleft}$ for the sequences S and S' from Figure 3.3.

3.5.3 . Simple Temporal Constraint

For each entity e in a $\mathcal{TKG}$, and each pair of time sequences (S, S') corresponding to two properties (P, P') describing e, our approach first derives simple temporal constraints $r(e, P, P')$ based on the number of inter-comparisons in $M_{\triangleright}$. Specifically: $\forall I \in S$, $\exists I' \in S'$, s.t. r($S.I$, $S'.I'$) holds and $(I, I') \in \Omega(S, S')$.

We note that in our method the relation $equals$ is relaxed to a non-symmetric constraint termed $subsumes$. In the example in Table 3.1, no simple temporal relation can be derived for the sequences S and S', because no unique relation applies to every relevant pair of intervals. But for each possible simple constraint an example is provided on figure 3.4.

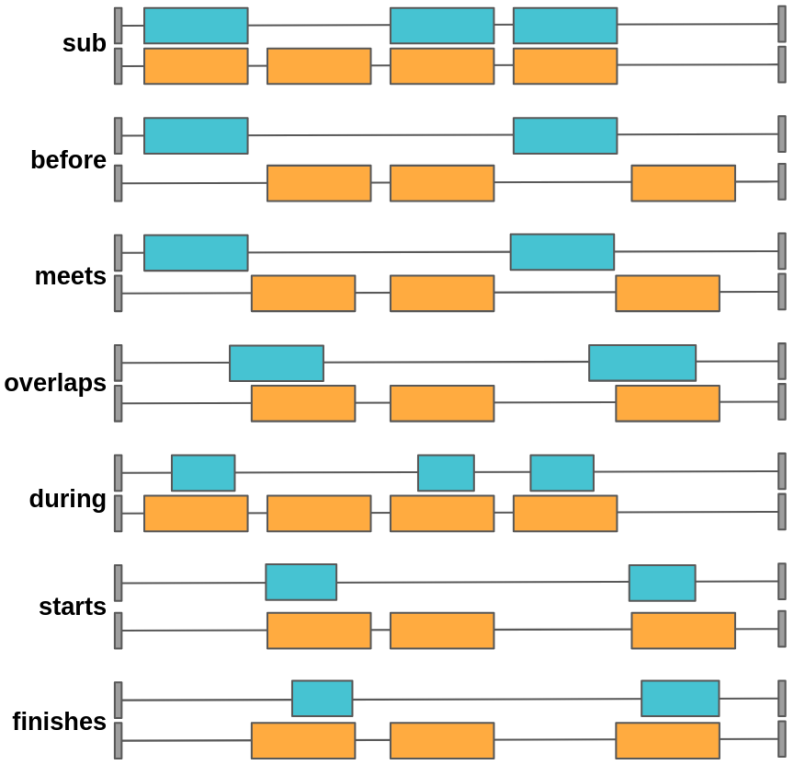


Figure 3.4: Simple temporal constraints

3.5.4 . Complex Temporal Constraints

To capture more nuanced dependencies, we define ten complex temporal constraints, which are organized hierarchically in the Complex Temporal Constraint Tree (CT^2) shown in Figure 3.5. The tree is designed to reflect a progression from rough temporal dependencies toward increasingly precise ones, where each level introduces more restrictive conditions on how intervals can relate to one another. For example, a general exclusion such as `NAND` only requires that no overlap occurs, while more specific constraints like `Sequence_Before` or `Closed_NAND` impose additional requirements on the ordering or contiguity of intervals. Importantly, constraints in CT^2 can be either symmetric or non-symmetric (e.g., `NAND` is symmetric, while `Sequence_Meets` is not), enabling the framework to capture both bidirectional exclusions and directional temporal orders. By structuring the constraints in this way, the framework makes explicit how broad temporal rules can be refined into more detailed and interpretable conditions, better aligning with the diversity of real-world temporal dependencies.

Joint Complex Temporal Constraints:

This category encompasses constraints where intervals from the two sequences share common sub-intervals. Illustrative examples are provided in Figure 3.7, and the corresponding formal definitions are given below.

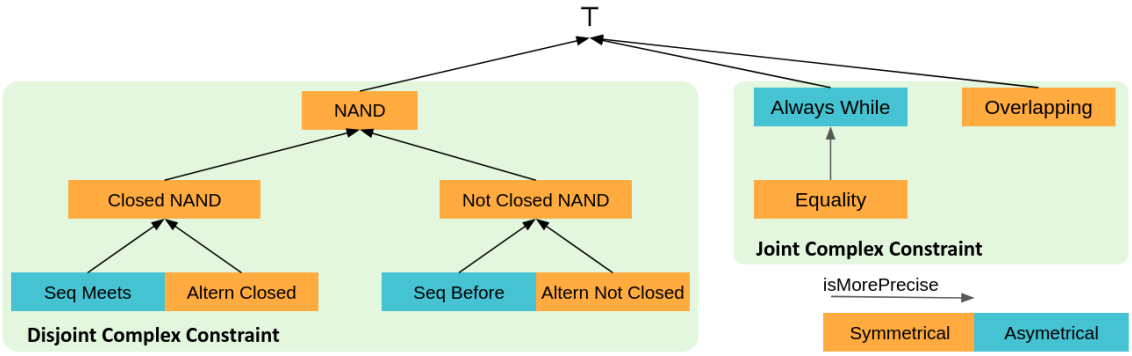


Figure 3.5: Complex Temporal Constraints Tree (CT^2)

- Overlapping This constraint captures situations where two properties of an entity must partially coincide in time, except for the last occurrence. For example, an actor may *workAt* a theater company while simultaneously *playsIn* a movie, with all but the final interval overlapping. Formally, for sequences S and S', every quadruple of S overlaps one in S', except the latest. Let us consider JR the set of intersecting relations (see subsection 3.3.2), the two time sequences S and S' of the properties P and P' respectively, and the matrix of inter-comparisons $M_{\triangleright}$ of S and S' fulfils the Overlapping constraint if:

$$M_{\triangleright}[overlaps][o(S,S')] + M_{\triangleright}[overlaps][o(S',S)] = |S| + |S'| - 1$$

- Always_while. This constraint enforces that every interval of one property is entirely contained within an interval of another. A typical example is a student whose *enrolledInCourse* intervals are always nested inside a larger *enrolledAtUniversity* interval. For sequences S and S', every quadruple of S must occur during one in S'. Formally, let us consider JR the set of intersecting relations (see subsection 3.3.2), the two time sequences S and S' of the properties P and P' respectively, and the matrix of inter-comparisons $M_{\triangleright}$ of S and S' fulfils the Always_while constraint if:

$$\sum_{r \in \{equals, during, starts, finishes\}} M_{\triangleright}[r][o(P,P')] = |S|$$

- Equality. This constraint requires that intervals match exactly between two properties. For instance, the mandate of a country's *headOfState* may coincide perfectly with their role as *commanderInChief*. In this case, the intervals of both properties must align one-to-one. More formally, the equality constraint is fulfilled if:

$$M_{\triangleright}[equals][o(S,S')] + M_{\triangleright}[equals][o(S',S)] = |S| + |S'|$$

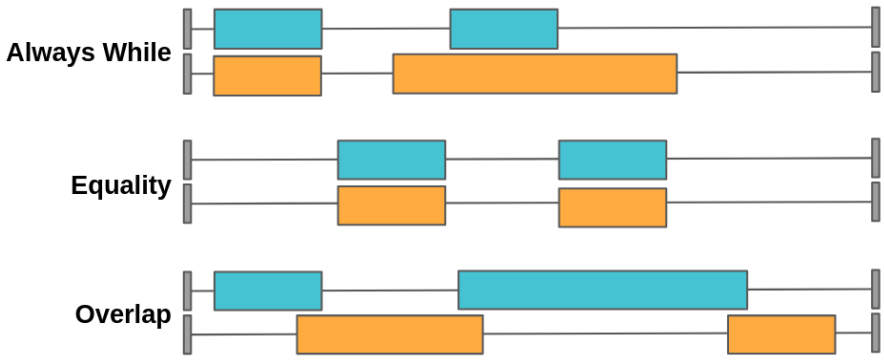


Figure 3.6: Joint Complex Constraints

Disjoint Complex Constraints:

Seven constraints in this category address cases with no interval overlap. Examples are given on the figure 3.6 and are defined as described below :

- NAND This constraint prohibits any temporal overlap between two properties. A clear example is a person who cannot be *presidentOf* two different countries at the same time, or a company that cannot be *headquarteredIn* two cities during overlapping years. For sequences S and S', the NAND constraint is fulfilled if:

$$\left(\sum_{r \in JR} M_{\triangleright}[r][o(S,S')] + M_{\triangleright}[r][o(S',S)] \right) = 0$$

- Closed_NAND. This stricter version of NAND additionally requires that intervals follow each other without gaps. For example, a CEO who steps down from one company and immediately takes a leadership role in another satisfies this condition. For sequences S and S', the Closed_NAND constraint is fulfilled if:

$$M_{\triangleright}[meets][o(S,S')] + M_{\triangleright}[meets][o(S',S)] + M_{\triangleleft}[meets][S] + M_{\triangleleft}[meets][S'] = |S| + |S'| - 1$$

- Not_Closed_NAND. In contrast, this constraint requires that gaps always occur between intervals. A politician finishing one *termInOffice* must spend some time out of office before starting another role. For sequences S and S', the Not_Closed_NAND constraint is fulfilled if:

$$M_{\triangleleft}[meets][S] + M_{\triangleleft}[meets][S'] = 0 \wedge \left(\sum_{r \in AA/\{before\}} M_{\triangleright}[r][o(S,S')] + M_{\triangleright}[r][o(S',S)] \right) = 0$$

- Alternation_Closed. This constraint expresses that intervals of two properties alternate without gaps. For example, an athlete alternating seasons between *playsForTeamA* and *playsForTeamB* would directly switch from one to the other. For sequences S and S', the Alternation_Closed constraint is fulfilled if:

$$M_{\triangleright}[meets][o(S,S')] + M_{\triangleright}[meets][o(S',S)] = |S| + |S'| - 1$$

$$\wedge \left(\sum_{r \in IA} M_{\triangleright}[r][o(S_1,S_2)] + M_{\triangleright}[r][o(S_2,S_1)] \right) = 0$$

- Alternation_Not_Closed. Here, alternation still occurs, but with gaps between intervals. A researcher alternating between *teachingAtUniversity* and *visitingResearcher* positions while taking sabbaticals in between illustrates this pattern. For sequences S and S', the Alternation_Not_Closed constraint is fulfilled if:

$$M_{\triangleright}[before][o(S,S')] + M_{\triangleright}[before][o(S',S)] = |S| + |S'| - 1$$

$$\wedge \sum_{r \in IA} (M_{\triangleright}[r][o(S_1,S_2)] + M_{\triangleright}[r][o(S_2,S_1)]) = 0$$

- Sequence_meets. This constraint requires that the last interval of one property ends exactly where the first interval of another begins. A politician whose role as *mayorOfCity* ends the same year their role as *governorOfRegion* begins is a typical example. For sequences S and S', the Sequence_Meets constraint is fulfilled if:

$$M_{\triangleright}[meets][r(S,S')] = 1 \wedge \left(\sum_{a \in DR} M_{\triangleright}[a][r(S,S')] + M_{\triangleright}[a][r(S',S)] \right) = 1$$

- Sequence_before. Similar to the previous case, but here a gap must exist between the two sequences. A musician who leaves one band and only later joins another after several years of solo work exemplifies this condition. For sequences S and S', the Sequence_Before constraint is fulfilled if:

$$M_{\triangleright}[before][r(S,S')] = 1 \wedge \left(\sum_{a \in DR} M_{\triangleright}[a][r(S,S')] + M_{\triangleright}[a][r(S',S)] \right) = 1$$

3.5.5 . Temporal Constraints Generalisation

We generalise temporal constraints across all entities of class C using two metrics that accounts for data imperfections such as erroneous temporal data or outlier entities (error rate) and for constraints that are not applicable to a significant portion of entities within C (generalisation rate).

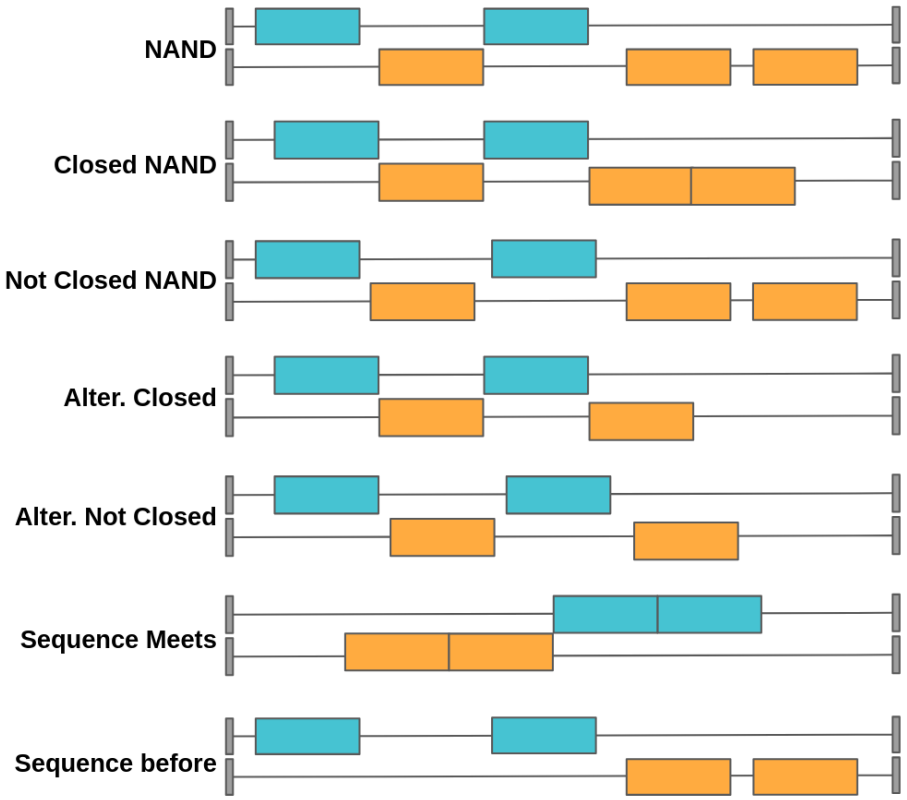


Figure 3.7: Disjoint Complex Constraints

Definition 15 (Error Rate) *Given a temporal constraint TC between two properties P and P', the set of entities $E_{P,P'}$ within the class C that are described by these two properties, and the sub-set of entities $X_{P,P'} \subseteq E_{P,P'}$ where TC is refuted, the Error Rate is defined as:*

$$ErrorRate(TC) = \frac{|X_{P,P'}|}{|E_{P,P'}|}$$

Definition 16 (Generalisation Rate) *Given a temporal constraint TC between two properties P and P', the set of entities E of the class C, and the set of entities $E_{P,P'} \subseteq E$ that are described by both properties, the Generalisation Rate is defined as:*

$$GeneRate(TC) = \frac{|E_{P,P'}|}{|E|}$$

Using an error threshold err and a generalisation threshold gen, a temporal constraint TC is generalised when: $ErrorRate(TC) \leq err \wedge GeneRate(TC) \geq gen$.

Algorithm 1 outlines the process for discovering and generalising temporal constraints between two properties P_1 and P_2, given a $\mathcal{TKG}$ and the thresholds err and gen. The algorithm initialises a HashMap ($mapTC$) to track the number of entities ($count$) fulfilling

Algorithm 1: Temporal Constraint Discovery & Generalisation

```
   Data: TKG, err, gen, P1, P2
1  begin
2      mapTC ← hashMap {tc1: 0, ..., tcn: 0}
3      for (e ∈ TKG.E) do
4          TP1 ← e.P1
5          TP2 ← e.P2
6          if (|TP1 + TP2| > 0) then
7              countEntities++
8              if (TempFunc(TP1) ∧ TempFunc(TP2)) then
9                  M▷ ← computeInterComp(TP1, TP2)
10                 M◁(TP1) ← computeIntraComp(TP1)
11                 M◁(TP2) ← computeIntraComp(TP2)
12                 pTC ← possibleTC(M▷, M◁(TP1), M◁(TP2))
13                 for (tc ∈ pTC) do
14                     mapTC.addOne(tc)
15     for ({tc : count} ∈ mapTC) do
16         if (ErrorRate(tc, count, countEntities) ≤ err) then
17             if (GenRate(tc, count, |E|) ≥ gen) then
18                 generalisedTC.add(tc)
19     globalTC ← filterRedundantTC(generalisedTC)
20     return globalTC
```

each temporal constraint, and a counter ($countEntities$) to track the number of entities described by both properties. For each entity in $\mathcal{TKG}$, we retrieve the temporal sequences (TP_1 and TP_2) for P_1 and P_2 and verify that they are both used to describe the entity (i.e. are not empty, line 6), and update the counter of entities accordingly. If the sequences are temporally functional (line 8), we compute their relevant inter- and intra-sequence comparisons, discovering possible temporal constraints for e and updating the HashMap accordingly. After the process has iterated over all entities, we evaluate whether each discovered temporal constraint can be generalised based on the error (line 16) and generalisation rates (line 17). Finally, we filter out the redundant constraints, by discarding from the final output any complex temporal constraint that have a more precise constraint in the $generalisedTC$ set.

Complexity Analysis.

Thanks to the relevance heuristic used to build the set of relevant inter-sequence comparisons $\Omega(S, S')$ (Definition 13), the discovery step does *not* compare all pairs of intervals

across two sequences. Both S and S' are ordered, and an interval in one sequence is only compared to its closest neighbors in the other (those that overlap or are immediately adjacent with no interval in between). This enables a two-pointer sweep, akin to a merge procedure, that scans each sequence once. Consequently, computing the set of relevant inter-sequence comparisons is linear in the sequence lengths:

$$\textbf{Inter-comparisons:} \qquad \mathcal{O}(|S| + |S'|).$$

Likewise, intra-sequence comparisons are restricted to successive intervals (Definition 14), hence linear as well:

$$\textbf{Intra-comparisons:} \qquad \mathcal{O}(|S|) \text{ for } S \text{ and } \mathcal{O}(|S'|) \text{ for } S'.$$

Putting this together, the cost to fill $M_{\triangleright}$ and $M_{\triangleleft}$ for a pair of properties (P, P') with sequences S and S' is $\mathcal{O}(|S|+|S'|)$. Aggregating over all entities e and all pairs of temporally functional properties $\mathcal{FP}_e$, whose number is $\frac{n(n-1)}{2}$ for $n = |\mathcal{FP}_e|$ we obtain:

$$\mathcal{O}\left(\sum_{e \in E} \sum_{(P,P') \in \mathcal{FP}_e} \left(|S_e^P| + |S_e^{P'}|\right)\right)$$

The generalisation step (error and generalisation rates) is a linear accumulation over entities and discovered constraints, hence does not change the overall asymptotic cost.

3.5.6 . Temporal Constraints for non-Functional Temporal Properties

Limiting to temporally functional properties in $\mathcal{TKG}$ may overlook significant constraints, especially for properties with diverse value types (e.g., `memberOf` linking to both musical groups and legislative bodies). To address this, we propose value-specialised properties to enable more precise constraint discovery, e.g.,figure 3.8 with Sonny Bono, `memberOf-SC` *(Sonny and Cher)* and `memberOf-USACongress`.

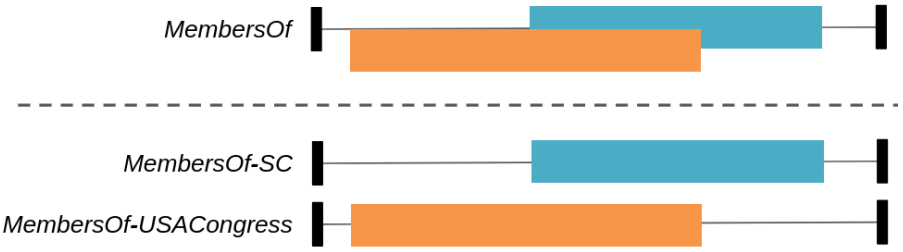


Figure 3.8: Example of value specialization

Definition 17 (Value Specialized Time Sequence) *The value specialized time sequence of an entity x of a property p and for a value $v \in \mathcal{I}$ (the set of entities as defined in the defintion 3) is the ordered set S of a set of quadruples $\{q_1, \ldots, q_n\}$, in the form of (x, p, v, I_k) such that $I_1.start$ is the earliest $I_n.start$ is the latest.*

Impact of Value Specialization.

Introducing value-specialized properties significantly increases the number of sequences to be processed. Instead of a single time sequence per property p, we now generate one sequence for each commonly shared value $v \in \mathcal{I}$, effectively multiplying the number of sequences by the number of selected values. If p admits $|V_p|$ specialized values, then the comparison cost for this property grows from $\mathcal{O}(|S|)$ to $\mathcal{O}(|V_p| \cdot |S_v|)$, where $|S_v|$ is the average length of the specialized sequences. In the global analysis, this means that the total cost becomes:

$$\mathcal{O}\left(\sum_{e \in E} \sum_{(p,p') \in \mathcal{FP}_e} (|V_p| + |V_{p'}|) \cdot (|S_e^p| + |S_e^{p'}|)\right).$$

While this refinement enables the discovery of more precise and semantically meaningful constraints, it also amplifies the computational burden, especially for highly connected properties with many distinct values. In practice, this motivates restricting value specialization to frequent values only, as illustrated in Figure 3.8, ensuring both scalability and statistical robustness from sufficient data samples.

3.6 . Constraint-based Temporal Fact Validation

This section extends our framework from the discovery and generalization of temporal constraints to their application in fact validation. The validation framework is outlined in Figure 3.9.

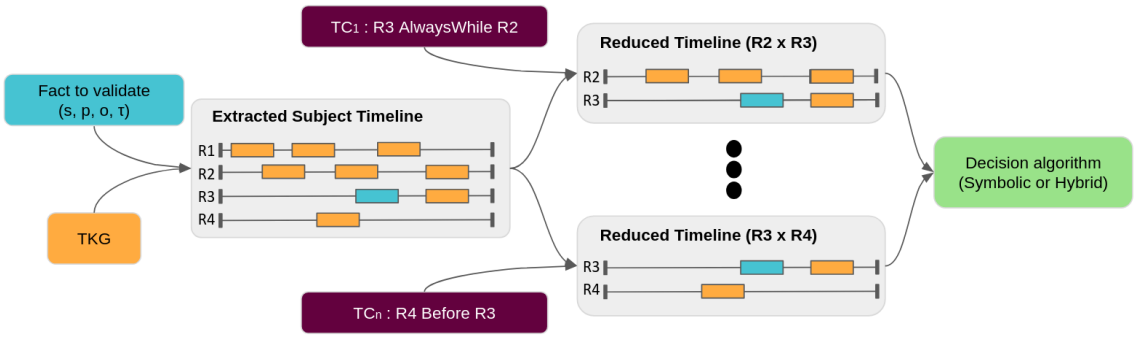


Figure 3.9: Constraint based Temporal Fact Validation framework

Given a fact $\langle s,\ p,\ o,\ \tau \rangle$ and a temporal knowledge graph TKG, the process begins by retrieving the subject timelines (depicted in orange), which contains all background information on the subject entity. The fact under validation (shown in blue) is integrated into this timeline. Subsequently, we identify relevant temporal constraints for the subject entity—constraints involving pairs of relations describing the subject. These constraints are then used to evaluate whether the addition of the fact adheres to or violates the constraints (Section 3.6.1). Finally, a decision algorithm consolidates the individual evaluations to determine whether the fact is validated or refuted (Section 3.6.2).

3.6.1 . Single Constraint Application

To validate a fact $\langle s,\ p,\ o,\ \tau \rangle$ in a Temporal Knowledge Graph ($\mathcal{TKG}$), we first identify all relevant temporal constraints. A constraint $tc = r(P_1, P_2)$, which involves properties P_1 and P_2, is considered as relevant if either the property corresponds to the predicate p of the fact, or corresponding to the pair $(p,\ o)$.

For each relevant constraint, we retrieve the time sequences S_1 and S_2 corresponding to properties P_1 and P_2 for the subject entity s. The time interval t of the fact is then incorporated into these sequences using one of two proposed insertion strategies as described on Figure 3.10:

1) Naive insertion strategy, where the time interval t is directly added to the sequence without addressing overlaps that might violate the temporal functionality of the property.

2) Remove insertion strategy, where any intervals overlapping with t are removed prior to the insertion.

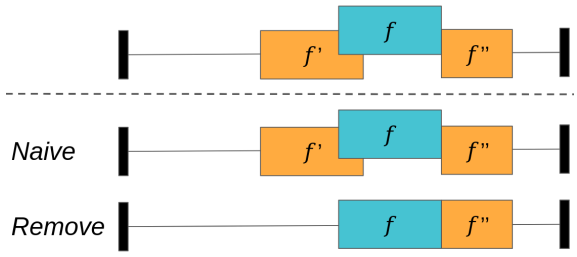


Figure 3.10: Insertion strategies for a new interval (blue)

In the following, we describe both strategies to highlight the key differentiating factors that are most relevant to our pipeline. While the Naive insertion allows us to preserve all intervals as they appear in the source TKG, it may lead to violations of temporal functionality as defined in definition 12. Consider, for instance, the property *presidentOf* with $(Obama, presidentOf, USA, [2009, 2017])$ or the property *memberOf* with $(JohnDoe, memberOf, TheRockers, [2000, 2005])$. If we attempt to validate false facts such as $(Obama, presidentOf, USA, [2016, 2020])$ or $(JohnDoe, memberOf, JazzBand, [2004, 2008])$, the Naive strategy will simply insert these new intervals, generating overlaps that contradict the assumption of temporal functionality and preventing our framework from being applied on the fact. In contrast, the Remove strategy discards the overlapping intervals prior to insertion, thus maintaining functional sequences but at the cost of erasing valid historical information. These situations illustrate the central trade-off between preserving the fidelity of the original knowledge and enforcing the formal consistency required by our framework.

Building on this trade-off, once a new interval has been inserted, whether preserved through the Naive strategy or adjusted through the Remove strategy, we then assess whether the resulting sequences S_1 and S_2 remain both temporally functional and non-empty. If these conditions are satisfied, we verify whether the sequences still adhere to

the constraint tc. A satisfied constraint supports the fact's temporal validity, while a violation indicates refutation.

3.6.2 . Overall Constraint Application

Moving beyond the single-constraint setting, we now aim to leverage the larger set of discovered constraints associated with the same property p. However, since these constraints may disagree when assessing a given fact, we propose two aggregation strategies, a symbolic method and a machine learning–based method, to derive a general decision on the fact's validity.

Symbolic combination strategy (voting-based).

This approach evaluates the temporal validity of a fact across all relevant constraints, as shown on the figure 3.11. For each constraint, we record whether it is satisfied, violated or non applicable (due to overlapping or lack of information), and the overall decision is obtained through a voting mechanism: if the proportion of satisfied constraints exceeds a predefined threshold, the fact is considered valid. This strategy has the advantage of being fully explainable, since it provides a transparent trace of the exact constraints that influenced the decision and makes it possible to justify each validation or refutation with concrete rules. Furthermore, the symbolic combination is lightweight from a computational perspective, ensuring that results can be produced quickly even when applied to large datasets. However, because all constraints are treated with equal importance, this strategy is less flexible in noisy settings, where some constraints may be more predictive than others or where exceptions are frequent.

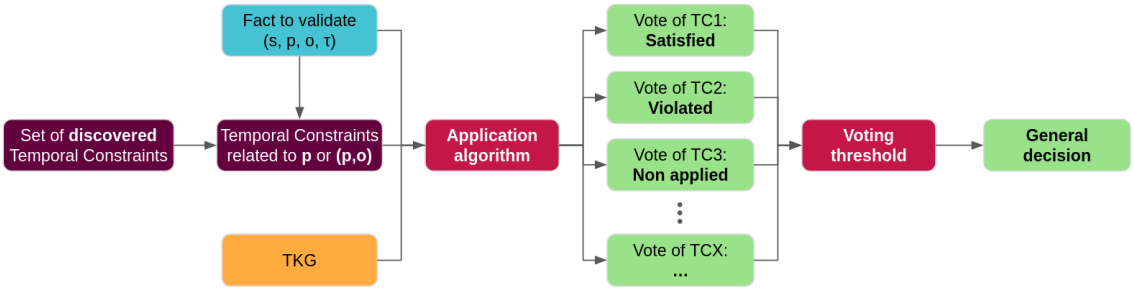


Figure 3.11: Symbolic combination strategy framework.

ML-based combination strategy.

This approach uses machine learning classifiers (e.g., random-forest, decision-tree, Neural Network) to integrate evaluations from all relevant constraints and is representated on the figure 3.12. Each constraint tc is assigned a numerical label to indicate if a constraint is satisfied (0) or violated (1). Additional labels (ranging from 2 to 4) reflect irrelevant constraints, disrupted temporal functionality, or empty time sequences. These labels are compiled into an $n \times m$ matrix, where n is the number of temporal constraints and m is the number of facts being validated. The matrix is paired with a ground truth vector for

training and testing the model, enabling the identification of the most effective constraints for temporal fact validation but at the cost of reduced explanation of the weight given for each constraint. This trade-off raises the challenge of ensuring that the learned decision boundaries remain interpretable enough to justify validation outcomes, an aspect that is crucial for maintaining trust in fact-checking applications. A possible mitigation lies in relying on inherently interpretable models, such as decision trees or attention-based mechanisms, which can highlight the relative importance of constraints while still benefiting from the adaptability of machine learning.[2]

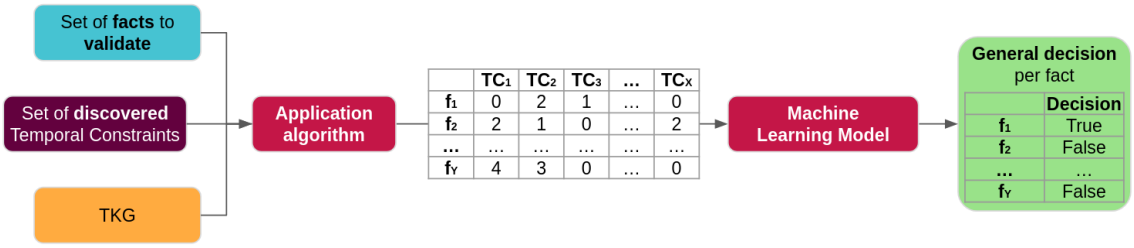


Figure 3.12: Machine learning combination strategy framework.

3.7 . Experimental Evaluation

This section presents the results of our experimental evaluation, highlighting the performance of the proposed temporal fact validation framework across several datasets. The source code and datasets of all the following experiments are available on GitHub[3]. All experiments were conducted on an "Intel® Xeon® E5-2630 v4" CPU with 10 cores and 128 GB RAM.

3.7.1 . Datasets and Experimental Setup

To evaluate our approach, we created three new datasets extracted from Wikidata [Vrandečić and Krötzsch(2014)], one of the most comprehensive knowledge graphs available. We focused on three distinct entity types: countries, musical groups, and politicians. Existing temporal datasets like TemporalFC [Qudus et al.(2023)] were unsuitable for our evaluation as they lacked essential features such as entity class information, literals, and proper temporal interval annotations. Table 3.3 provides an overview of our datasets, highlighting their varying characteristics in terms of number of entities and temporal facts (represented as quadruples).

The datasets were split into training (80%), validation (10%), and test (10%) sets. Negative samples were generated by corrupting only the temporal component of existing facts.

[2]Beyond model choice, post-hoc explainability techniques such as SHAP or LIME can also be applied to approximate the contribution of each constraint to the final decision, thus improving transparency without altering the prediction pipeline.

[3]https://github.com/SoulardThibaut/TemporalConstraints.git

For each entity, timestamps were randomized within a slightly extended range of its active lifespan, so as to produce plausible but invalid intervals. Temporal constraints were discovered from the positive training set only, while the augmented dataset including both positive and negative samples is used to train the ML classifier. A more detailed discussion of this negative sampling strategy, its variants, and its implications for benchmarking is provided in the next chapter (4).

Class	# Entities	# Quadruples
Country (Q6256)	205	183 249
Musical Group (Q215380)	55 507	131 476
Politician (Q82955)	658 445	2 085 232

Table 3.3: Description of the three datasets extracted from Wikidata

3.7.2 . Key Experiments and Results

To fine-tune our approach, we adopted a step-by-step strategy, starting with the decision strategy on combining temporal constraints (either symbolic or neuro-symbolic). Then, we examined the type of temporal constraints allowed (relation-level constraints or value-specialising properties), followed by the insertion strategy (either naive or remove). Lastly, we fine-tuned the error threshold alongside the generalisation threshold (default values: $ET = 5\%$ and $GT = 5\%$). Each experiment was evaluated using two metrics: *Accuracy* (Acc.) and the *Coverage* (Cov.) (see Defintion 18). Additionally, we tracked the running time (R.T), accounting for both temporal constraint discovery and the decision process for fact validation or refutation.

Definition 18 (Coverage) *Coverage is defined as the percentage of facts that can be verified by at least one temporal constraint over the total number of facts to be verified:*

$$coverage = \frac{|Facts_{_verified}|}{|Facts|}$$

Class	Deci. Type	Acc.	Cov.	R.T
Country	Symbolic	79.5	9.4	**2m 30s**
	ML-based	**80.4**	9.4	52m
Music Gr.	Symb.	64.0	37.5	**2m 50s**
	ML-based	**64.3**	37.5	5m 50s
Politician	Symb.	61.6	44.0	**44m**
	ML-based	**62.3**	44.0	1h 30m

Table 3.4: Constraints decision strategies

Decision Algorithm: Symbolic vs. ML-Based Combination.

Our first experiment assessed the impact of the decision strategy used to aggregate the outcomes of temporal constraints, by comparing the symbolic voting-based approach with the ML-based combination.

The results presented in Table 3.4 show that the symbolic strategy is highly efficient in terms of runtime, requiring only a few minutes even on large datasets. However, this comes at the cost of a slight decrease in accuracy compared to the ML-based approach, which systematically achieved marginally better results across all datasets. For instance, in the case of the Country dataset, symbolic reasoning reached an accuracy of 79.5% in about two minutes, while the ML-based strategy pushed the accuracy to 80.4% but required around an hour of computation. These results highlight a fundamental trade-off between transparency and efficiency on the one hand, and predictive performance on the other. While the symbolic strategy offers full explainability by explicitly listing the constraints used in each validation, the ML-based strategy adapts better to noisy settings by assigning different weights to constraints according to their predictive utility.

In practice, this suggests that the symbolic strategy is better suited for scenarios where traceability and real-time validation are essential, whereas the ML-based approach should be preferred in large-scale deployments where marginal improvements in accuracy justify the additional computational cost.

Class	Const. Type	Acc.	Cov.	R.T
Country	R	80.4	9.4	**52m**
	R & RxV	**90.8**	**14.1**	2h 35m
Music Gr.	R	64.2	37.5	5m 50s
	R & RxV	64.2	37.5	5m 50s
Politician	R	61.6	44.0	1h 30m
	R & RxV	61.6	44.0	1h 30m

Table 3.5: Constraints value specialization

Temporal Constraint Types: Regular vs. Value-Specialized.

The second experiment investigated the benefit of introducing value-specialized constraints in addition to regular constraints.

As shown in Table 3.5, the impact is dataset dependent. For the *Country* dataset, value specialization led to a remarkable increase in both accuracy (from 80.4% to 90.7%) and coverage (from 9.4% to 14.1%). This improvement is explained by the fact that entities of the same class often share recurrent values (e.g., many countries being members of the same international organizations). By distinguishing between these values, the system is able to discover finer-grained constraints that generalize across a larger portion of entities. In contrast, for the *Musical Group* and *Politician* datasets, no significant differ-

ence was observed, as the variety of associated values did not provide opportunities for specialization.

These results illustrate that while value specialization may not always be beneficial, it can drastically improve performance in contexts where entities exhibit common structural patterns. From a methodological perspective, this validates the importance of tailoring constraint discovery strategies to the semantic characteristics of the target dataset rather than assuming a one-size-fits-all solution.

Class	Insert Type	Acc.	Cov.	R.T
Country	Naive	**90.8**	14.1	**2h 35m**
	Remove	88.5	**14.4**	2h 54m
Musical Gr.	Naive	64.2	37.5	5m 50s
	Remove	64.2	37.5	5m 50s
Politician	Naive	**62.3**	44.0	**1h 30m**
	Remove	62.1	**46.9**	1h 32m

Table 3.6: Interval insertion strategies comparison

Insertion Strategies: Naive vs. Remove.

The third experiment compared two strategies for inserting candidate facts into the subject timeline: the *naive* strategy, which simply appends the interval regardless of overlaps, and the *remove* strategy, which eliminates overlapping intervals to preserve temporal functionality.

Results reported in Table 3.6 show that the naive approach yields slightly higher accuracy in most cases, while the remove strategy marginally increases coverage. For example, in the *Country* dataset, accuracy dropped from 90.8% to 88.5% when switching from naive to remove, but coverage increased from 14.1% to 14.4%. This trade-off reflects the different assumptions behind each strategy: the naive insertion favors strict preservation of the observed data, even at the cost of potential inconsistencies, whereas the remove strategy ensures temporal coherence by discarding conflicting intervals, thus allowing a larger set of facts to be validated. From a practical standpoint, the choice between the two strategies should depend on the application domain. In highly curated KGs with minimal noise, the naive strategy may suffice, whereas in noisier or user-generated graphs like Wikidata, the remove strategy is preferable as it provides more robust validation despite the small loss in accuracy.

Tuning Error and Generalization Thresholds.

The final experiment evaluated the influence of the error threshold (err) and the generalization threshold (gen) on the discovered temporal constraints.

As reported in Table 3.7, decreasing the generalization threshold and increasing the error tolerance led to a noticeable gain in coverage, with only a limited decrease in accuracy.

Class	gen	err	Acc.	Cov.	R.T
Country	2	10	-	-	-
	2	5	**88.6**	16	8h 50m
	5	10	87.9	**17.2**	4h 30m
	5	5	88.5	14.4	**2h 54m**
Musical Gr.	2	10	**64.6**	**38.2**	6m 6s
	2	5	**64.6**	**38.2**	5m 54s
	5	10	64.2	37.5	**5m 51s**
	5	5	64.2	37.5	**5m 51s**
Politician	2	10	63.4	**51.9**	1h 35m
	2	5	62.1	49.9	**1h 32m**
	5	10	**63.5**	48.9	1h 34m
	5	5	62.1	46.9	**1h 32m**

Table 3.7: Thresholds comparison

For instance, on the *Politician* dataset, setting $gen = 2$ and $err = 10$ resulted in a coverage of 51.9%, compared to 44.9% in the default configuration, with only a marginal accuracy loss. However, these improvements came at the cost of significantly higher runtimes, in some cases exceeding four hours due to the proliferation of candidate constraints (more than $24,000$ constraints in the worst case). This experiment demonstrates that constraint discovery can be flexibly adjusted to trade off between accuracy, coverage, and scalability.

In practice, the optimal configuration will depend on the requirements of the target application: lower thresholds are beneficial when maximizing coverage is essential, while stricter settings are preferable when computational efficiency or high precision are prioritized. These observations also reveal the need for advanced feature selection mechanisms to automatically filter redundant or uninformative constraints, thereby improving both runtime and interpretability.

3.7.3 . Summary and Takeaways

The experiments presented in Section 3.7.2 show that our temporal fact validation framework is sensitive to configuration choices, and that each component contributes differently to performance. Several general lessons can be drawn:

- There is a clear trade-off between accuracy and efficiency: symbolic strategies provide fast and fully explainable results, while ML-based aggregation yields slightly higher accuracy at the cost of runtime and interpretability.
- The usefulness of value-specialized constraints depends on the homogeneity of the dataset. They are highly effective in contexts where entities share common values, but less impactful for heterogeneous data.
- The insertion strategy reflects assumptions about data quality: naive insertion is effective when data is clean and consistent, whereas remove insertion provides

robustness in noisier settings.

- Tuning error and generalization thresholds allows balancing coverage against computational cost, but also reveals scalability challenges when too many candidate constraints are generated.

Overall, these findings highlight that our framework should be seen as a flexible toolbox rather than a fixed pipeline. Its components can be adapted to prioritize explainability, accuracy, or coverage depending on the needs of the target application. This adaptability is a central contribution, and it also motivates the next chapter, which focuses on benchmarking temporal validity across knowledge graphs.

3.8 . Conclusion

In this chapter, we introduced an explainable framework for the validation of temporal facts in knowledge graphs. The core contribution lies in extending Allen's Interval Algebra from simple intervals to time sequences, which enabled the discovery of both simple and complex temporal constraints. These constraints capture rich temporal dependencies between properties of entities and are automatically generalized at the class level, thus alleviating the need for manual rule design. Building on these constraints, we proposed two complementary validation strategies: a symbolic approach, ensuring full interpretability through explicit rule tracing, and a machine-learning approach, improving robustness in large-scale and noisy scenarios.

The experimental evaluation conducted on three Wikidata-derived datasets demonstrated the effectiveness of our framework. In particular, we achieved high validation accuracy (up to 90%) while maintaining reasonable coverage, and we showed how specialized temporal constraints and insertion strategies can significantly affect results. These experiments highlight an important trade-off between symbolic reasoning and ML-based combination strategies: while the latter improves accuracy, the former provides better transparency and efficiency. The framework therefore positions itself at the intersection of interpretability and adaptability, two often conflicting requirements in temporal reasoning over knowledge graphs.

Nevertheless, some limitations remain. First, coverage is still limited, as many properties lack discoverable temporal constraints or are associated with sparse entity timelines. Second, scalability challenges arise when increasing the number of constraints or lowering thresholds, since the search space grows quickly and impacts runtime. Finally, the evaluation is currently limited to three entity classes, which, while diverse, do not cover the full heterogeneity of temporal knowledge graphs. These aspects mark clear directions for future improvements.

Overall, this chapter demonstrated that temporal fact validation is not only feasible but also explainable, provided that appropriate temporal constraints are discovered and exploited. It thus lays the groundwork for a new research direction where explainability and data-driven adaptability can jointly enhance the reliability of temporal knowledge graphs.

3.9 . Future Work

Building on the foundations established in this chapter, several promising research avenues can be pursued:

3.9.1 . Scalability and Explainability.

A key research direction concerns the scalability of the ML-based validation strategy together with the preservation of explainability. As illustrated in Table 3.7, using more permissive thresholds leads to the discovery of a large number of temporal constraints. While this increases the number of features available to the machine learning algorithm, it also causes scalability issues, since many of these constraints are not relevant to the fact being validated. In our current implementation, every fact is represented by a feature vector of size n, where n corresponds to the full set of discovered constraints. In practice, however, only a much smaller subset n_p, those constraints related to the property p of the fact, is truly applicable. The inefficiency arises from always learning on n instead of on n_p, which results in high-dimensional but largely redundant representations. Reducing dimensionality by focusing only on n_p would allow us to keep thresholds slack in order to discover richer patterns while remaining efficient. Yet this introduces a new difficulty, since such a strategy would imply training distinct models for different subsets of features depending on the relation type.

Several research directions could be explored to overcome this limitation. Feature selection methods, clustering techniques, or dimensionality reduction could be applied to reduce the number of active constraints, directly improving scalability while still capturing meaningful patterns. Attention mechanisms, as introduced in transformer architectures, offer another option by dynamically weighting the most relevant constraints for each fact. Beyond these approaches, integrating temporal constraints into Large Language Models represents an important avenue. Such integration could inject structured knowledge absent from pretraining and provide a principled way for models to justify their decisions within an interpretable framework.

However, all these strategies raise questions about explainability. While feature selection or symbolic filtering preserve transparency, embeddings, attention mechanisms, and LLM integration tend to weaken the strict interpretability guaranteed by symbolic rules. One possible compromise is to design hybrid strategies that embed symbolic constraints within ML-based models. These would aim to retain interpretability while benefiting from the improved accuracy of data-driven methods. The central challenge, however, lies in ensuring that explanations remain faithful to the underlying symbolic rules, since hybridiza-

tion often risks producing only approximate or post-hoc justifications. The overall goal is therefore to balance scalability and robustness with the need for genuine and faithful explainability in temporal fact validation.

3.9.2 . Cross-Knowledge Graph Transfer

Another promising direction is to assess whether temporal constraints discovered in one knowledge graph (e.g., Wikidata) can be transferred to others (e.g., YAGO [Mahdisoltani et al.(2013)]). Such transfer would demonstrate the robustness and universality of temporal constraints while helping bootstrap validation in graphs that are less complete or less curated. However, this direction also raises several challenges. Knowledge graphs often differ in their schemas, with heterogeneous property definitions and class hierarchies that complicate the alignment of temporal constraints. They also vary in temporal granularity, since some encode precise timestamps while others rely on coarse intervals, and in coverage, with curated graphs such as YAGO being semantically rich but sparse, while DBpedia is broader but noisier. These differences question whether a constraint discovered in one setting can be meaningfully applied in another.

From the perspective of explainability, transferring constraints across heterogeneous graphs introduces further difficulties. Symbolic constraints retain their interpretability, but their meaning can be distorted when underlying ontologies diverge. Statistical or data-driven adaptations, on the other hand, might increase transferability but weaken the ability to trace decisions back to explicit rules. The challenge is therefore to ensure that transferred constraints preserve both their semantic validity and their explanatory power across different knowledge graphs. Achieving such transfer, together with consistent alignments between entities and schemas, would enable cross-KG veracity assessment, where the plausibility of a temporal fact can be confronted across multiple graphs. This perspective extends temporal fact validation frameworks toward cross-graph reasoning capable of detecting inconsistencies and reinforcing the overall reliability of distributed knowledge. The resource KG Nexus, presented in Chapter 6, constitutes a first essential step toward realizing this transfer and alignment infrastructure.

3.9.3 . Constraints Across Classes and Property Paths

A further direction is to extend constraint discovery beyond intra-class dependencies and consider constraints that span multiple classes or involve property paths. This would allow us to capture more complex temporal patterns, such as linking political mandates to historical events or associating musical careers with broader artistic movements. Such constraints would enrich temporal reasoning by accounting for interactions that go beyond a single entity type, thereby reflecting the interconnected nature of real-world knowledge.

However, this extension raises significant challenges. The search space of potential constraints grows rapidly when property paths are considered, making discovery and validation computationally demanding. Moreover, constraints discovered across heterogeneous classes may lack semantic clarity or may apply only to a narrow subset of entities, which risks reducing generalizability. From an explainability perspective, constraints that

span multiple classes or long property chains are harder to interpret, since the justification of a validation decision involves reasoning across multiple semantic layers.

Despite these challenges, pursuing this direction would provide the ability to model richer cross-domain dependencies. If handled carefully, it could extend temporal fact validation to scenarios where correctness depends not only on an entity's local timeline but also on how its history interacts with events and roles in related domains.

3.9.4 . Advanced Temporal Reasoning.

Another promising direction is to incorporate generalized extensions of Allen's Algebra [Ligozat(1991)] in order to reason over mixed temporal granularities and irregular temporal events. Such extensions would allow the framework to align facts expressed at different levels of precision, for instance combining century-level historical events with day-level political mandates. They would also enable the treatment of complex situations such as recurring events, uncertain dates, or incomplete intervals. This added expressiveness would make temporal fact validation applicable to a wider range of domains, including scientific datasets where measurements occur at irregular intervals, or historical archives where time information is often approximate.

Nevertheless, adopting generalized algebras comes with several challenges. Increasing the expressive power of temporal reasoning inevitably raises computational costs, as the number of possible relations grows significantly. Mixed granularities also raise semantic ambiguities: aligning coarse and fine intervals may introduce inconsistencies or require additional disambiguation rules. From an explainability perspective, more complex operators reduce the readability of constraints, since the justification of a validation decision must now rely on relations that are less intuitive than the original Allen primitives.

In spite of these difficulties, advanced temporal reasoning offers a valuable opportunity to capture the complexity of real-world temporal dynamics. If designed carefully, it could provide more accurate and context-sensitive validations, extending the scope of temporal knowledge graphs to domains where standard interval reasoning is insufficient.

3.9.5 . Benchmarking and Comparative Evaluation.

A critical next step is the creation of a benchmark for temporal fact validation that integrates both our framework and competing approaches, such as temporal scope prediction methods [Ling et al.(2021), Islakoglu et al.(2024), Jain et al.(2020)]. A standardized benchmark would provide a fair ground for comparison, ensuring that different approaches are evaluated on the same datasets, with consistent metrics and under comparable experimental conditions. This is essential for measuring real progress in the field, since current evaluations are often fragmented, relying on distinct datasets or inconsistent definitions of the validation task.

Designing such a benchmark raises several challenges. There is the question of dataset construction: existing temporal knowledge graphs vary in coverage, granularity, and data quality, which makes it difficult to ensure both realism and fairness. On the other hand, appropriate evaluation metrics must go beyond accuracy to also capture explainability, coverage, and scalability, since these aspects are central to the relevance of

temporal fact validation in practical settings.

In summary, building a benchmark for temporal fact validation is not only necessary to measure the strengths and weaknesses of existing methods, but also to guide future research toward approaches that combine accuracy, scalability, and explainability. This point naturally connects to the broader trajectory of this thesis: the next chapter introduces a dedicated benchmark for temporal validity in knowledge graphs, which directly addresses this research need and provides the foundation for systematic comparative evaluation.

Chapter 4

Benchmarking Temporal Validity in Knowledge Graphs

*The work presented in this chapter has been in collaboration with **Fatiha Saïs, and Joe Raad**. Part of this work has been published in :*

SOULARD Thibaut, SAÏS Fatiha, and RAAD Joe.
When Facts Expire: Benchmarking Temporal Validity in Knowledge Graphs,
in : CIKM 2025.

In the previous chapter, we introduced an explainable framework for temporal fact validation based on the discovery and application of temporal constraints. While the experiments demonstrated the effectiveness of this approach, they also revealed the need for systematic evaluation protocols capable of comparing different methods under consistent conditions. Building on this motivation, the present chapter introduces a dedicated benchmark for temporal validity in knowledge graphs. Unlike generic temporal knowledge graphs designed in existing benchmarks for link prediction, this benchmark specifically targets the validation task by focusing on corrupted temporal intervals and by offering data enriched with semantic information from publicly accessible knowledge graphs. It thus directly addresses the limitation identified in the previous chapter and provides the foundation for fair and reproducible comparisons between symbolic, neural, and hybrid approaches to temporal fact validation.

4.1 . Introduction

Benchmarks play a central role in the development of machine learning and knowledge representation methods. In the context of Temporal Knowledge Graphs (TKGs), resources such as ICEWS[García-Durán et al.(2018), Gastinger et al.(2024)], GDELT[Leetaru and Schrodt(2013)], YAGO[Dasgupta et al.(2018),

García-Durán et al.(2018)], or Wikidata[Gastinger et al.(2024), Dasgupta et al.(2018), Lacroix et al.(2020)]-derived corpora have been instrumental in advancing temporal link prediction by providing large-scale, temporally annotated datasets. Yet, these resources assume that the observed facts are correct and evaluate only the ability of models to predict missing ones. None of them isolate the task of verifying whether the temporal scope of existing facts is valid. Moreover, they generally omit ontologies and literals, thereby restricting the evaluation of symbolic and hybrid methods. As a consequence, temporal fact validation still lacks a common benchmark, making it difficult to ensure comparability and reproducibility across studies.

The importance of temporal validity is well illustrated by real-world examples. In the biomedical domain, the fact $(\text{COVID-19}, \text{declaredAs}, \text{PublicHealthEmergency})$ was valid from early 2020 until May 2023, when the World Health Organization lifted the emergency status. A system unaware of this temporal scope would incorrectly classify COVID-19 as still under emergency management. In the geopolitical domain, $(\text{United Kingdom}, \text{memberOf}, \text{European Union})$ held until January 2020, when Brexit came into effect. Ignoring such temporal boundaries would result in outdated or misleading inferences. These examples underscore that validation is not a secondary concern but a prerequisite for trustworthy reasoning.

To address these shortcomings, we introduce in this chapter a benchmark specifically designed for temporal fact validation. Built from Wikidata[Vrandečić and Krötzsch(2014)], it departs from completion-oriented datasets by corrupting only the temporal dimension of facts while preserving their structural components. This design isolates the temporal aspect of reasoning and allows models to be tested on their ability to detect incorrect or inconsistent time intervals, without conflating the task with entity or relation prediction. To support a broad range of approaches, the benchmark also integrates ontological hierarchies, textual descriptions, and datatype properties, enabling symbolic, neural, and hybrid reasoning paradigms to be evaluated on equal footing.

In this chapter, we therefore ask the following research question: **How can temporal fact validation be benchmarked systematically across symbolic and neural paradigms, in a way that ensures fairness, reproducibility, and scalability?**

The remainder of this chapter is organized as follows. Section 4.2 reviews related work on temporal reasoning tasks, modeling paradigms, and datasets. Section 4.3 outlines the main characteristics that we believe are necessary for a benchmark. Section 4.4 details the construction of the benchmark, including data extraction, preprocessing, and negative sampling strategies. Section 4.5 introduces the temporal fact validation task defined on top of the benchmark and reports preliminary evaluation results. Finally, Section 4.6 concludes the chapter, while Section 4.7 outlines perspectives for future improvement of the benchmark.

4.2 . Related Work

As we have briefly presented in the previous chapter, Temporal Knowledge Graphs (TKGs) have been extensively studied in recent years[Cai et al.(2024)], giving rise to a variety of reasoning tasks, modeling paradigms, and benchmark datasets. Yet, despite this growing literature, the specific problem of *temporal fact validation* remains underexplored. Previous works have primarily concentrated on predicting missing temporal facts or answering time-sensitive queries, while the evaluation of whether existing facts are valid within their asserted temporal scope has received little attention. This asymmetry explains why the previous chapter had to rely on ad hoc datasets: a systematic benchmark dedicated to validation simply does not exist.

To position our contribution, this section reviews three complementary strands of related work. First, we examine the *reasoning tasks* that have been defined over TKGs, highlighting their goals and usage differences. Then, we describe the existing datasets in the field of Knowledge Completion and outline the requirements for designing a benchmark for temporal validation.

4.2.1 . Temporal Knowledge Graph Reasoning Tasks

Reasoning over Temporal Knowledge Graphs (TKGs) has led to the development of a diverse range of tasks that often extend or adapt their canonical counterparts from the static setting [Zhang et al.(2024)]. These tasks span various objectives, including improving entity alignment techniques, enabling more expressive question answering, and performing temporally aware prediction.

Entity Alignment

The objective of entity alignment is to identify pairs of entities that correspond to the same real-world object, following the `sameAs` semantics defined in OWL[1]. In the static case, this task reduces to comparing structural or semantic similarities between two knowledge graphs. However, such an approach can be misleading when the entities under comparison evolve or change across time.

Consider the following example. In two static knowledge graphs that lack temporal qualifiers, the neighborhoods of the original *Blade Runner* (1982) in the first graph and of its later remastered edition in the second, *Blade Runner: The Final Cut* (2007), may appear nearly identical:

[1] https://www.w3.org/TR/owl-ref/#sameAs-def

KG_1 : $\langle BladeRunner1982,\ hasTitle,\ \text{“}BladeRunner''\rangle$
$\langle RidleyScott,\ hasDirected,\ BladeRunner1982\rangle$
$\langle WarnerBros,\ produced,\ BladeRunner1982\rangle$
$\langle BladeRunner1982,\ hasGenre,\ ScienceFiction\rangle$
$\langle BladeRunner1982,\ hasLanguage,\ English\rangle$

KG_2 : $\langle BladeRunner2007,\ hasTitle,\ \text{“}BladeRunner : TheFinalCut''\rangle$
$\langle RidleyScott,\ hasDirected,\ BladeRunner2007\rangle$
$\langle WarnerBros,\ produced,\ BladeRunner2007\rangle$
$\langle BladeRunner2007,\ hasGenre,\ ScienceFiction\rangle$
$\langle BladeRunner2007,\ hasLanguage,\ English\rangle$

From this static perspective, an alignment algorithm could easily (and incorrectly) infer:

$$\langle BladeRunner1982,\ owl{:}sameAs,\ BladeRunner2007\rangle$$

Introducing temporal annotations, however, resolves this ambiguity. Once time intervals are considered, the distinction between the two becomes explicit:

KG_1 : $\langle BladeRunner1982,\ hasTitle,\ \text{“}BladeRunner'',\ [1982, today]\rangle$
$\langle RidleyScott,\ hasDirected,\ BladeRunner1982,\ 1982\rangle$
$\langle WarnerBros,\ produced,\ BladeRunner1982,\ 1982\rangle$

KG_2 : $\langle BladeRunner2007,\ hasTitle,\ \text{“}BladeRunner : TheFinalCut'',\ [2007, today]\rangle$
$\langle RidleyScott,\ hasDirected,\ BladeRunner2007,\ 2007\rangle$
$\langle WarnerBros,\ produced,\ BladeRunner2007,\ 2007\rangle$

The temporal knowledge graph now reveals that *Blade Runner* (1982) and *Blade Runner: The Final Cut* (2007) correspond to distinct entities released at different times, despite sharing almost identical neighborhoods. This temporal differentiation prevents a false positive in entity alignment, underscoring how TKGs reduce ambiguity and improve reasoning accuracy compared to static KGs.

Temporal Question Answering

Another important reasoning task over TKGs is question answering [Song et al.(2023), Jiao et al.(2023)], where the goal is to retrieve answers to natural language questions grounded in the graph. In a static knowledge graph, questions can only be answered without reference to time, which severely restricts the range of possible queries. For instance, one may ask *“Who directed Blade Runner?”* and obtain the answer *Ridley Scott*, but such a graph cannot distinguish between the 1982 original release and the 2007 remastered edition.

By contrast, temporal question answering allows queries that explicitly involve time. Consider the question:

“Who directed Blade Runner in 1982?”

A static knowledge graph containing both *Blade Runner (1982)* and *Blade Runner: The Final Cut (2007)* would either fail to distinguish the two or provide a contradictory answer set. A temporal knowledge graph, however, can resolve the ambiguity by exploiting temporal qualifiers:

$$\langle RidleyScott,\ hasDirected,\ BladeRunner1982,\ 1982\rangle$$
$$\langle RidleyScott,\ hasDirected,\ BladeRunner2007,\ 2007\rangle$$

The system can thus correctly return *Ridley Scott* in the context of 1982, while distinguishing it from later editions.

Similarly, temporal qualifiers allow temporal queries that would otherwise be more complex to resolve. For example, the question

"Who was the President of the United States in 2010?"

cannot be reliably answered in a static graph that contains both *Obama* and *Trump* as possible candidates. With temporal annotations, however, the graph explicitly encodes:

$$\langle Lincoln,\ headOfState,\ USA,\ [1861, 1865]\rangle$$
$$\langle Obama,\ headOfState,\ USA,\ [2009, 2017]\rangle$$

and thus correctly returns *Obama* as the answer.

These examples illustrate how TKGs enable fine-grained temporal reasoning, extending question answering capabilities far beyond the scope of static KGs.

Temporal Link Prediction and Forecasting

Link prediction is a central reasoning task in both static and temporal knowledge graphs, where the objective is to infer missing information from incomplete triples. In the static case, this typically involves predicting the head, relation, or tail of a triple such as $\langle Obama,\ headOfState,\ ?object\rangle$. Extending this setting to temporal knowledge graphs introduces time as an additional component, so that the task may also focus on predicting the temporal validity of a fact, for instance in $\langle Obama,\ headOfState,\ USA,\ [?start{-}?end]\rangle$. Thus, temporal link prediction simultaneously encompasses the classical prediction of entities and relations, as well as the identification of appropriate time intervals.

Recent surveys [Cai et al.(2022), Wang et al.(2023)] distinguish two principal scenarios.

- **Interpolation** aims to infer missing facts within the observed time window. For example, if $\langle Obama,\ headOfState,\ USA,\ [2009, 2017]\rangle$ and $\langle Lincoln, headOfState, USA, [1861, 1865]\rangle$ are known, a model may predict related facts such as the corresponding vice presidents within the same periods.
- **Extrapolation** aims to forecast facts beyond the observed window, such as predicting the next head of state of the United States after 2029.

While these approaches focus on *completing* the graph by generating new facts, they leave aside a complementary but equally important problem: ensuring that the temporal scope of existing facts is consistent with the rest of the graph. This is precisely the goal of *temporal fact validation*. Unlike prediction, validation does not seek to create new knowledge but to verify the plausibility of what is already asserted. For example, a validation system should flag $\langle Obama,\ headOfState,\ USA,\ [2019, 2021]\rangle$ as incorrect, since it contradicts the established sequence of U.S. presidents. Despite its importance for applications such as fact-checking and misinformation detection, temporal validation remains underexplored, with only a few preliminary attempts such as TemporalFC [Qudus et al.(2023)], which is limited to timestamp-based reasoning and does not explicitly model temporal constraints.

4.2.2 . Temporal Link Prediction Dataset Characteristics

The ability to study and compare temporal reasoning tasks critically depends on the availability of suitable datasets. Indeed, tasks such as interpolation, extrapolation, and temporal validation can only be meaningfully evaluated when the underlying temporal information is both accurate and representative of real-world dynamics. In this section, we therefore review prominent TKG datasets that have shaped the field, with particular attention to their temporal representations and evaluation protocols. Notably, the NeurIPS Temporal Graph Benchmark (TGB) initiative [Gastinger et al.(2024)] has recently provided a unified benchmark suite designed to foster systematic comparison across models. Table 4.1 summarizes the main characteristics of these datasets, including their domains, temporal encoding strategies, time granularity, and the types of generalization they support.

Name	Dom.	Infor.	T. Repr.	Prec.	Infer.	OLS.
DBpedia34K [Nayyeri et al.(2022)]	Broad	T,G	TStamps	Y	E,T	Could
DBpedia124K [Qudus et al.(2023)]	Broad	T	TStamps	Y	E	Could
GDELT [Leetaru and Schrodt(2013)]	Geopo.	T	TStamps	15 Min	E,R,T	No
ICEWS14 [García-Durán et al.(2018)]	Geopo.	T	TStamps	Day	E	No
ICEWS05-15 [García-Durán et al.(2018)]	Geopo.	T	TStamps	Day	E	No
ICEWS18 [Jin et al.(2020)]	Geopo.	T	TStamps	Day	E,R,T	No
TKGL ICEWS [Gastinger et al.(2024)]	Geopo.	T	TStamps	Day	E,R,T	No
TKGL Polecat [Gastinger et al.(2024)]	Geopo.	T	TStamps	Day	E,T	No
TKGL Smallpedia [Gastinger et al.(2024)]	Broad	S,T	Intervals	Y	E,R,T	Could
TKGL Wikidata [Gastinger et al.(2024)]	Broad	S,T	Intervals	Y	E,R,T	Could
Wikidata12k [Dasgupta et al.(2018)]	Broad	T	Intervals*	Y	E,T	Could
Wikidata432K [Lacroix et al.(2020)]	Broad	S,T	Intervals*	Var.	T	Could
Yago3K [Nayyeri et al.(2022)]	Broad	T,G	TStamps	Y	E	Could
Yago11K [Dasgupta et al.(2018)]	Broad	T	Intervals*	Var.	E,T	Could
Yago15K [García-Durán et al.(2018)]	Broad	S,T	Intervals*	Var.	E,T	Could

Table 4.1: Structural and temporal characteristics of TKG datasets

Name	# Facts	# Entities (deg.)	# Relations (use)	# Timestamps (use)
DBpedia34k	35 845	29 557 (2)	7 (5 120)	873 (41)
DBpedia124k	131 285	110 791 (2)	7 (18 755)	123 (1 067)
GDELT	1.73M	7 146 (485)	238 (7 287)	2 138 (811)
ICEWS14	72 826	6 869 (21)	230 (317)	365 (399)
ICEWS05-15	368 962	10 094 (73)	251 (1 470)	4 017 (18)
ICEWS18	373 018	21 085 (35)	251 (1 486)	240 (1 554)
TKGL ICEWS	21.7M	79 938 (544)	546 (39 786)	7 187 (3 023)
TKGL Polecat	2.49M	117 435 (43)	32 (77 910)	1 193 (2 090)
TKGL Smallpedia	775 514	31 556 (49)	504 (1 539)	98 (7 913)
TKGL Wikidata	13.97M	711 721 (39)	1 008 (13 854)	1 999 (6 986)
Wikidata12k	32 497	12 544 (5)	24 (1 354)	600 (108)
Wikidata432k	7.22M	432 715 (33)	407 (17 750)	39 692 (364)
Yago3k	161 708	116 624 (3)	8 (20 214)	123 (1 315)
Yago11k	16 408	10 618 (3)	10 (1 641)	4 119 (8)
Yago15k	17 130	4 098 (8)	10 (1 713)	189 (181)

Table 4.2: Entity, relation, and temporal statistics for TKG datasets (temporally contextualized edges only).

Existing datasets can broadly be grouped into **two main domains**. The first category consists of *broad encyclopedic knowledge graphs*, such as Wikidata, YAGO, and DBpedia. These resources are derived from collaboratively curated or semi-automatically extracted encyclopedic content and thus cover a wide range of domains, from history and geography to culture and science. Their temporal dimension typically appears in the form of qualifiers or validity intervals attached to factual statements, for instance encoding the mandate of a political leader, the lifespan of a person, or the release year of a movie. While their coverage is extremely broad, the temporal information is often sparse and heterogeneous, since it depends on the availability of explicit timestamps in the source material.

The second category is composed of *geopolitical event graphs*, including ICEWS, GDELT, and Polecat. Unlike encyclopedic KGs, these datasets are specifically designed to capture dynamic event streams, where each entry corresponds to a political, diplomatic, or economic event extracted from news sources. They exhibit high temporal resolution, often down to daily or even hourly granularity, and are commonly used for tasks such as political stability assessment or conflict forecasting. For example, ICEWS records fine-grained interactions such as $\langle Country\, A,\ protest,\ Country\, B, 2019-04-15\rangle$, while GDELT aggregates millions of events across the globe with additional sentiment and actor-type annotations. These datasets thus provide dense temporal coverage but are more limited in scope, as their focus remains confined to the geopolitical domain.

Taken together, encyclopedic KGs and event-centric graphs represent complementary perspectives: the former emphasize *breadth of domains* with relatively coarse temporal annotations, whereas the latter emphasize *temporal density and precision* within a narrower application area.

In addition, existing benchmarks differ along several important dimensions that directly impact their usability for temporal reasoning :

Temporal Representation

One fundamental distinction concerns whether datasets encode temporal information as *timestamps* or as *intervals*. Event-centric datasets such as ICEWS and GDELT rely on timestamps, with each fact tied to a specific point in time, reflecting the instantaneous nature of political events. By contrast, encyclopedic resources like YAGO15K or TKGL Wikidata favor interval representations, where facts are valid over a continuous duration (e.g., a person's mandate or a historical treaty). These two strategies reflect different modeling assumptions: timestamps capture fine-grained dynamics but struggle with long-term validity, while intervals enable reasoning over durations but require careful handling of overlapping or uncertain boundaries.

Temporal Precision

The granularity of temporal annotations also varies widely. Some benchmarks are limited to coarse values, as in DBpedia124K where only years are available, making it difficult to distinguish between events occurring in the same year. Others provide extremely fine resolution: GDELT, for example, records events down to 15-minute intervals, enabling real-time monitoring but also producing massive volumes of data. Intermediate cases like Wikidata432K mix granularities (years, months, days), creating heterogeneity that complicates the design of learning models.

Partial Timestamps

A recurring challenge is the prevalence of incomplete temporal data. Datasets such as YAGO11K and Wikidata12K include unknown boundaries, for instance $[?, 1990]$ or $[2005, ?]$, when the start or end of a fact is unknown. While such partial information is common in real-world knowledge graphs, most current benchmarks and models treat missing boundaries with simplistic placeholders, leaving the issue largely underexplored. Developing methods that can reason under partial temporal knowledge remains an open challenge.

Inference Scope

Benchmarks also differ in the type of generalization they expect from models. Depending on the dataset split, systems may need to generalize to unseen timestamps (T), entities (E), or relations (R). For instance, TKGL ICEWS and ICEWS18 are particularly demanding, as they test all three dimensions simultaneously. This design better reflects real-world deployment scenarios, where both new entities and new time periods must be handled, but it also makes comparison across models less straightforward.

Ontology and Literal Support

Finally, despite being extracted from rich knowledge graphs such as Wikidata or DBpedia, most existing temporal benchmarks omit ontological information (e.g., class hierarchies) and literals (e.g., numerical or textual attributes). As a result, users who wish to exploit this metadata for reasoning must reconstruct it independently, leading to inconsistencies and hampering fair comparison.

Structural characteristics

Beyond these semantic aspects, structural characteristics, as presented in Table 4.2, also play a critical role in shaping model behavior and evaluation outcomes. The datasets differ by several orders of magnitude in size and relational complexity. On the smaller end, resources such as YAGO11k (16k facts, 10k entities) or DBpedia34k (36k facts, 29k entities) provide only limited training signals, which can hinder the learning capacity of both symbolic approaches and neural architectures. At the other extreme, large-scale benchmarks like TKGL ICEWS (21.7M facts, nearly 80k entities, and 7.2k timestamps) or TKGL Wikidata (14M facts, more than 700k entities) are well suited to neighborhood-aware deep models but pose significant scalability challenges due to their volume. In addition to size, the number of relations also influences modeling strategies. Event-centric datasets such as ICEWS14 (230 relations) or ICEWS18 (251 relations) offer a relatively focused schema, while encyclopedic resources like TKGL Wikidata or TKGL Smallpedia include more than 500 distinct relations, reflecting the complexity of real-world knowledge yet complicating generalization and transfer across domains.

Temporal coverage

Temporal coverage also varies widely. Some datasets, such as ICEWS05-15, span more than 4k distinct timestamps, while others like DBpedia124k or YAGO3k cover just over one hundred points in time, leading to a coarse-grained temporal view. Precision further differs across families: GDELT, with 1.7M facts distributed over more than 2k timestamps, captures dense daily events, whereas YAGO11k contains 4k timestamps but with only a handful of facts per time point. This imbalance means that even datasets with broad temporal horizons may provide very sparse supervision at each interval.

Challenges in Evaluating Temporal Validity

Taken together, existing research on temporal reasoning tasks and datasets highlights a fundamental trade-off between *coverage* and *density*. Event-centric resources such as GDELT or TKGL ICEWS emphasize dense temporal resolution but remain limited to narrow geopolitical domains, whereas encyclopedic datasets like DBpedia124k or Wikidata432k provide broad entity coverage at the cost of sparse or heterogeneous temporal annotations. This diversity complicates direct comparison across methods and, more importantly, none of these benchmarks explicitly isolate the task of temporal fact validation.

As a result, consistency and reliability in evaluating whether facts are valid within their temporal scope remain underexplored. This gap motivates the benchmark introduced in the next section, which is specifically designed to assess models on their ability to reason about temporal validity rather than completion or forecasting.

4.3 . Benchmark Description

To address the limitations identified in existing temporal knowledge graph datasets, we introduce a novel benchmark specifically designed for the task of temporal fact validation. Unlike prior resources, which focus primarily on completion or forecasting, our benchmark emphasizes the verification of temporal plausibility. It is characterized by the following key features:

Negative sampling via temporal corruption

A central challenge in evaluating temporal fact validation is the construction of realistic negative examples. Instead of relying on random sampling, which may generate trivially false or semantically incoherent triples, we design a corruption strategy focused on the temporal dimension. Valid intervals are systematically perturbed into plausible but incorrect alternatives, for instance shifting a presidential mandate by a few years. By ensuring that corrupted intervals remain non-overlapping with valid ones, we obtain challenging test cases that require fine-grained temporal reasoning rather than superficial heuristics.

Temporal isolation

A recurrent limitation of existing benchmarks is that they fail to distinguish temporal reasoning with static structural cues. As a result, models can obtain deceptively strong performance without actually exploiting temporal information. For instance, [Jain et al.(2020)] report that purely static embedding models, which ignore time altogether, still achieve competitive results against temporally-aware counterparts. This indicates that evaluation was largely driven by structural regularities in the graph (e.g., frequent co-occurrence of entities) rather than genuine temporal reasoning.

To address this issue, our benchmark enforces a strict separation between temporal and non-temporal information. During negative sampling, only the temporal component of a fact is corrupted, while entities and relations remain intact. This design prevents models from relying on non-temporal shortcuts and ensures that correct predictions cannot be made without considering time. In practice, a system that succeeds on our benchmark must therefore demonstrate the ability to handle temporal constraints explicitly, rather than relying on generic link prediction strategies.This temporal isolation is crucial for fact validation, where the key question is not whether a relation holds in general, but whether it holds at the appropriate point in time.

Ontology and literal inclusion

While most temporal benchmarks reduce the graph into simple triples, our benchmark explicitly integrates schema-level information (e.g., class hierarchies, domain and range constraints) as well as literals such as dates, numerical attributes, or textual descriptions. This enrichment allows the evaluation of methods that combine symbolic reasoning with numerical or textual grounding, bridging the gap between purely neural approaches and hybrid systems. For example, consider the case of a professional boxer participating in a lightweight competition. Besides temporal information specifying when the competition takes place, the plausibility of the fact also depends on the literal attribute `hasWeight`. If the knowledge graph records that the boxer weighed 72kg at the time of the competition, then the statement $\langle Boxer, competesIn, LightweightChampionship, 2021 \rangle$ should be considered invalid, since the ontology constrains the lightweight class to a maximum of 61kg. Without access to literal values and ontological rules, such inconsistencies would remain undetected.

Including these dimensions reflects the reality of large scale Knowledge Graphs such as Wikidata, where ontologies and literals are inseparable from factual content. Supporting them explicitly allows the benchmark to probe temporal reasoning that goes beyond structural patterns, so that evaluation captures both semantic and numerical constraints. Semantic constraints arise from the schema and the intended meaning of relations, for instance type compatibility, domain and range that may evolve through time, functional and cardinality expectations for roles, or implications that link events and states across intervals. A person who holds the office of mayor must be typed as human (except for some dogs), the office must correspond to a city level position, and the tenure must not overlap with a legally incompatible role. Numerical constraints revolve around arithmetic over literals and timestamps, such as minimal or maximal mandate duration, age at the time of an event, non overlap or bounded gaps between successive intervals, monotonic evolution of quantities through time, or aggregation checks like the count of appearances per season. Taken together, these two families ensure that systems are assessed on their ability to respect meaning as defined by the schema while also reasoning over quantities and time, which yields a more faithful measure of temporal validity across heterogeneous graphs.

Scalable and fair graph sampling.

To account for the diversity of deployment settings, we provide multiple strategies for entity selection and graph slicing. This enables evaluation under different levels of sparsity and scale, ranging from small subgraphs suitable for symbolic methods to larger, denser settings required for deep learning approaches. By defining standardized sampling criteria, we also ensure fairness across methods, preventing models from being advantaged or disadvantaged by dataset-specific biases.

Taken together, these features fill an important gap in the landscape of temporal knowledge graph benchmarks. Our resource supports the underexplored task of tempo-

ral fact validation by offering semantically rich data, realistic temporal corruptions, and explicit ontological context. It enables rigorous evaluation focused on temporality rather than static structure, while remaining scalable and broadly adaptable. The benchmark is publicly available on Zenodo[2], and the construction code is released on GitHub[3], both under the MIT License[4].

In the following section, we describe in detail the process by which the benchmark was constructed, including the sources of temporal data, the corruption strategies, and the sampling procedures.

4.4 . Benchmark Construction

Having outlined the key features that distinguish our benchmark from existing temporal datasets, we now turn to its practical construction. The objective of this section is to detail how raw temporal information was extracted, processed, and enriched in order to create a resource suitable for rigorous temporal fact validation. The construction pipeline follows a progressive logic: we begin by extracting temporally qualified statements from Wikidata, then preprocess and filter the resulting graph to ensure sufficient connectivity and descriptive richness of entities, and finally generate negative samples and semantic enrichments that make the benchmark both challenging and versatile.

4.4.1 . Data extraction

The construction of our benchmark begins with the extraction of temporally contextualized facts from Wikidata, one of the largest publicly available knowledge bases, which provides collaboratively curated, multilingual, and schema rich descriptions of real world entities across a wide range of domains. We relied on the official May 25th, 2023 dump[5], which provides a stable and reproducible snapshot of the graph. Wikidata's statement model is particularly suitable for temporal reasoning, as qualifiers can be used to attach time related metadata directly to subject predicate object triples.

To capture temporal information, we focused on three widely used properties: `Point In Time` (**P585**), `Start Time` (**P580**), and `End Time` (**P582**). The first corresponds to punctual events such as the date of a specific award ceremony or a match result, while the second and third delimit intervals describing longer-lasting situations such as political mandates, memberships, or artistic collaborations. By considering both punctual and interval-based annotations, we ensured that the extracted data covers a wide spectrum of temporal phenomena, from instantaneous events to multi-decade relations.

This extraction yielded a total of **21,766,539** temporally annotated statements, of which **11,638,976** are timestamped (via P585) and **10,127,563** are interval-based (via P580/P582). Notably, many intervals are only partially defined, with missing start or end

[2] https://doi.org/10.5281/zenodo.17201109
[3] https://github.com/SoulardThibaut/WhenFactExpire
[4] enabling unrestricted use and modification
[5] https://www.wikidata.org/wiki/Wikidata:Database_download

boundaries (such as `[?, 1990]`), reflecting the incomplete nature of historical records. While these partial definitions introduce additional challenges for modeling, we chose to preserve them in order to maintain fidelity to the source data and to encourage the development of methods capable of reasoning under temporal uncertainty.

In total, the extracted facts involve **8,160,539** unique entities, illustrating both the richness and the heterogeneity of Wikidata. This large-scale temporal foundation allows our benchmark to inherit the diversity of Wikidata's domains, while focusing specifically on the temporal dimension that is often underexploited in existing benchmarks.

4.4.2 . Graph Preprocessing

Temporal Scopes

After extracting the raw temporal data, we divided the benchmark into two complementary temporal scopes in order to test distinct model capabilities.

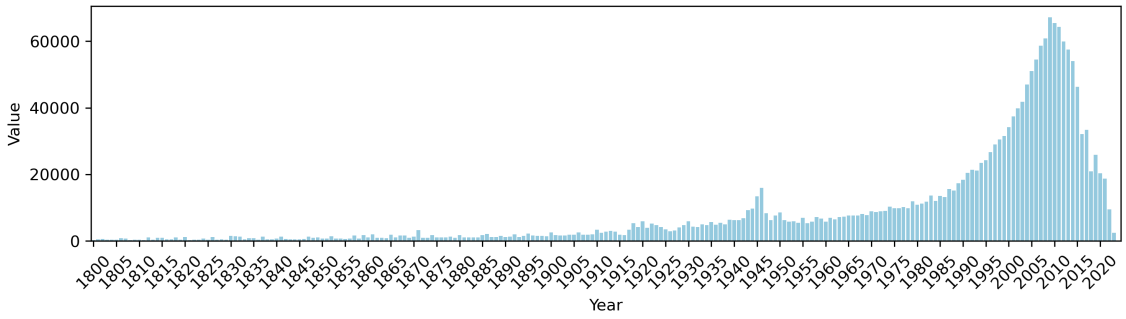


Figure 4.1: Number of timestamps per year in the reduced temporal scope (TR).

The first setting corresponds to the **Reduced temporal scope (TR) [1900–2023]**. This period provides a rich and relatively consistent set of historical facts, concentrated in the modern era where documentation is abundant. Figure 4.1 plots the distribution of timestamps per year within this scope, slightly extended toward the 19th century to highlight the rapid sparsity that occurs before 1900. Within the chosen window, the density of facts is highly uneven: the period 2000–2015 shows a sharp peak of temporal annotations, whereas the early 20th century remains comparatively underrepresented. This imbalance poses a concrete challenge for learning systems. Models must avoid overfitting to the abundant data of recent decades while still maintaining accurate representations of earlier, sparsely documented periods. As such, the TR setting is designed to stress-test temporal generalization within a constrained but non-uniform historical horizon.

The second setting expands to the **Full temporal scope (FD) [-1000–2023]**, covering more than three millennia of human history. This broader window introduces the dual challenge of extreme sparsity and long-range temporal reasoning. As shown in Figure 4.2, the distribution is highly skewed: only isolated spikes appear in the distant past (e.g., around year 0 or 400), often reflecting singular well-documented events, whereas the density of temporal annotations rises dramatically from the Renaissance onward and be-

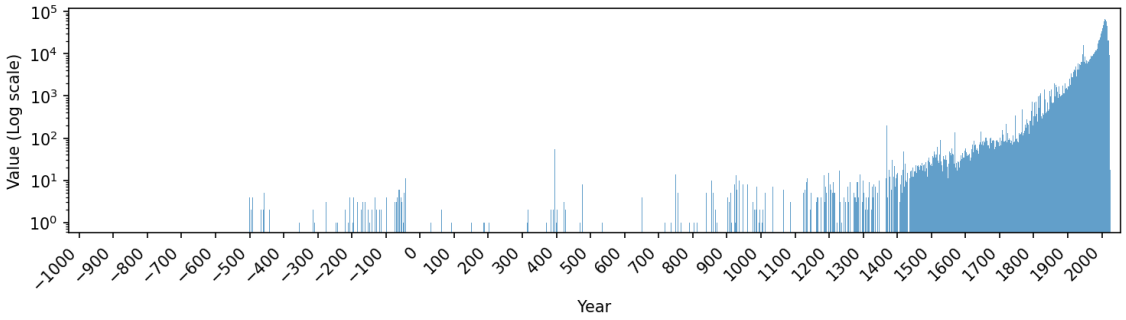


Figure 4.2: Number of timestamps per year in the full temporal scope (FD), logarithmic scale on the y-axis.

comes dominant after 1400. Importantly, even unused timestamps from sparse regions remain part of the search space for negative sampling, thereby increasing the difficulty of fact validation. Models evaluated on the FD setting must therefore be capable of handling highly imbalanced data, a great number of unique timestamps and reasoning over both dense modern intervals and scattered ancient ones.

Taken together, the TR and FD scopes provide complementary evaluation settings: the reduced scope emphasizes dense and relatively consistent modern data, while the full scope stresses long-range reasoning under extreme sparsity. This dual design ensures that models are tested both on their ability to generalize across uneven temporal distributions and to scale across vastly different time horizons.

Entity Degree.

Beyond temporal scope, the structural density of the graph also has a direct impact on model behavior. Sparse graphs provide broader coverage but limited neighborhood information, whereas dense graphs concentrate more local structure and are often more favorable to symbolic or path-based methods. To control this dimension, we applied an iterative filtering procedure based on entity degree. Specifically, we retained only those entities connected to at least N distinct neighbors. The filtering was repeated until stability, ensuring that all remaining entities satisfied the minimum connectivity threshold.

As expected in graph structures, increasing the threshold N reduces the number of retained entities according to an exponential trend (Figure 4.3). Lower values of N yield larger but sparser graphs, while higher values produce smaller yet denser graphs. This trade-off allows us to define four benchmark configurations: ***Extra-Small*** ($N = 15$) retaining only 0.15% of entities for rapid prototyping and debugging, ***Small*** ($N = 10$) with 1%, ***Medium*** ($N = 8$) with 2.7%, and ***Large*** ($N = 4$) with 15.5%. These settings enable controlled evaluation across different structural regimes, from highly condensed subgraphs to larger, more representative slices of Wikidata.

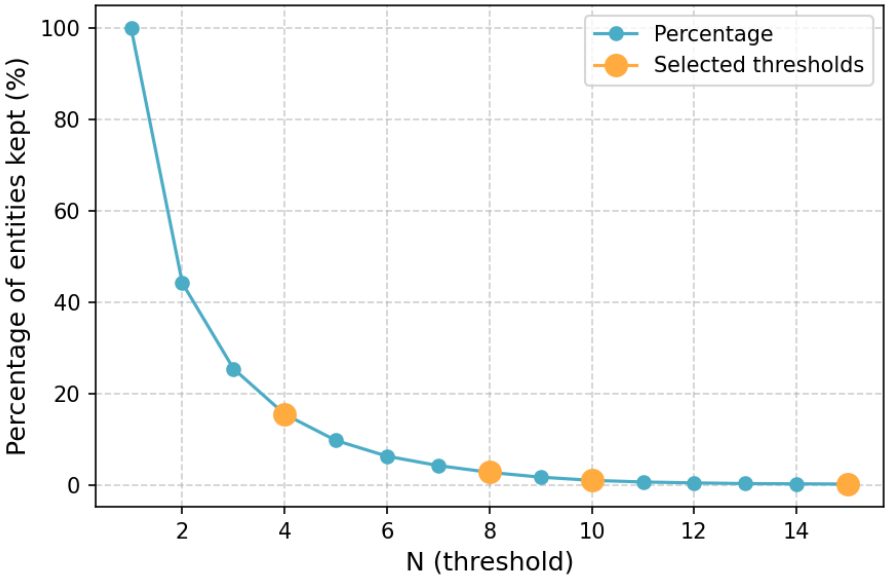


Figure 4.3: Percentage of entities retained per degree threshold N.

Temporal Precision.

A final axis of variation concerns the granularity of temporal annotations. Temporal reasoning can unfold at very different scales, from long-term historical intervals to short-lived events. To reflect this diversity, we define two complementary settings. The **Year-level (Y)** configuration aggregates timestamps at the level of calendar years. This reduces the number of distinct temporal points and increases their reuse across facts, which encourages models to reason at the macro-level over long-duration intervals such as political mandates, dynastic successions, or the lifespan of cultural artifacts.

By contrast, the **Day-Month-Year (D)** configuration preserves the full temporal resolution of Wikidata. This produces a much larger and sparser timestamp space, where most temporal points are unique or appear only a handful of times. The resulting setting is considerably more challenging: models must scale to millions of distinct temporal anchors while still capturing fine-grained phenomena such as seasonality in sports competitions, repeated election cycles, or even events tied to specific days. This high-resolution regime complements the coarser year-level view by testing not only scalability but also the precision with which models can handle temporal overlaps and short-lived facts.

Together, these two levels of temporal precision provide complementary evaluation grounds: the year-level setting stresses generalization across extended intervals, while the fine-grained day-level setting exposes models to sparsity, scalability, and the need for exact temporal discrimination.

4.4.3 . Negative Sample Generation

Because knowledge graphs typically store only true facts, constructing a benchmark for validation requires the creation of reliable negative examples. A key challenge is to design negatives that are *plausible yet incorrect*, so that models cannot dismiss them with trivial heuristics. Randomly replacing entities or relations would yield easily detectable false facts, which would not genuinely test temporal reasoning. Instead, our approach focuses exclusively on corrupting the temporal dimension of existing facts within a plausible window.

For each positive fact $f = \langle s,\ p,\ o,\ \tau \rangle$ (represented in blue on figure 4.4), we first identify the complete temporal sequence associated with the pair (s, p). This sequence defines the full set of valid intervals (in green) for that relation, ensuring that corruptions remain context-aware. We then construct a *search window* based on the lifespan of the subject s (black boundaries), extended by 10% to avoid trivial shifts (grey boundaries), and capped by the global temporal boundaries of the benchmark (e.g., [-1000, 2023] in the FD scope and [1900,2023] for the TR scope). From this window, we identify all unused intervals or *gaps*, and select one at random to generate a negative interval τ_n. The resulting negative fact is $f_n = \langle s,\ p,\ o,\ \tau_n \rangle$ in orange in our example.

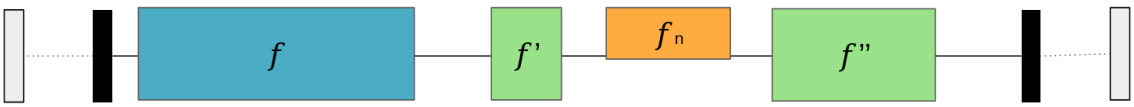


Figure 4.4: Schematic of negative sampling by temporal corruption.

This strategy ensures that negatives are realistic (e.g., shifting a presidential mandate by a few years, or placing an athlete's competition outside the period matching their weight category) but still inconsistent with the true timeline. If no gap is available because the sequence is fully covered, the corresponding positive fact is removed in order to maintain a strict 1:1 balance between positive and negative samples.

It is important to note that the difficulty of generating such negatives depends directly on the temporal scope and granularity. In dense modern periods (TR, Day-level), gaps are scarce and intervals are tightly packed, producing highly constrained negatives. In sparse historical periods (FD, Year-level), gaps are more frequent, but negatives risk being too easy unless carefully restricted by the search window. This interplay illustrates how negative sampling interacts with the other design axes of the benchmark.

While this heuristic yields challenging and contextually valid negatives, it also introduces limitations. For instance, partially overlapping corruptions may represent "semi-true" intervals, which we currently exclude. Future work could investigate graded or soft validity scores instead of strict binary labels. Nonetheless, by systematically targeting the temporal dimension, our strategy creates evaluation scenarios that require models to reason explicitly about time rather than relying on structural cues.

4.4.4 . Semantic Information Integration

To enable more sophisticated fact validation, our benchmark is enriched with semantic information, a feature often absent in existing Temporal Knowledge Graph (TKG) datasets. This enrichment reflects the reality of large-scale resources such as Wikidata, where facts rarely exist in isolation but are contextualized by ontologies, textual metadata, and datatype attributes. We integrate three complementary dimensions:

Ontological Structure (Class hierarchies).

For each entity, we retrieve its associated classes as well as the hierarchical relations that connect these classes. This ontological backbone captures the semantic organization of the graph, where subclass relations encode inheritance of meaning and constraints. For instance, if we know that `Lightweight` $\sqsubseteq$ `WeightClass` and `Boxer` $\sqsubseteq$ `Athlete`, then any fact linking a boxer to a lightweight competition must remain coherent with the semantics defined at both the class and superclass levels. Such ontological rules support consistency checks that extend beyond individual triples, allowing reasoning systems to identify semantic incoherences or violations of inherited expectations. Beyond symbolic reasoning, this hierarchical structure also benefits data driven models. Deep learning architectures that rely on neighborhood aggregation, such as RGCN [Schlichtkrull et al.(2018b)], can leverage shared semantic definitions across entities of the same class to enhance representation learning and improve generalization. This interplay between structure and learning is particularly relevant in the context of neuro symbolic approaches like TransOWL [d'Amato et al.(2021)], where ontological priors provide a structured foundation that guides and constrains the learning process.

Textual Information.

Entity labels and textual descriptions can be leveraged by language models for contextual understanding. Textual metadata plays a crucial role in disambiguation when ontology and literals alone are insufficient. For instance, the entities *Blade Runner (1982)* and *Blade Runner: The Final Cut (2007)* share the same director, genre, and production studio, but their textual descriptions specify their release years and edition types, which helps a model distinguish between them. By reintegrating such information extracted from the original sources, often omitted in existing datasets, our benchmark restores part of the contextual richness inherent to real world knowledge graphs. This inclusion encourages models to exploit semantic and contextual cues that naturally occur in authentic data, leading to reasoning processes that are closer to real usage scenarios. As a result, the evaluation environment becomes not only more realistic but also better suited to assess a model's ability to generalize beyond simplified or artificially constrained settings.

Data-Type Properties.

Static attributes of entities can be used for cross-fact validation. Consider the two facts `(Ali, weight, 90kg, [1974-01-01, 1974-12-31])` and `(Ali, championship, Heavyweight, 1974)`. A model can use the weight value to validate the second fact by checking whether it satisfies the minimum weight requirement for the "Heavyweight" class in boxing during that time period. This creates a powerful mechanism for detecting semantic inconsistencies that would remain invisible in purely structural benchmarks.

Together, these three forms of semantic information ensure that validation operates not only at the structural level of temporal triples, but also at the ontological, textual, and numerical levels that characterize real-world knowledge graphs. By supporting them explicitly, our benchmark enables both symbolic and neural models to be evaluated and compared under richer, more realistic conditions, and fosters the development of hybrid approaches that integrate multiple reasoning paradigms.

4.4.5 . Benchmark Statistics

The statistics of our benchmark, presented in Table 4.3, underscore the diversity of learning challenges encountered by time-aware models, which must accommodate widely varying configurations in the number of facts, entities, relations, and temporal parameters. The following paragraphs explain the construction of the table and discuss how each of these dimensions shapes model behavior and contributes to the overall complexity of temporal reasoning.

The table reports the number of positive facts (negative facts are generated by temporal corruption with a strict 1:1 ratio), together with the counts of unique entities, relations, and timestamps. For clarity, the notation "deg." refers to the average degree of entities (i.e., the mean number of distinct neighbors per entity), while "use" indicates the average number of times a relation or timestamp is reused across facts.

Several trends stand out. First, entity connectivity varies strongly with the configuration. In the `XS` settings, entities display high average degrees (around 50), producing dense graphs that are advantageous for symbolic or path-based approaches. By contrast, the `Large` settings reduce the average degree to about 12, resulting in sparser graphs that better reflect real-world conditions but also demand that neural models propagate information across longer paths. This contrast is visible in Figures 4.5 and 4.6.

Second, relation usage scales with dataset size. The smallest configurations involve only 40–50 relations, while `FD-Large` exceeds 400. This mirrors the heterogeneity of Wikidata and increases the difficulty of generalization: models must handle both very frequent relations and rare ones that provide little training signal. The highly skewed frequency distribution of relations, not fully captured by average usage, remains an important challenge for evaluation.

Most critically, temporal granularity introduces a sharp trade-off. In the `TR-Y` scope, only 126 distinct timestamps exist, but they are heavily reused across thousands of facts, leading to dense overlaps that stress models' ability to reason about intervals. By contrast, the `FD-D` scope contains over one million distinct timestamps, most of which occur

Scope	Prec.	Size	# Facts	# Entities (deg.)	# Relations (use)	# Temp. (use)
TR	Y	Extra Small	1 189	440 (51)	40 (283)	126 (142)
		Small	30 169	35 925 (17)	227 (1 344)	126(4 293)
		Medium	57 951	70 643 (16)	255 (2 263)	126 (8 208)
		Large	164 461	280 905 (12)	366 (4 477)	126 (22 923)
	D	Extra Small	1 105	443 (52)	41 (279)	45 072 (1)
		Small	33 318	37 898 (18)	229 (1 452)	45 072 (13)
		Medium	64 201	76 645 (17)	259 (2 494)	45 072 (26)
		Large	189 851	316 371 (12)	372 (5 079)	45 072 (75)
FD	Y	Extra Small	1 303	461 (52)	40 (298)	3 026 (6)
		Small	34 027	39 675 (17)	245 (1 383)	3 026 (199)
		Medium	63 192	77 210 (16)	279 (2 273)	3 026 (376)
		Large	182 383	314 111 (12)	408 (4 492)	3 026 (1 072)
	D	Extra Small	1 151	464 (52)	41 (295)	1 104 275 (1)
		Small	36 281	41 642 (18)	246 (1 488)	1 104 275 (1)
		Medium	70 572	83 203 (17)	282 (2 491)	1 104 275 (1)
		Large	207 823	349 995 (12)	411 (5 073)	1 104 275 (3)

Table 4.3: Statistics of the benchmark positive samples: counts of facts, unique entities (with average degree), relations (with average use), and timestamps (with average use)

only once. This creates a sparse temporal space where memorization is impossible and models must instead scale to extreme timestamp cardinality while still detecting plausible intervals.

Finally, the systematic progression from `Small` to `Large` configurations provides a controlled scaling axis. This allows researchers to begin with manageable graphs for rapid prototyping and then progressively test scalability and robustness under increasingly realistic and sparse conditions.

Taken together, these statistics highlight both the strengths and the limitations of the benchmark. On the one hand, the multi-axis design (temporal scope, entity degree, temporal precision) exposes diverse methodological strengths and weaknesses, reflecting heterogeneous real-world temporal reasoning. On the other hand, extreme configurations such as `FD-D` introduce sparsity so severe that they function more as stress-tests than as realistic scenarios. This duality is intentional: it enables both fair comparison on tractable settings and exploration of scalability limits. In the following section, we build on these statistical observations to define the temporal fact validation task and its evaluation protocol.

4.5 . Temporal Fact Validation Task

In this section, we present the preliminary results of our constraint-based validation approach introduced in Chapter 3, alongside experiments with state-of-the-art deep learning models originally designed for temporal completion. Before discussing these results, it is necessary to examine how classical temporal completion frameworks can be adapted

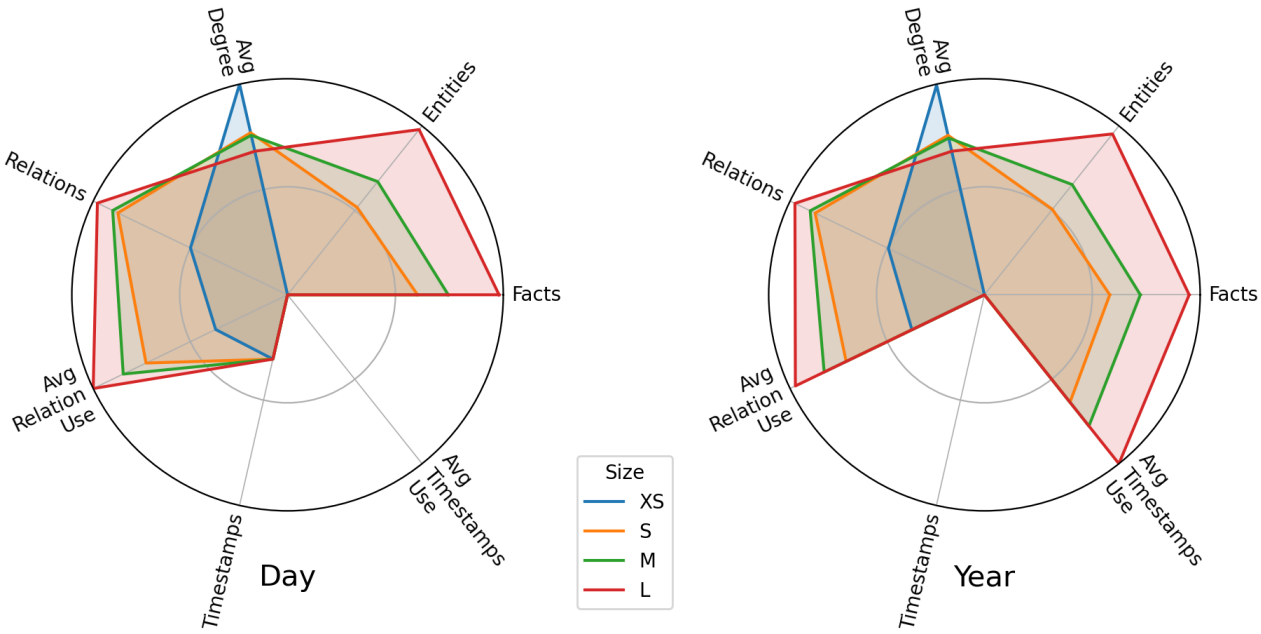


Figure 4.5: Radar plots for the reduced temporal scope (TR), comparing day-level (TR-D) and year-level (TR-Y) settings through logarithmic scale.

to the specific needs of validation. To this end, we first review the classical frameworks for static and temporal completion, then detail the methodological adaptations required to transform these ranking-oriented models into validation-oriented ones, and finally report the preliminary results obtained on our benchmark under these adaptations.

4.5.1 . Classical framework for both static and temporal completion

Static completion framework.

In the static setting, the task of completion is defined as receiving a partial triple

$$q \in \{\langle s,\ p,\ ?\rangle,\ \langle s,\ ?,\ o\rangle,\ \langle ?,\ p,\ o\rangle\}$$

and returning a ranking over the missing slot. A model defines a scoring function

$$\phi_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$$

that assigns a score to any candidate triple. The convention is that lower values are better for margin-based translational models such as TransE [Bordes et al.(2013)], while higher values are better for bilinear or deep learning families.

At inference time, given a query, the framework enumerates the candidate set for the missing position, that is $\mathcal{E}$ for head or tail prediction and $\mathcal{R}$ for relation prediction. Each candidate is then scored by ϕ_θ and the results are sorted to obtain a ranking. In practice, the *filtered protocol* is used: any triple already known as true in training, validation, or test

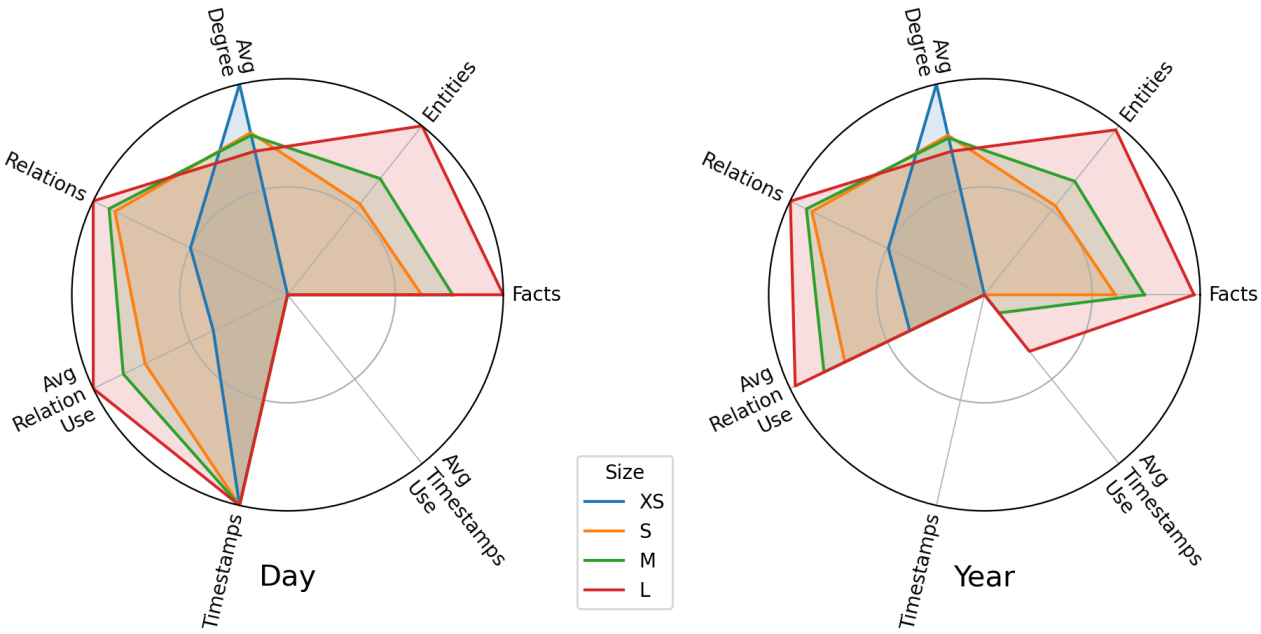


Figure 4.6: Radar plots for the full temporal scope (FD), comparing day-level (FD-D) and year-level (FD-Y) settings through logarithmic scale.

is removed from the candidate list. Evaluation is then performed using ranking metrics such as Mean Reciprocal Rank (MRR) and $hits@k$, aggregated over both head and tail queries.

The computational complexity of a naive forward pass is $\mathcal{O}(|\mathcal{E}|)$ per query for entity completion and $\mathcal{O}(|\mathcal{R}|)$ for relation completion. Importantly, the raw scores ϕ_θ are only meaningful in a *relative* sense within a single candidate set; they are not calibrated across relations or query types. As a result, the framework provides answers by ranking candidates rather than by absolute acceptance.

Temporal completion framework.

In the temporal setting, the task is extended to partial quadruples of the form

$$q \in \{\langle s,\ p,\ o\ ,\ ?\rangle,\ \langle s,\ p,\ ?,\ \tau\rangle,\ \langle s,\ ?,\ o,\ \tau\rangle,\ \langle ?,\ p,\ o,\ \tau\rangle\}.$$

The overall functioning of the framework remains similar, but the search space grows significantly depending on the definition of the temporal component τ. If τ is modeled as a single timestamp (e.g., the year 2000) within a bounded window $[0, 2023]$, the search complexity is $\mathcal{O}(|T|)$. By contrast, if τ is defined as an interval, the number of possible candidates grows quadratically to $\mathcal{O}(|T|^2)$. This quadratic growth makes interval reasoning particularly expensive at inference time.

In addition to this computational challenge, interval predictions raise non-trivial issues of validation. A predicted timestamp can be directly compared with a gold stan-

dard, but an interval prediction such as $[2000, 2005]$ is harder to evaluate against a gold interval, since it may partially overlap and therefore be simultaneously right and wrong. To address this ambiguity, several temporal frameworks such as Timemplex and TKBI [Jain et al.(2020), Cai et al.(2021)] have introduced specialized metrics that explicitly account for temporal overlap and partial correctness.

It is therefore essential to thoroughly examine these temporal metrics before evaluating models in the interval setting, as they determine how success and error are quantified in practice.

Temporal metrics.

To evaluate predictions in the interval setting, we first introduce the operator

$$size(I) = I_e - I_s,$$

where I_s and I_e denote the start and end of the interval I. Based on this operator, the classical *Intersection over Union* (IoU) can be defined as follows.

Definition 19 (IoU – Interval over Union) *Let I and I' be two intervals. The IoU value is defined as:*

$$IoU(I, I') = \left\{ \begin{array}{ccc} \dfrac{size(I \cap I')}{size(I \cup I')} & , & \textit{if } I \cap I' \neq \emptyset \\ 0 & , & \textit{otherwise} \end{array} \right\} \in [0, 1]$$

This measure provides an intuitive first estimate of similarity between two intervals. However, as soon as the intervals do not overlap, the IoU value drops to zero and loses any information about how far apart the intervals actually are. For instance, in case $\{4, 5\}$ of Figure 4.7, both intervals receive a score of zero with the ground truth G despite being at different distances from G. To address this limitation, in the work of [Rezatofighi et al.(2019)] they have proposed to use the *generalized Intersection over Union* ($gIoU$).

Definition 20 ($gIoU$ – Generalized Intersection over Union) *Let I and I' be two intervals. The $gIoU$ value is defined as:*

$$gIoU(I, I') = IoU(I, I') - \frac{size\left((I \uplus I') \backslash (I \cup I')\right)}{size(I \uplus I')}, \in [-1, 1]$$

where the operator $\uplus$ denotes the smallest single contiguous interval covering both I and I', that is $I \uplus I' = [I.s = \min(I.s, I'.s), I.e = \max(I.e, I'.e)]$.

To normalize values into $[0, 1]$, the authors have defined this derived metric :

$$gIOU'(I^G, I^P) = \frac{gIOU(I^G, I^P) + 1}{2}, \in [0, 1]$$

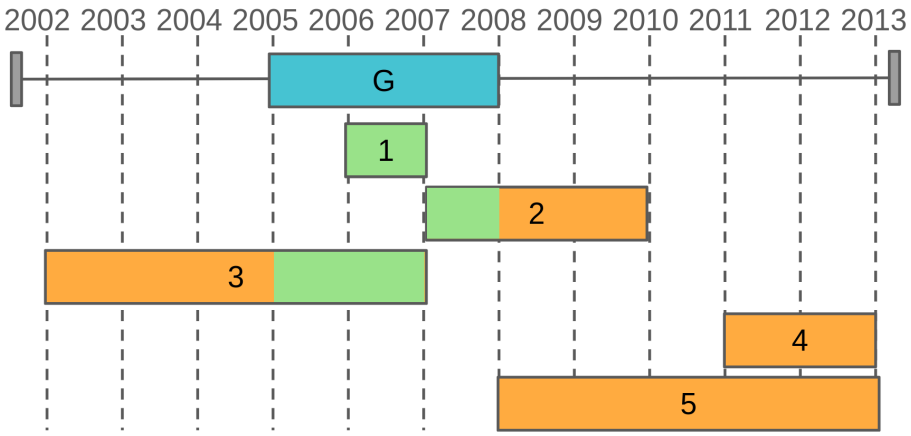


Figure 4.7: Examples of predicted intervals (numbered intervals) compared to the expected interval (G).

Although $gIoU'$ improves discriminability compared to plain IoU, it still assigns relatively high similarity to intervals with minimal overlap. As noted in [Jain et al.(2020)], even the smallest shared subinterval results in a score of at least 0.5 (see examples $\{1, 2, 3\}$ in Figure 4.7). To refine this behavior, they proposed the *affinity-enhanced IoU* ($aeIoU$).

Definition 21 ($aeIoU$ – Affinity Enhanced IoU) *Let I and I' be two intervals. The $aeIoU$ value is defined as:*

$$aeIoU(I, I') = \frac{\max\{1, size(I \cap I')\}}{size(I \uplus I')}, \in [0, 1].$$

The $aeIoU$ metric provides a finer granularity for overlapping intervals, distinguishing cases where the overlap is very small from those with larger intersections. However, as pointed out in [Cai et al.(2021)], it fails to discriminate non-overlapping intervals adequately. For example, in Table 4.4, intervals 4 and 5 receive identical scores despite interval 5 being visibly closer to the ground truth. To overcome this limitation, they introduced the *generalized affinity-enhanced IoU* ($gaeIoU$).

I^X	$IoU(I^G, I^X)$	$gIoU'(I^G, I^X)$	$aeIoU(I^G, I^X)$	$gaeIoU(I^G, I^X)$
1	$1/3$	$2/3$	$1/3$	$1/3$
2	$1/5$	$3/5$	$1/5$	$1/5$
3	$2/6$	$2/3$	$1/4$	$1/4$
4	0	$5/16$	$1/8$	$1/32$
5	0	$1/2$	$1/8$	$1/8$

Table 4.4: Values of each metric with respect to the example in Figure 4.7.

Definition 22 ($gaeIoU$ – Generalized Affinity Enhanced IoU) *Let I and I' be two intervals. The $gaeIoU$ value is defined as:*

$$gaeIOU(I^G, I^P) = \begin{cases} \dfrac{size(I^G \cap I^P)}{size(I^G \uplus I^P)} & size(I^G \cap I^P) > 0 \\ \dfrac{1}{size(I^G \uplus I^P) \cdot gap(I^G, I^P)} & \textit{otherwise} \end{cases}$$

with

$$gap(I, I') = \min(I_{max}, I'_{max}) - \max(I_{min}, I'_{min}) + 1.$$

Finally, in line with the classical evaluation practice based on $hits@k$, we also extend each temporal metric into a ranking-oriented variant. Concretely, for every query we compute the metric value over the top-k predictions and retain the best score among them, which we denote as $metric@k$. This extension is particularly useful in cases where the correct interval is not ranked first but still appears among the top candidates, ensuring that near-correct predictions are not overlooked.

In the following, we rely on these metric variants to evaluate how classical completion frameworks can be adapted to the novel task of temporal fact validation.

4.5.2 . Preliminary Experimental Analysis of the Adaptation to Temporal Validation

Having described the classical framework of temporal completion and its associated evaluation metrics, we now turn to their adaptation for the task of fact validation. The key intuition is that if a metric provides meaningful scores, then values close to 0 should indicate that a candidate fact is false, while values close to 1 should suggest that the fact is true. On this basis, we hypothesize that applying a learned threshold on the final $metric@k$ score is sufficient to infer the validity of a fact.

Definition 23 (Truthness) *We define the function $truthness$, which takes as input a $metric@k$ score and a threshold t, and outputs:*

$$truthness(metric@k, t) = \begin{cases} True, & \textit{if } metric@k > t, \\ False, & \textit{otherwise.} \end{cases}$$

The threshold t is intended to be learned with respect to each metric and each value of k. In this preliminary study, however, we adopt a simpler strategy: we observe how accuracy evolves across possible thresholds and select the best-performing t for each setting. This allows us to assess the discriminative power of different metrics without introducing an additional layer of model complexity.

For our experiments, we rely on two temporal embedding models: TNT Complex [Lacroix et al.(2020)] and Timeplex [Jain et al.(2020)]. Both operate in the complex vector space ($\mathbb{C}^d$) but differ in their scoring formulation and training strategy, reflecting two distinct philosophies of temporal modeling.

TNT Complex This model extends the static ComplEx formulation originally introduced by [Trouillon et al.(2016)]. In their subsequent work, the authors move from the static scoring function $Re(\langle s,\ p,\ \overline{o}\rangle)$ to a temporally conditioned variant $Re(\langle s,\ p,\ \overline{o},\ \tau\rangle)$, referred to as TComplex. The TNT Complex model further generalizes this idea by allowing some relations to remain static while others vary over time. In our experimental setting, however, all relations are temporally dependent, making both variants effectively equivalent.

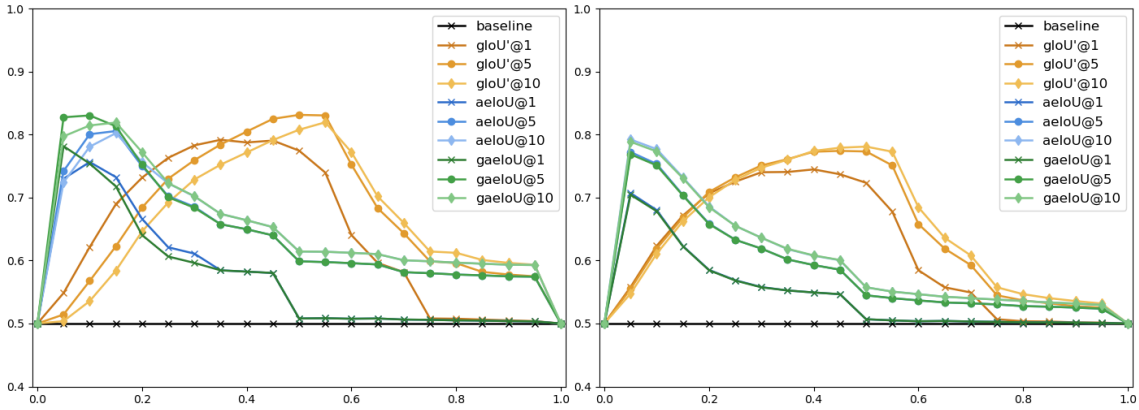
Timeplex In contrast, the Timeplex model [Jain et al.(2020)] introduces a richer scoring mechanism defined as

$$\phi(s,\ p,\ o,\ \tau) = \langle s, p^{SO}, \overline{o}\rangle + \alpha\langle s, p^{ST}, \overline{\tau}\rangle + \beta\langle o, p^{OT}, \overline{\tau}\rangle + \gamma\langle s, o, \overline{\tau}\rangle.$$

This formulation highlights the main distinction with TNT Complex: the parameterization of multiple interaction matrices, which provides greater modeling flexibility at the cost of increased training complexity. Beyond this base variant, the original article also introduces soft temporal constraints in the loss function $\mathcal{L}$, namely **Relation Recurrence**, which limits the number of times a property can occur (for instance, birth once or Olympic Games every four years); **Ordering Between Relations**, enforcing logical sequences such as being born before dying; and **Time Gaps Between Relations**, constraining temporal distances (for example, the average lifespan between birth and death). These constraints guide the model toward more coherent temporal patterns and are particularly beneficial when learning from sparse data, as they allow underrepresented relations to benefit from structured temporal knowledge.

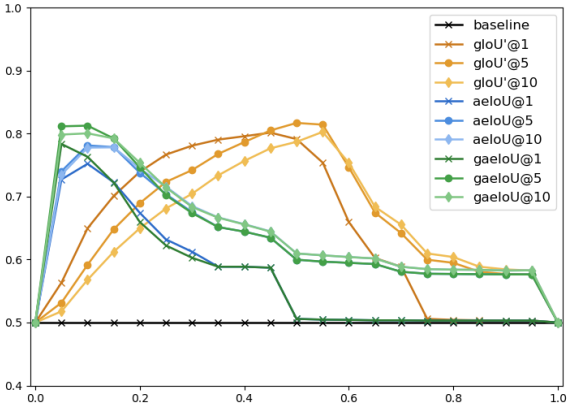
These models are trained on the four available settings of size `XS`, and their predictions are evaluated on the corresponding validation sets. For each metric, we consider the three commonly used values of $k \in \{1, 5, 10\}$, as these are standard in the literature on knowledge graph completion.

On the figures 4.8 and 4.9, we can observe the different curves for each $metrics@K$ respectively for the novel Timeplex and for the model TNT Complex. Even though we have 4 kind of settings per size, we were not able to perform the framework on *Day Full Data* as we were out of memory for this one. However, for the rest of the setting we can observe that we can achieve at most 83% for the Year Time Reduced setting for Timeplex and close results for the other two settings. On the other hand, while TNT Complex seems to have similar performance although slightly worse, it seems that for the setting Year Full Data the model is not able to learn anything and obtain only a random behavior. Hence our adaptation seems to be sucessful for the both models on restricted settings but when working with a large time window (Full data) it seems that the adaption is not able to achieve meaningful score for the second model although the first model is still able to get good results. Overall, our adaptation proves to be successful, even though some models are more affected by particular settings. Furthermore, it is very interesting to observe the behavior of each metric. Indeed, the $aeIoU$ (blue gradient) and the $gaeIoU$ (green gradient) share a very close behavior with a strong sensibility to very small value



(a) Setting : Year Time Reduced

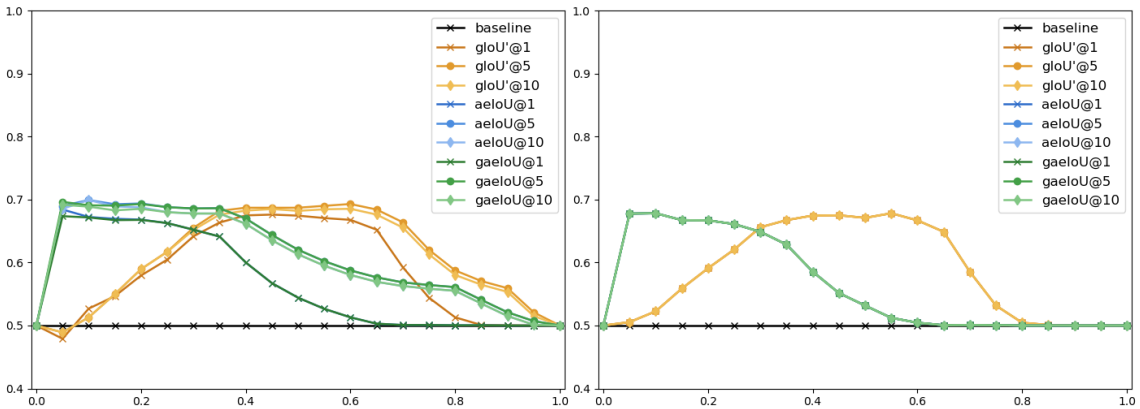
(b) Setting : Day Time Reduced



(c) Setting : Year Full Data

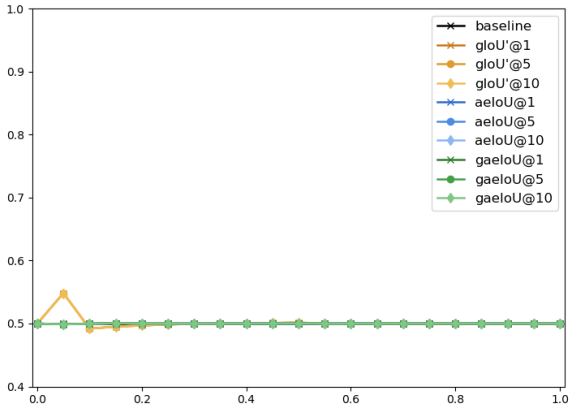
Figure 4.8: Accuracy of Timplex model with respect to the threshold value for the different settings of Extra Small.

which is closely related to their definition as this metric can yield values very close to zero for non overlapping intervals. However, we still remain good results until 0.5 where these metrics seems to suffer the most as the threshold is to strict and starts to remove positive facts from the positive set. On the other hand, the $gIoU'$ metric seems to be much more centered around 0.5 which again seems to be in expectation with its definition as it is the smallest value where we have an overlapping with the gold interval. We also observe a sharp descent after crossing the 0.6 threshold. Furthermore, it is interesting to observe that we do not go below 0.5 accuracy which is the random baseline here (in black), except for the second model in which for early threshold for the Year Time Reduced we can deep slightly below the random range. Finally, we can have a deeper look for each possible K, if we look at each metrics the worst behavior possible seems to be with $K = 1$ with scores always lower than the other values. On the other hand, it seems that $K = 5$ or $K = 10$ have very close results and depending on the temporal setting one can perform slightly better than the other. This behavior is observed for each metric which is very interesting.



(a) Setting : Year Time Reduced

(b) Setting : Day Time Reduced



(c) Setting : Year Full Data

Figure 4.9: Accuracy of TNT Complex model with respect to the threshold value for the different settings of Extra Small.

Figures 4.8 and 4.9 report the evolution of the different $metric@K$ values with respect to the threshold value for the models Timeplex and TNT Complex, respectively. Each experiment was run under four temporal settings, although the *Day Full Data* configuration could not be executed due to memory limitations. For the remaining setups, Timeplex achieves its best performance of approximately 83% on the *Year Time Reduced* setting, with comparable results in the other reduced configurations. By contrast, TNT Complex displays slightly lower overall accuracy and, more importantly, fails to learn meaningful patterns in the *Year Full Data* setting, where its predictions are indistinguishable from random behavior. These observations suggest that our adaptation is effective for both models on reduced temporal windows, but that some architectures, such as TNT Complex, are unable to scale to larger time spans, whereas Timeplex remains competitive.

Beyond global accuracy, the figures also highlight differences in the behavior of the evaluation metrics. The $aeIoU$ (blue gradient) and $gaeIoU$ (green gradient) curves exhibit

very similar shapes, with strong sensitivity to small values. This behavior is consistent with their definitions, since both metrics yield values close to zero when intervals do not overlap. Nevertheless, they maintain relatively high performance until the threshold reaches 0.5, beyond which the stricter criterion discards an increasing number of correct predictions. In contrast, the $gIoU'$ metric tends to concentrate values around 0.5, which again aligns with its definition, as this is the minimal score assigned to overlapping intervals. A sharp decrease is observed beyond a threshold of 0.6, indicating the limits of this measure. Interestingly, model accuracy rarely falls below the random baseline (shown in black at 0.5), except in the case of TNT Complex under the *Year Time Reduced* setting, where performance dips slightly below randomness at low thresholds.

Finally, varying the parameter K provides further insights. Across all metrics, the worst performance is consistently obtained for $K = 1$, while larger values ($K = 5$ and $K = 10$) produce substantially better and almost indistinguishable results. Depending on the setting, either $K = 5$ or $K = 10$ can marginally outperform the other, but both clearly dominate the single-prediction scenario. This consistent behavior across metrics underlines the importance of allowing multiple candidate predictions when evaluating temporal fact validation.

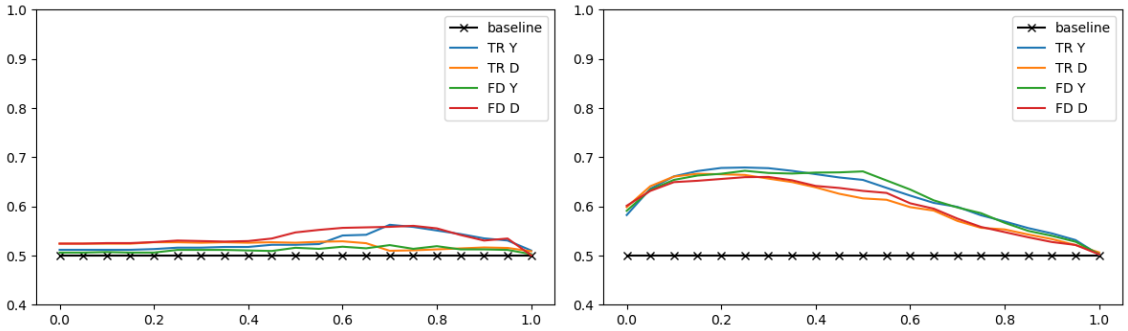
Overall, these results highlight both the promise and the limitations of adapting completion frameworks to validation: while some models scale effectively and maintain high performance under reduced settings, others degrade when exposed to larger temporal windows. At the same time, the choice of evaluation metric and the parameter K prove decisive in shaping our interpretation of model behavior.

Symbolic Adaptation

To ensure a fair comparison across all evaluated frameworks, a minor adaptation was introduced for the symbolic approaches described in Chapter 3. In this revised setting, we do not impose a fixed validation threshold. Instead, for each fact under verification, the framework computes the proportion of satisfied temporal constraints. If this proportion exceeds a threshold t, the fact is inferred as valid; otherwise, it is classified as invalid. This procedure differs from our previous proposal, where we assumed that an optimal threshold could be directly related to the exception rate defined in Definition 15.

The modified approach was applied to the Extra Small configuration, and the results are reported in Figure 4.10. Two experimental settings were considered: one where constraints are defined only at the level of relations (Figure 4.10a), and another where constraints are specified for both properties and properties with their associated values as described in definition 17 (reported on Figure 4.10b). These experiments were conducted in a purely symbolic context, without any neuro-symbolic component, in order to reduce the number of hyperparameters and to isolate the intrinsic behavior of the rule-based validation mechanism.

Unlike the sharp performance variations observed in deep learning models, the symbolic framework exhibits a smoother evolution of accuracy with respect to the threshold parameter. Nonetheless, distinct calibration trends can still be identified. In the case of



(a) Setting : Constraint on Relations only

(b) Setting : Constraint on Relation and Relation specified

Figure 4.10: Accuracy of the symbolic framework with respect to the threshold value for the different settings of Extra Small.

relation-only constraints (Figure 4.10a), the framework achieves its best results for thresholds around $t = 0.6$. Conversely, when relation-value constraints are incorporated (Figure 4.10b), the optimal performance is reached for lower thresholds, typically between $t = 0.2$ and $t = 0.6$. These results suggest that finer-grained constraints allow the framework to make reliable validation decisions with less stringent satisfaction ratios.

4.5.3 . Comparative Evaluation of Symbolic and Adapted Frameworks

Now that we have introduced the adaptation of the Temporal Completion framework to the Temporal Validation task and the description of our Symbolic Approach in the chapter 3. We can now apply these models on the different sizes of our setting.
The results reveal a clear accuracy versus scalability trade off across models and dataset sizes. On the smallest size XS, deep models reach the highest accuracies but do not scale to larger sizes without relaxing temporal detail. Timeplex attains 83% on TR Y, drops modestly to 79% on TR D where day level precision increases the number of temporal parameters while reducing samples per cell, and still achieves 82% on FD Y, but runs out of memory on FD D when both day precision and a full temporal scope jointly expand the parameter space. On size S, we observe that with less temporally intensive settings the models can still produce results: Timeplex reaches 74% on TR Y and 74% on FD Y, yet remains out of memory on TR D and FD D, while TNTComplex falls to 53% on TR Y and is out of memory on the other three settings, indicating weak robustness as the graph grows even slightly. In contrast, the symbolic variant that uses only properties remains close to the random baseline but scales to larger sizes, with XS accuracies between 52% and 56%, and stable values between 54% and 57% on S, M, and L across all columns. Enriching symbolic constraints with values markedly improves accuracy on XS, from 56% to 68% on TR Y, from 53% to 67% on TR D, from 52% to 67% on FD Y, and from 56% to 66% on FD D, although this configuration exceeds the time budget on S, M, and L. Notably, on XS FD Y the symbolic model with values reaches 67%, outperforming TNTComplex at 57%, and it stays within three points of TNTComplex on TR Y 68% versus 70% and matches it

Size	Model	TR Y	TR D	FD Y	FD D
XS	Symbolic (Only R) [fig. 4.10a]	56%	53%	52%	56%
	Symbolic (R and RxV) [fig. 4.10b]	68%	67%	67%	66%
	Timeplex [fig. 4.8]	83%	79%	82%	OOM
	TNTComplex [fig. 4.9]	70%	67%	57%	OOM
S	Symbolic (Only R)	57%	53%	57%	54%
	Symbolic (R and RxV)	OOT	OOT	OOT	OOT
	Timeplex	74%	OOM	74%	OOM
	TNTComplex	53%	OOM	OOM	OOM
M	Symbolic (Only R)	56%	54%	56%	54%
	Symbolic (R and RxV)	OOT	OOT	OOT	OOT
	Timeplex	OOM	OOM	OOM	OOM
	TNTComplex	OOM	OOM	OOM	OOM
L	Symbolic (Only R)	57%	54%	57%	54%
	Symbolic (R and RxV)	OOT	OOT	OOT	OOT
	Timeplex	OOM	OOM	OOM	OOM
	TNTComplex	OOM	OOM	OOM	OOM

Table 4.5: Accuracy for the best threshold on the set with the accuracy metric (OOM means out of memory and OOT means out of time)

on TR D at 67%. Overall, deep models offer the best accuracy on XS and can retain some performance on S under year precision, but lack memory scalability under day precision and larger scopes. The properties only symbolic baseline offers stable scalability with moderate accuracy, and the value enriched symbolic approach closes much of the gap to deep models on XS yet requires algorithmic and search space improvements to meet time constraints on larger sizes. We therefore identify two engineering priorities: pruning and ranking of candidate value constraints to limit the number of instantiated checks, and faster validation routines to lift the out of time barrier without sacrificing the accuracy gains observed on XS.

4.6 . Conclusion

Validating facts with respect to time remains a fundamental yet underexplored challenge for knowledge-infused AI. In this chapter, we moved beyond the traditional focus on knowledge graph completion and addressed the task of temporal fact validation. This required rethinking both the evaluation protocols and the modeling assumptions inherited from completion frameworks.

Our study underscores two central limitations. First, although classical metrics for knowledge graph completion perform well on ranking based tasks, they are poorly matched to validation scenarios that require explicit assessment of probability calibration and temporal overlap. Second, state of the art deep learning architectures show clear weaknesses when confronted with real world temporal structures, particularly in settings

with long intervals or wide temporal windows. These findings indicate that success in completion tasks does not necessarily translate into reliable temporal validation.

To confront these challenges, we introduced a dedicated benchmark that systematically generates temporal corruptions and accommodates datasets with high timestamp cardinality. This benchmark is designed to expose the trade-offs between accuracy, calibration, and scalability, thereby forcing models to confront issues of temporal granularity and parameter efficiency.

By releasing this resource to the community, our goal is to encourage the development of models that are not only performant but also temporally grounded and robust across diverse settings. In doing so, we hope to shift the field of knowledge graph research toward approaches that treat temporal validity as a first-class requirement, ultimately leading to more reliable reasoning systems for dynamic, time-aware knowledge.

4.7 . Future Work

Several directions naturally extend this chapter by enriching the proposed benchmark and making it more representative of real-world of temporal fact validation.

4.7.1 . Overlapping negative sampling

The behavior of deep learning approaches suggests that results are strongly influenced by the negative sampling strategy used in the benchmark. In the current version, we adopted a strict non-overlapping protocol, where negative samples are constructed to be entirely disjoint from the gold interval. This design choice simplifies evaluation but also explains why we observed a sharp drop in accuracy around the 0.5 threshold, which coincides with the region where partially overlapping intervals would naturally occur. Introducing overlapping negatives would create a more realistic setting in which models must distinguish between fully incorrect facts and partially valid ones. Such an extension would push models toward a finer understanding of temporal validity, though it raises important challenges: depending on the semantics of a relation, a partially overlapping interval may represent a fact that is still acceptable or one that should be considered entirely false. For instance, in the relation $residesIn$, a partial temporal overlap could remain acceptable, as the transition between residences is often gradual and inconsistently recorded across sources. In contrast, for $headOfStates$, even a minimal overlap would render the fact false, since the office can only be held by a single individual at any given time. Addressing this ambiguity would be crucial for future iterations of the benchmark.

4.7.2 . Multi-scale temporal granularity

The current benchmark relies on a single granularity for temporal annotations, typically expressed at the level of years or days. However, many real-world knowledge graphs contain facts that span heterogeneous granularities, ranging from coarse (centuries or decades) to fine (minutes or seconds). Extending the benchmark to support multiple granularities simultaneously would allow us to study how models handle temporal ab-

straction and refinement, and whether their predictions remain consistent when switching between levels of detail. This would also enable the evaluation of models capable of operating across different scales of data size, a capacity that is increasingly important for applications such as historical analysis or biomedical event tracking.

4.7.3 . Uncertain and fuzzy temporal annotations

Another promising direction concerns the integration of uncertain or fuzzy temporal boundaries, which are frequent in practice but absent from the current benchmark. Many historical or scientific facts cannot be localized to a precise date or interval but are instead annotated with approximate periods, confidence ranges, or qualitative labels such as "circa 1800", indicating around the year 1800. Supporting such representations would significantly increase the realism of the benchmark and open the way for evaluating models under uncertainty. It would also enable the design of metrics that explicitly measure robustness to temporal vagueness, thereby aligning evaluation more closely with the conditions faced in real-world knowledge graph applications.

4.7.4 . Improving the symbolic approach through semantic generalization

Although not directly related to the benchmark contribution, an important limitation of the symbolic approach introduced in Chapter 3, and more specifically its Value-Specified Constraints, was revealed through the real-world configuration of the benchmark. In the current benchmark setting, specifying constraints by their exact values significantly increases the number of constraints to be checked for each fact, which in turn affects computational efficiency. A promising improvement would be to introduce a semantic abstraction mechanism, where constraints are expressed not by individual values but by the semantic class they belong to. For instance, a constraint such as $(memberOf,\ NATO)$ could be generalized to $(memberOf,\ MilitaryAlliance)$. Such abstraction would drastically reduce the number of instantiated constraints while preserving their semantic relevance, thereby improving scalability. Moreover, the current implementation does not exploit the class hierarchy to propagate constraints across sub- and superclasses. Integrating such hierarchical reasoning could further enhance the framework's efficiency and expressiveness, leading to a more semantically grounded and computationally efficient symbolic model.

4.7.5 . Diverse datasets

Finally, the benchmark is evaluated on a limited number of temporal knowledge graphs. While sufficient to highlight methodological challenges, this restriction limits the scope of conclusions regarding model generalization. A natural extension is to incorporate datasets from different domains and contexts, such as biomedical graphs with highly irregular temporal patterns, event-centric graphs with dense timestamp annotations, or historical graphs characterized by long and uncertain intervals. The inclusion of such heterogeneous sources would not only broaden the coverage of temporal validation but also reveal domain-specific weaknesses in existing approaches, thereby guiding the development of more robust and versatile validation methods.

Chapter 5

Studying Key Transferability for Scalable Data Linking Between Knowledge Graphs

The work presented in this chapter has been in collaboration with ***Fatiha Saïs, Joe Raad, and Gianluca Quercini****. Part of this work has been published in :*

SOULARD Thibaut, SAÏS Fatiha and RAAD, Joe.
Étude de transférabilité des clés pour le liage de données entre graphes de connaissances*.*
in Revue Ouverte en Intelligence Artificielle. ***Under Publication***

SOULARD Thibaut, SAÏS Fatiha, RAAD Joe, and QUERCINI Gianluca.
Efficient Key-based Data Linking Through Key Transfer Between Knowledge Graphs*.*
In : Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing. 2024. p. 1642-1649.

SOULARD Thibaut, SAÏS Fatiha, RAAD Joe, and QUERCINI Gianluca.
Étude de transférabilité des clés pour le liage de données entre graphes de connaissances*.*
In : 34es Journées francophones d'Ingénierie des Connaissances (IC 2023)@ Plate-Forme Intelligence Artificielle (PFIA) 2023.

The previous chapters focused on temporal reasoning, addressing how facts evolve over time and how models can validate their temporal consistency through explainable constraints and benchmarking resources. Having established methods for understanding when a fact remains valid, we now turn to the complementary question of how knowledge itself can be improved and propagated across graphs. This chapter therefore shifts the focus from temporal dynamics within a single Knowledge Graph to structural interoperability between multiple ones. Specifically, it investigates the transfer of keys discovered

in a source static Knowledge Graph to a target static Knowledge Graph via property alignments, with the dual objective of accelerating identity resolution and enriching entity descriptions so that both learning-based and symbolic frameworks can benefit from denser, less noisy data. In the longer term, this transfer paradigm could also be extended to the temporal dimension, enabling the propagation of discovered temporal constraints across graphs and thereby supporting cross-graph validation of temporal consistency. We formalize the transfer setting on static graphs, present a rewriting-and-validation pipeline, and evaluate its effect on runtime and interlinking quality under heterogeneous and harmonized vocabularies.

5.1 . Introduction

In the broader effort to improve the completeness, quality, and interoperability of Knowledge Graphs, one recurring challenge is establishing reliable identity links between entities described across different datasets. Such links enable the reconciliation of complementary descriptions of the same real-world objects, the propagation of knowledge across sources, and ultimately support the FAIR principles of data reuse in open and industrial contexts. Without such links, knowledge integration remains fragmented, limiting the impact of KGs in downstream applications such as question answering, recommendation, or scientific discovery.

Among the various strategies for entity alignment [Noy et al.(2017), Christophides et al.(2020)], key-based linking offers a particularly precise and explainable mechanism. By leveraging discriminative sets of properties, keys make it possible to differentiate entities in a way that is both transparent and verifiable. Compared to similarity-based or embedding-based methods, which may behave like black boxes, key-based approaches provide guarantees that facilitate debugging, validation, and trust in the generated links. These characteristics are crucial for domains where accountability and interpretability are required, such as digital humanities or biomedical knowledge integration.

However, the benefits of key-based methods are tempered by significant computational challenges. Discovering keys is typically a resource-intensive task, especially on large graphs with thousands of properties and millions of triples (e.g., Wikidata). Moreover, in current practice, key discovery is performed independently for each dataset, even when similar or equivalent structures are present. This redundancy leads to duplicated effort, wasted computational resources, and a lack of systematic reuse. More importantly, it raises a fundamental research question: under which conditions can keys discovered in one KG be reused in another, and what are the limitations of such transfer?

This issue is closely connected to ontology alignment. While property and class correspondences are well studied and form the basis of many linking approaches, the question of how these alignments can support the transfer of higher-level constructs such as keys remains unexplored. If keys can be successfully reused across graphs through alignment, they could serve as robust building blocks for scalable and explainable alignment

pipelines, reducing the overhead of systematic rediscovery.

This chapter addresses, to the best of our knowledge, the first systematic investigation of key transferability across knowledge graphs. We introduce a formal framework for key transfer, analyze its theoretical properties, and conduct experiments on real-world datasets such as DBpedia, Wikidata, and YAGO. The contributions of this study are twofold: (i) a better understanding of the conditions under which keys remain valid and discriminative when applied to a new graph, and (ii) empirical evidence that key reuse can significantly reduce computational costs while preserving alignment quality. Beyond efficiency, this work highlights how transferable keys could act as an intermediate layer between low-level ontology alignments and high-level semantic integration, bridging a gap in current data linking approaches.

The remainder of this chapter is organized as follows: Section 5.2 reviews related work on key discovery and data linking. Section 5.3 introduces the formal definitions and terminology used throughout this work. In Section 5.4, we describe the methodology developed to assess the feasibility and effectiveness of key transfer. Section 5.5 presents the experimental setup, evaluation metrics, and a discussion of the results. Finally, Section 5.6 concludes with a summary of findings while Section 5.7 offers directions for future work.

5.2 . Related Work

When aiming to combine or exploit data from different sources, it becomes crucial to detect and reconcile descriptions that refer to the same real-world entity (e.g., the same person, book, or gene). This task, known as *data linking*, is fundamental to ensuring the completeness and interoperability of knowledge graphs (KGs).

A wide range of techniques have been proposed in the literature to address this problem. Some rely on similarity measures [Ngomo and Auer(2011)], others use machine learning approaches [Mudgal et al.(2018), Fanourakis et al.(2022)], or leverage domain knowledge expressed in ontologies [Saïs et al.(2009), Al-Bakri et al.(2015)].

One of the main challenges in data linking lies in the heterogeneity of data, which makes entity comparison difficult, particularly when modeling conventions, schemas, or vocabularies differ between graphs. Another major challenge concerns the computational cost of comparing entity descriptions. Indeed, data linking typically involves pairwise comparisons of property-value pairs from two graphs G_1 and G_2, resulting in a search space of size $n_1 \times n_2$, where n_1 and n_2 are the number of instances in each graph. This task becomes even more complex when properties are multi-valued or inconsistently used across datasets.

In this context, *keys* play an essential role in guiding the discovery of identity links. A key is a minimal set of properties that uniquely identifies an entity within a graph. However, manually defining such keys is generally impractical at web scale due to the volume of data and schema diversity. Consequently, several methods have been developed to automatically discover keys, both in relational databases [Sismanis et al.(2006)] and in knowl-

edge graphs [Symeonidou et al.(2014), Symeonidou et al.(2017), Atencia et al.(2020)].

A straightforward approach to using keys for data linking between G_1 and G_2 involves discovering keys independently in each graph, then intersecting or merging them to identify common structures [Pernelle et al.(2013)]. However, this process is often computationally expensive and may yield few or no shared keys. An alternative solution is provided by Linkkey [Atencia et al.(2020)], which jointly analyzes both graphs and relies on schema alignments to discover *link keys*, keys that are simultaneously valid in both datasets. While this improves precision, it still requires access to both graphs and incurs the full cost of key discovery.

To reduce the data and computation required for key-based linking, we investigate in this work the problem of *key transferability*, whether keys discovered in a source graph G_1 can be reused in a target graph G_2. More precisely, we aim to determine under what conditions a set of keys mined from G_1 remains valid and useful for entity linking in G_2. If such transfer is possible, it would eliminate the need to repeat key discovery on every new graph, thereby saving time and resources, and potentially simplifying the overall linking pipeline.

To the best of our knowledge, the question of key transferability has not yet been addressed in the literature. This chapter proposes to fill that gap by introducing a formal framework and an experimental evaluation of key reuse across heterogeneous knowledge graphs.

5.3 . Definitions and Notions

This section introduces the key concepts and formal definitions used throughout our approach for studying key transferability between knowledge graphs.

KGs may vary in vocabulary, granularity, and schema, which leads to challenges in interlinking them. A key step in enabling data interoperability is to align the properties used in different KGs.

Definition 24 (Property Alignment) *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two knowledge graphs. A property alignment between two properties $p_1 \in \mathcal{G}_1$ and $p_2 \in \mathcal{G}_2$ is a semantic correspondence indicating that p_1 and p_2 are equivalent. We denote this by $p_1 \equiv p_2$.*

A described in the definition 10 a key is a set of properties that uniquely identifies each instance of a class. In the context of knowledge graphs, keys can be used to detect identity links (`owl:sameAs`) between entity descriptions.

If a given set of properties is declared as a key for a class, failure to satisfy key uniqueness within a graph may indicate erroneous or duplicate data. Conversely, when comparing instances from two distinct KGs, the satisfaction of a key by both instances suggests that they may refer to the same real-world entity. Therefore, every pair of instances that shares the same values for all properties in a key can be treated as a candidate for identity linking.

Several key semantics have been proposed in the Semantic Web literature (see [Atencia et al.(2014)] for a theoretical and experimental comparison). These semantics differ mainly in how they treat multi-valued properties and missing values. In this work, we deliberately focus on the *S-key* semantics, which corresponds to the semantics of the `owl:hasKey` constructor[1]. The rationale behind this choice lies in the inherent incompleteness of knowledge graphs: we cannot expect two graphs to provide exactly the same set of values for a given entity, since one source may be more precise or richer than another. In contrast, the *F-key* semantics requires that every value in the property sets must be present on both sides for two instances to be considered equal. While theoretically sound, such a strict requirement often makes alignment more fragile and computationally harder to achieve in practice. By adopting S-keys, we allow the linking process to be more tolerant to heterogeneity and partial descriptions, as two instances are considered equal as soon as they share at least one common value for each property in the key.

For example, in DBpedia a person may be described with the triple `birthPlace: Paris`, whereas in Wikidata the same entity may be associated with the more precise information `place of birth` ($P19$)`: Paris` ($Q90$)`, 14th arrondissement of Paris` ($Q187153$). Under F-key semantics, these two descriptions would fail to match since the sets of values are not identical. In contrast, S-keys allow us to consider these instances as candidates for alignment, as they share at least one common value for the property `birthPlace`. This tolerance makes S-keys better suited for heterogeneous and incomplete knowledge graphs.

Definition 25 (Semantics of an S-Key) *Let C be a class[2] and let $\{p_1, \ldots, p_n\}$ be a set of properties. This set is an* S-key *for C if the following first-order logic formula holds:*

$$\forall x \forall y \forall z_1 \ldots z_n \left(C(x) \wedge C(y) \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow x = y \right)$$

Declaring that $\{p_1, \ldots, p_n\}$ is an *S-key* for a class C is denoted by *S-key*$(C, (p_1, \ldots, p_n))$. We use $prop(k)$ to denote the set of properties forming the key k.

Definition 26 (Key Instantiation) *Let k be an* S-key$(C, (p_1, \ldots, p_n))$ *for a class C in a knowledge graph $\mathcal{G}$. The instantiation of k for an instance i of class C in $\mathcal{G}$ is defined as the set of value tuples:*

$$\pi(k, i) = \{(v_1, \ldots, v_n) \mid \{(i, p_1, v_1), \ldots, (i, p_n, v_n)\} \subseteq \mathcal{G}\}$$

For example, let us consider a key $k = \{rdfs{:}label,\ wdt{:}P17\,(Country)\}$ for the class $C = City$ and an instance $i = Paris$. In the knowledge graph, the following triples may be present:

[1] https://www.w3.org/TR/owl2-direct-semantics/

[2] This may also be a class expression in OWL2: https://www.w3.org/TR/owl2-direct-semantics/#Class_Expressions

- $\langle Paris,\ rdfs{:}label,\ "Paris"\rangle$
- $\langle Paris,\ wdt{:}P17$ (country)$,\ wd{:}Q142$ (France)$\rangle$
- $\langle Paris,\ wdt{:}P17$ (country)$,\ wd{:}Q70972$ (Kingdom of France)$\rangle$

In this case, two different instantiations exist:

$$\pi(k,i) = \{("Paris", wd{:}Q142)\}, \quad \pi(k,i) = \{("Paris", wd{:}Q70972)\}.$$

5.4 . Key Transfer Methodology

To link the instances of a graph G_1 with those of a graph G_2, the traditional approach to key-based data linking first consists in discovering an initial set of keys K_1 in graph G_1 and another set K_2 in graph G_2, followed by applying a key merging procedure. This merging can either be a simple intersection of the discovered keys in both graphs, up to rewriting over equivalent properties, or a more sophisticated procedure. For instance, as proposed in [Pernelle et al.(2013)], it may involve computing the Cartesian products of property sets appearing in K_1 and K_2, retaining only the minimal keys (i.e., a key k_1 such that there is no key k_2 with $prop(k_2) \subset prop(k_1)$).

When the RDF data graphs are described using different ontologies, property alignments (see Definition 24) are needed to rewrite the keys from one graph in terms of the properties aligned in the other (see Section 5.4.1 for the alignment procedure and the definition of key rewriting).

The keys resulting from the merging process are then considered valid in both graphs and can therefore be used to link their respective instances. However, when the graphs are large or contain many properties, the computational cost of key discovery may become prohibitively high, sometimes requiring several days. To address this, we experimentally investigate the possibility of reducing this cost by transferring keys discovered in one graph to another. For this, we rely on key quality measures, namely support and discriminability, defined below. These metrics help evaluate to what extent a key, when transferred, maintains its uniqueness and identity-linking potential.

5.4.1 . Property Alignment

To compute alignments between properties of two RDF graphs, we rely on the ontologies that govern those graphs and follow a semantic matching procedure adapted from [Soulard(2022)]. The method exploits both terminological and conceptual signals that are directly available in the ontologies, namely labels, descriptions, property types, domains, ranges, and class level metadata. Below, we provide a concise step by step description.

Step 1. Data extraction and normalization. For each ontology, we extract for every property its label and description when available, its type `owl:DatatypeProperty` or `owl:ObjectProperty`, and its declared domain and range. For classes referenced in

domains and ranges, we also extract their labels and descriptions. Text is lowercased and stripped from markup, while numeric and date literals are kept unchanged to preserve meaning.

Step 2. Text embedding for labels and descriptions. We encode labels and descriptions with a transformer based sentence encoder such as BERT [Devlin et al.(2018)]. When both a label and a description exist, we obtain two vectors per entity and keep them separate in order to weight them later. We do not remove stop words since the transformer captures context that can be essential for disambiguation.

Step 3. Class level similarity for domains and ranges. We compute cosine similarity between class label vectors and, when present, between class description vectors. The class similarity is the average of the available signals. If a class lacks a description, only label similarity is used. These scores are cached for later reuse when comparing properties that reference these classes.

Step 4. Property level similarity within type partitions. We compare only properties of the same type. For a candidate pair P in ontology O_1 and P' in ontology O_2, we compute

$$\mathrm{Sim}(P, P') \;=\; \mathrm{avg}\Big(\mathrm{Sim}_{\ell},\, \mathrm{Sim}_{d},\, \mathrm{Sim}_{\mathrm{dom}},\, \mathrm{Sim}_{\mathrm{rng}}\Big),$$

where Sim_{ℓ} and Sim_{d} are cosine similarities between label embeddings and description embeddings, and $\mathrm{Sim}_{\mathrm{dom}}$ and $\mathrm{Sim}_{\mathrm{rng}}$ are the class similarities for domain and range retrieved from Step 3. The average is taken over the signals that are present. This design allows graceful degradation when some ontology fields are missing.

Step 5. Conservative selection with global and relative thresholds. From all cross ontology pairs we first discard those with $\mathrm{Sim}(P, P') < \gamma$ using a global threshold $\gamma \in [0, 1]$. We then iterate pairs in decreasing order of similarity and build alignments with a default one to one policy. To accommodate structural heterogeneity and benign redundancy, we allow one to many alignments controlled by a relative margin ρ: if P is aligned to P'_1 with score s_1, any other P'_k with score $s_k \geq s_1 - \rho$ is also kept. The same rule is applied symmetrically when starting from P'.

Step 6. Output and provenance. For each retained pair we assert `owl:equivalentProperty` together with the final similarity score and record the evidence used to compute it, namely which textual and class signals contributed to the decision. This provenance supports inspection and reuse in downstream tasks.

Remarks.

- (i) Partitioning by property type avoids invalid matches between datatype and object properties.
- (ii) Using domains and ranges helps resolve polysemy, for example distinguishing a name of a person from a name of a place.
- $iii)$ When ontologies are multilingual, the same pipeline applies per language, and transitive closure over already aligned multilingual pivots can be used to seed additional candidates, as detailed in [Soulard(2022)].
- (iv) This method supports m-to-n property alignments and is particularly effective when ontologies contain implicit alternative properties or differ structurally. A representative example of a 1-to-n alignment would be matching Wikidata's $wk{:}P625$ (coordinate location) with $geo{:}lat$ and $geo{:}long$, or aligning $wk{:}P35$ (head of state) with $db{:}monarch$ and $db{:}leaderName$ in DBpedia.

These alignments can then be used to rewrite keys (see Definition 27) discovered in a graph $\mathcal{G}$ using the aligned properties of another graph $\mathcal{G}'$.

Definition 27 (Key Rewriting) *Let $\mathcal{G}$ and $\mathcal{G}'$ be two RDF graphs and let $\mathcal{M} = \{(p_1 \equiv p'_1), \ldots, (p_m \equiv p'_m)\}$ be a set of m property alignments between $\mathcal{G}$ and $\mathcal{G}'$.*

Let k be an S-key*$(C, (p_1, \ldots, p_n))$ for a class C in $\mathcal{G}$. A rewriting $\rho(k, \mathcal{M})$ of k in $\mathcal{G}'$, using the alignments in $\mathcal{M}$, is a set of* S-keys *of the form* S-key*$(C', (p'_1, \ldots, p'_n))$, such that there exists an alignment $\{(p_1 \equiv p'_1), \ldots, (p_n \equiv p'_n)\} \subseteq \mathcal{M}$ and we have $C \equiv C'$.*

If we take our previous example of a key k for a class C City, from Wikidata to DBpedia we would expect $k = \{rdfs{:}label \equiv rdfs{:}label, wdt{:}P17\ (country) \equiv dbr{:}Country\}$.

5.4.2 . Key Quality Measures

To assess key quality, we define the support of a key, the number of exceptions, and the discriminability rate.

Definition 28 (Support of a Key) *Let k be an* S-key*$(C, (p_1, \ldots, p_n))$ for a class C in $\mathcal{G}$. The support of k in $\mathcal{G}$ is the number of instances of C that have at least one value for each property in $prop(k)$. Formally:*

$$support(k) = \mid \{x \mid \forall p \in prop(k), \exists y, (x, p, y) \in G\} \mid$$

The relative support of a key with respect to the number of class instances is given by:

$$support^R(k) = \frac{support(k)}{\mid \{x \mid C(x) \in G\} \mid}$$

For example, an instance i such as wd:Q189792 (Zenopolis an ancient Roman and Byzantine city in Isauria) does not have a country relation in Wikidata resulting in our previous Key k not supporting this instance. Hence support measures how many instances a key could potentially match to form identity links.

Another measure is the key's discriminability, which evaluates how uniquely the key identifies each instance. As in [Atencia et al.(2020), Soru et al.(2015)], this is computed based on the number of partitions of instances that share the same values for the key properties and are reduced to a single element.

Definition 29 (Instance Partition Induced by a Key) *Given a key k in an RDF graph G defined as k =S-key$(C,(p_1,\ldots,p_n))$, the partition of class instances induced by k is the set: $\Delta(k,G)=\{\delta_1,\delta_2,\ldots,\delta_m\}$, where each δ_i is a subset of instances sharing the same values for $prop(k)$.*

Definition 30 (Discriminability Rate of a Key) *The discriminability rate of a key k =S-key$(C,(p_1,\ldots,p_n))$ is the ratio of singleton partitions to the total number of partitions in $\Delta(k,G)$:*

$$discr(k,G)=\frac{\mid\{\delta_i\mid\delta_i\in\Delta(k,G),\mid\delta_i\mid=1\}\mid}{\mid\Delta(k,G)\mid}$$

When knowledge graphs contain duplicate entities (i.e., the unique name assumption does not hold) or properties with erroneous values, strict key discovery becomes infeasible. For this reason, approaches such as [Symeonidou et al.(2014)] introduced keys that tolerate a given number of exceptions.

Definition 31 (Number of Exceptions for a Key) *The number of exceptions of a key k =S-key$(C,(p_1,\ldots,p_n))$ is the number of instances of C that do not satisfy the key. Formally:*

$$ex(k)=\mid\{x\mid\forall p\in prop(k),\exists y,y\neq x,V(x,p)\cap V(y,p)\neq\emptyset\}\mid$$

Since this definition becomes less relevant when property values are missing (e.g., fax-number P2900 in Wikidata is present in only 31 instances), we propose a refined definition in which the definition takes into account the support of the key:

Definition 32 (Relative Exception Rate of a Key) *The relative exception rate of a key k =S-key$(C,(p_1,\ldots,p_n))$ is computed with respect to the number of instances supporting the key:*

$$ex^R(k)=\frac{ex(k)}{support(k)}$$

5.4.3 . Key Transfer Approach

To assess the transferability of keys from a source graph G_1 to a target graph G_2, we followed these steps also shown in the figure 5.1:

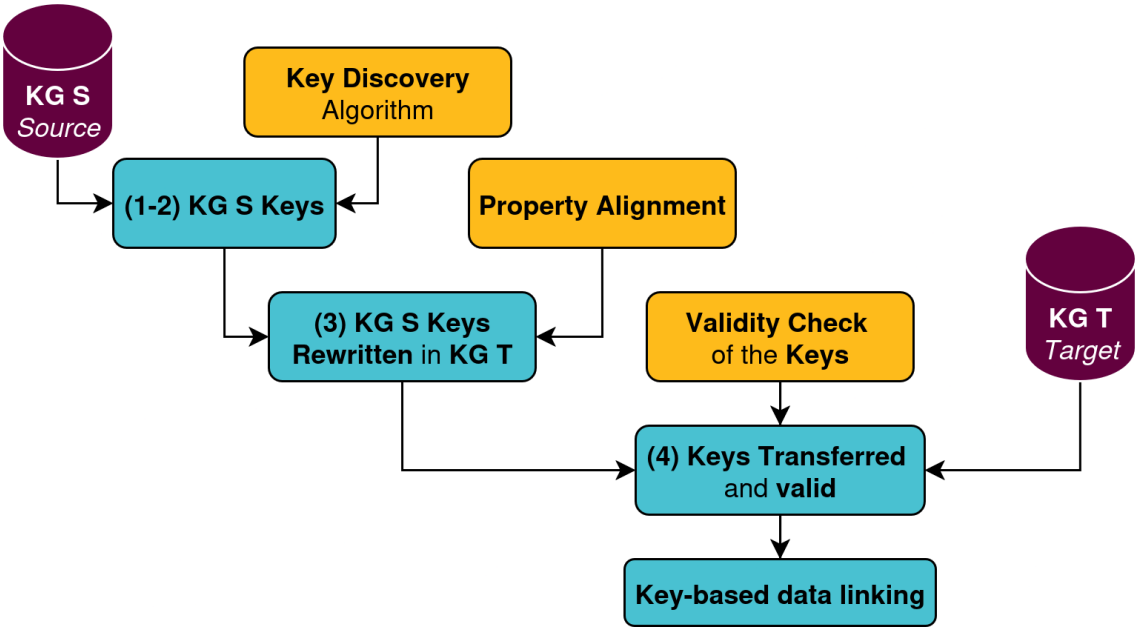


Figure 5.1: Key transfer workflow.

(1) Key discovery in the source graph. For each instance class in G_1 aligned with a class in G_2, we applied the SAKey tool [Symeonidou et al.(2014)] to discover a set of *S*-keys $\mathcal{K}_1$. SAKey accepts a parameter specifying the allowed number of exceptions but does not provide quantitative measures such as support or discriminability.

(2) Selection of high-quality keys within the source graph. Each discovered key is evaluated based on its number of exceptions and support. These are used to calculate its relative exception rate $ex^R(k)$. Only keys satisfying a predefined threshold ex^R_{max} are retained.

(3) Rewriting of selected keys using property alignments between both graphs. This step identifies the subset $K_1 \subseteq \mathcal{K}_1$ of keys discovered in G_1 for which property alignments with G_2 exist. These keys are then rewritten for use in the target graph.

(4) Evaluation of rewritten key quality in the target graph. Each rewritten key is evaluated in G_2 to measure its relative exception rate $ex^R(k)$ using the same threshold as in step (2). Keys that have degenerated into non-keys (i.e., $ex^R(k) > ex^R_{max}$) are discarded for the linking stage.

5.5 . Experimental Evaluation

In this section, we present the experimental evaluation of our framework on a series of datasets. All experiments were conducted on a machine equipped with an Intel® Xeon® E5-2630 v4 processor (10 cores) and 128 GB of RAM.

For the evaluation, we considered two types of dataset configurations: (i) two graphs using heterogeneous ontologies, where a property alignment tool must first be applied, potentially resulting in many-to-many alignments; and (ii) two graphs using a unified ontology, meaning that both graphs share the same set of properties.

5.5.1 . Graphs with Heterogeneous Vocabularies

We conducted the first experiment using two heterogeneous graphs extracted from DBpedia[3] and Wikidata[4], which rely on distinct vocabularies to describe their entities and concepts (that is, they use different sets of classes and properties). Ontology heterogeneity is handled using the property and class alignment procedure introduced in Section 5.4.1. Additionally, we assume that these graphs may contain erroneous or redundant entity descriptions. In order to discover keys under such imperfect conditions, we allow a maximum relative exception rate ex^R_{max}, chosen within the interval [0% ; 10%].

The objectives of this experimental evaluation are threefold: (i) to assess the performance gain of a key-based data linking approach when applying key transfer from a source graph to a target graph; (ii) to evaluate the quality of the transferred keys by measuring their impact on the discriminability of entities in both graphs; and (iii) to assess the importance of property alignment validation to prevent key degeneration. The rest of this section presents the workflow we followed during the experiment, describes how the datasets were prepared, and finally reports the results in terms of the number of keys obtained before and after transfer, the discriminability of the keys in both graphs, and the execution time, to determine whether key transfer leads to performance improvements.

Experimental Methodology

We consider a source graph G_S, a target graph G_T, a set $\mathcal{M}$ of equivalence alignments between properties from the two graphs, and a maximum relative exception threshold ex^R_{max}.

1. Data extraction and property alignment: Extract instances from G_S and G_T, and compute $\mathcal{M}$, the set of aligned properties.
2. Discovery of the set of *S*-keys SK in G_S, where each key satisfies $ex^R(k) < ex^R_{max}$.
3. Use of $\mathcal{M}$ to rewrite the discovered keys, selecting a subset $RK \subset SK$ in which each key has all its properties aligned to at least one property in G_T.

[3] https://www.dbpedia.org/
[4] https://www.wikidata.org/

4. Computation of support, relative exception rate, and discriminability for each key in RK within G_S.

5. Computation of support, relative exception rate, and discriminability for each key in RK within G_T.

Datasets

We have used two datasets extracted from DBpedia and Wikidata.

	# Entities	# Properties	Processed # Properties	# Triples
$DB{:}Person$	1 863 013	18 960	239	12 474 844
$WK{:}Q5$	3 020 916	3 506	135	7 503 002

Table 5.1: Class $Person$/$Human$ in DBpedia/Wikidata

In the case of DBpedia, we extracted the portion of the RDF graph corresponding to instances of the $Person$ class (i.e., resources with $rdf{:}type$ equal to $Person$) from a full DBpedia dump dated December 2022. As shown in Table 5.1, this initial dataset contains 1,863,013 entities and 18,960 properties. Upon inspection, we observed that a significant number of these properties are associated with incorrectly encoded or invalid IRIs, meaning IRIs that do not correspond to any property defined in the relevant ontologies. For example, the property $dbo{:}award21n, 2$ appears instead of the expected $dbo{:}award$. Such erroneous properties were removed from the final dataset.

For Wikidata, we extracted all instances of the Human class from the HDT-encoded Wikidata dump dated March 3, 2021[5]. These instances are identified by the triple pattern $\langle ?IRI,\ \ wdt{:}P31, wd{:}Q5\rangle$. The resulting dataset comprises 3,020,916 entities and 3,506 properties.

Properties alignment

For both datasets, we also considered the ontologies describing both DBpedia and Wikidata graphs, which contain 4,604 and 10,472 properties respectively. We applied a property alignment algorithm described in [Soulard(2022)], which uses a similarity measure based on embeddings of the textual descriptions associated with the properties and classes in a low-dimensional vector space.

After data extraction, a preprocessing step was applied to reduce the number of unusable properties. This step was motivated by the runtime complexity of the key discovery algorithm, which depends on the number of properties in a graph. To address this, we retained only those properties used in both graphs for which at least one alignment exists. This strategy was adopted because non transferable properties or those not shared between the graphs cannot be used for key rewriting. Additionally, we kept only properties

[5]Available on the RDF HDT service: `https://www.rdfhdt.org/datasets/`

of type $owl{:}DatatypeProperty$, that is, those whose values are literals. In contrast, a key composed of $owl{:}ObjectProperty$ can be leveraged only by methods that propagate similarity scores across instance pairs, such as [Al-Bakri et al.(2015), Saïs et al.(2009)]. After preprocessing, we obtained 239 and 135 properties and 12,474,844 and 7,503,002 triples for DBpedia and Wikidata, respectively, as shown in Table 5.1.

Results

	ex^R_{max}	0%	0.5 %	2%	5%	10%
DB	SK	835	642	642	643	648
	RK raw	1,572	1,199	1,199	1,200	1,205
	RK (sup$\neq$ 0)	69	64	64	65	68
	RK verified	49	52	54	56	60
WK	SK	357	237	238	240	241
	RK raw	475	327	328	330	331
	RK (sup$\neq$ 0)	82	82	82	83	83
	RK verified	41	48	49	55	57

Table 5.2: Key discovery results for the Person/Human datasets using different relative exception rates (SK $\rightarrow$ Source Key, RK $\rightarrow$ Rewritten Key)

Number of Discovered Keys Table 5.2 shows the number of source keys (SK) for each knowledge graph, along with the number of rewritten keys (RK) that were validated in the target graph. This validation is performed in two steps. In the first step, we filter out rewritten keys with no instantiation (see Definition 26), that is, with a support of zero. In the second step, we check whether the rewritten keys comply with the maximum allowed relative exception rate.

The main reason for this filtering is the absence of instantiation of the rewritten keys in the target graph. This lack of instantiation can be explained by different perspectives or contexts used to describe entities—for example, one graph might describe persons from a social perspective while another uses a professional one. Additionally, n-to-m alignments between properties may lead to unnecessary rewritings, especially when the aligned properties are not actually used to describe `Human` entities.

We observe a sharp drop in the number of keys between the thresholds 0% and 0.5%. At 0% no exception is allowed, so any candidate with even a tiny error is rejected. For example, if the candidate key {P1} has an exception rate of 0.1%, it is discarded at the 0% threshold. The miner must then add properties until the exception rate reaches zero, producing alternatives such as P1, P2 or P1, P3. When the threshold is relaxed to 0.5%, many near perfect single property keys become valid; these simpler keys subsume several multi property candidates, which explains the drop in the number of keys found.

Finally, when comparing the number of source and rewritten keys, we see that these filtering steps are essential. From DBpedia, we lose over 90% of keys, and from Wikidata, over 80%.

	ex^R_{max}	0%	0.5 %	2%	5%	10%
From DB	DB	0.3%	80.7%	80.7%	80.7%	80.7%
	WK	0.9%	28.9%	28.9%	28.9%	28.9%
From WK	WK	0.3%	0.9%	0.9%	1.0%	1.0%
	DB	0.1%	80.7%	80.7%	80.7%	80.7%

Table 5.3: Discriminability of transferred keys according to the maximum relative exception rate.

Discriminability of Keys Table 5.3 reports the discriminability of key sets in both directions, that is, considering DBpedia as the source and Wikidata as the target, and vice versa. Discriminability, as defined in def 30, refers to the percentage of entities in a graph that can be uniquely identified by at least one mined key, given a relative exception threshold $\tau = ex^R_{\text{max}}$. The table presents results for τ values ranging from 0% to 10%. as defined in def 30,

As a concrete illustration, consider a graph with 100 entities. If the mined keys uniquely identify 83 entities with $\tau = 0\%$, the discriminability is 83%. When the threshold is relaxed to $\tau = 5\%$, a key that previously produced a small number of exceptions may become acceptable, which can increase the discriminability, for example from 72% to 88%. This example highlights how tolerance to exceptions directly influences the proportion of entities that can be uniquely identified by at least one key.

We observe that from 0.5% onward, discriminability reaches a plateau (80.7% and 28.9% from DBpedia; 0.9% and 80.7% from Wikidata), indicating that increasing the exception rate does not improve entity discrimination. This plateau is particularly important because higher thresholds introduce more noise, that is, more exceptions, which makes entity matching more difficult. These results also highlight a limitation of the key discovery method: on Wikidata, the maximum discriminability reached by discovered keys applied to the same graph is only 1.0%, showing that very few entities can be uniquely identified under strict exception constraints.

Execution Time Table 5.4 reports execution times for each strategy and starting point. The most time-consuming step is the source key discovery for DBpedia (279 minutes), which is the bottleneck of the classical key merging strategy. Therefore, by starting with the graph having the lowest execution time in the **SK discovery** phase, Wikidata due to its lower number of properties, we can drastically reduce the overall runtime to just 14.5 minutes compared to 321.7 minutes at a 0% exception threshold. However, if we begin with DBpedia, we lose the benefit of this strategy and remain within the same runtime range as the classical merging approach.

Discussion This initial experimental evaluation allowed us to analyze the behavior of rewritten keys on source and target graphs. Although verifying the rewritten keys in the target graph is necessary, we have shown that these rewritten keys can still discriminate a substantial percentage of entities in the target graph (Table 5.3), even when more than

	ex^R_{max}	0%	0.5 %	2%	5%	10%
Classic DB ∩ WK	SK Discovery DB	279.3	276.0	270.8	278.0	282
	SK Discovery WK	5.1	5.0	5.1	5.0	5.1
	SK Evaluation DB	33.7	25.3	25.4	25.4	25.5
	SK Evaluation WK	3.4	1.5	1.5	1.5	1.6
	Total	**321.7**	**308.0**	**302.9**	**310.1**	**313.9**
Transfer From DB	SK Discovery	279.3	276.0	270.8	278.0	282
	SK Evaluation	33.7	25.3	25.4	25.4	25.5
	RK Evaluation	11.7	11.6	11.7	11.9	11.7
	Total	**324.8**	**313.0**	**308.0**	**315.4**	**319.2**
Transfer From WK	SK Discovery	5.1	5.0	5.1	5.0	5.1
	SK Evaluation	3.4	1.5	1.5	1.5	1.6
	RK Evaluation	5.9	5.8	5.9	6.0	5.9
	Total	**14.5**	**12.4**	**12.6**	**12.6**	**12.6**

Table 5.4: Execution time in minutes for each scenario and processing step. (SK $\rightarrow$ Source Key, RK $\rightarrow$ Rewritten Key)

80% of the original keys are discarded during validation (Table 5.2). However, the use of relative exception thresholds in SAKey [Symeonidou et al.(2014)] affects the monotonicity of keys and, consequently, the completeness of the transferred key set. As illustrated in Table 5.5, key $K_1 = \{P_1, P_2\}$ complies with a 50% exception threshold (with 40% exceptions), while $K_2 = \{P_1, P_2, P_3\}$ exceeds this threshold with the same number of exceptions but lower support, resulting in 100% exceptions. In contrast, $K_3 = \{P_1, P_2, P_3, P_4\}$ meets the allowed threshold with no exceptions.

	P_1	P_2	P_3	P_4
e_1	A1	A2	-	-
e_2	B1	B2	A3	B4
e_3	B1	B2	A3	C4
e_4	D1	D2	-	-
e_5	E1	E2	-	-

Table 5.5: Example of non-monotonic keys.

Completeness is particularly important, as more specific keys (with lower support) may distinguish entities that more general keys (with higher support) cannot. The example in Table 5.5 illustrates this: key K_1 cannot distinguish e_2 from e_3, whereas K_3 can. Therefore, one might aim to increase discriminability by adding more specific rewritten keys that differentiate special cases. However, doing so is non-trivial, as key monotonicity is lost with relative exception thresholds. Of course, adding keys increases the runtime for both key discovery and entity matching, as more values must be compared. This limitation applies equally to both classical and key-transfer approaches and thus does not impact the current evaluation, though it may improve results in future key-based methods.

5.5.2 . Graphs with Harmonized Vocabulary

In this second experimental evaluation, we use several datasets provided by Symeonidou et al. [Symeonidou et al.(2017)], which are extracted from DBpedia and YAGO. Their property names have been harmonized (i.e., standardized to use the same IRIs) using a semi-automated ontology alignment process. This evaluation focuses primarily on a qualitative comparison between our key-transfer-based data linking approach and the classical approach, which relies on the intersection of keys discovered in both graphs. We also present quantitative results regarding key characteristics and execution times.

To do so, we use the datasets provided by [Symeonidou et al.(2017)] and compare the quality of the final entity alignment achieved by both approaches. As in the previous experiment, we varied the maximum relative exception rate across four values (i.e., $[0\%, 0.5\%, 2\%, 5\%]$). As previously observed, performance typically plateaus once a small exception threshold is permitted. The experimental procedure closely mirrors that of the first experiment (Section 5.5.1), except for steps 1 and 2, which are no longer required.

		# Entities	# Properties	# Triples
Actor	DB	5 807	16	47 303
	YAGO	108 424	16	514 790
Album	DB	85 069	5	297 072
	YAGO	137 167	5	381 119
Book	DB	29 849	7	123 686
	YAGO	41 855	7	92 516
City	DB	19 229	17	652 042
	YAGO	83 560	17	1 100 896
Film	DB	82 124	9	684 800
	YAGO	123 912	9	533 542
Moutain	DB	16 401	5	32 981
	YAGO	67 721	6	116 781
Museum	DB	1 826	7	7 970
	YAGO	21 152	7	81 671
Univ.	DB	10 352	9	120 919
	YAGO	23 351	9	131 812

Table 5.6: Characteristics of the 8 classes in DBpedia and YAGO.

Once validated keys are found in both graphs, they are used to link entities sharing at least one value for every property in the key. However, multiple identity links may be suggested for the same entity using different keys. Since the Unique Name Assumption (UNA) is nearly respected in both DBpedia and YAGO datasets, we defined a strategy to prevent n-to-m links that could reduce the precision of the approach. The strategy ranks keys by support (higher support indicates a better key) and exception rate (lower exception rate is better), and applies linking in that order. For instance, with keys ranked as $[k_1, k_2]$, if k_1 links e with e', and k_2 suggests a link between e and e'', then k_2's link is discarded since e was already matched by a better key.

Datasets

We used the `dbpedia-yago` dataset made publicly available in [Symeonidou et al.(2017)][6]. This dataset uses a harmonized vocabulary and is accompanied by a gold standard containing all identity links to be discovered. We selected 8 classes that differ in characteristics, some involving large volumes of data—for example, the $City$ class includes more than one million triples. The characteristics of each class, including the number of entities, properties, and triples, are shown in Table 5.6.

Interestingly, the number of triples is not directly correlated with the number of entities or properties (e.g., compare $Film$ and $City$).

Execution Time

	Step	S-K Discovery	S-K Verification	T-K Verification	Total
Actor	DB	3s	58s	2m 08s	**3m 10s**
	YAGO	4m 12s	4m 02s	1m 54s	9m 50s
	INTER	4m 15s	4m 41s	-	8m 57s
Album	DB	1m 56s	13s	14s	2m 23s
	YAGO	12s	25s	29s	**1m 06s**
	INTER	2m 8s	38s	-	2m 47s
Book	DB	4s	22s	17s	**43s**
	YAGO	6s	36s	23s	1m 6s
	INTER	10s	58s	-	1m 8s
City	DB	23s	15m 15s	6m 20s	22m 8s
	YAGO	2m 4s	3m 34s	3m 16s	**8m 55s**
	INTER	2m 27s	18m 49s	-	21m 17s
Film	DB	6m 25s	1m 21s	1m 04s	**8m 51s**
	YAGO	3m 05s	20m 03s	49s	23m 57s
	INTER	9m 30s	21m 24s	-	30m 54s
Moun.	DB	1s	15s	16s	**32s**
	YAGO	4s	21s	22s	47s
	INTER	5s	36s	-	41s
Mus.	DB	1s	23s	35s	59s
	YAGO	2s	26s	20s	**48s**
	INTER	3s	50s	-	52s
Univ.	DB	9s	27s	33s	**1m 09s**
	YAGO	12s	37s	29s	1m 18s
	INTER	21s	1m 04s	-	1m 25s

Table 5.7: Execution time for source key discovery, source key verification, and transferred key verification for each scenario (i.e., from DBpedia, from YAGO, and from intersection).

[6] Available at https://github.com/lgalarra/vickey

	Step	#Keys	Size of Keys	#T-K	Size of T-K
Actor	DB	93	2.24	50.7	2.3
	YAGO	183	3.08	156.7	3.0
	INTER	25	2.08	25.0	2.1
Album	DB	1	2.0	1.0	2.0
	YAGO	3	2.0	2.0	2.0
	INTER	1	2.0	1.0	2.0
Book	DB	5	2.80	5.0	2.8
	YAGO	8	2.375	3.0	2.2
	INTER	1	2.0	1.0	2.0
City	DB	178	2.88	159.0	2.9
	YAGO	114	2.5	81.3	2.6
	INTER	40	2.5	40.0	2.5
Film	DB	14	3.86	12.3	3.8
	YAGO	9	2.89	5.7	3.7
	INTER	0	-	0	-
Moun.	DB	14	2.0	3.0	2.0
	YAGO	10	2.0	4.3	2.0
	INTER	3	2.0	3	2.0
Mus.	DB	14	2.14	11.3	2.1
	YAGO	10	2.10	9.3	2.1
	INTER	8	2.0	8.0	2.0
Univ.	DB	17	2.18	14.3	2.1
	YAGO	19	2.26	16.3	2.3
	INTER	12	2.08	12.0	2.1

Table 5.8: Number and average size of source keys and transferred keys per dataset and scenario (i.e., from DBpedia, from YAGO, and from intersection).

Table 5.7 compares execution times for the key transfer approach in two scenarios (DBpedia as source and YAGO as target, and vice versa) with the classical key intersection approach (line "INTER" in Table 5.7). The transfer-based approach is faster across all 8 classes. For example, for the $Film$ class, execution time is reduced by 73% when DBpedia is used as the source.

As in the previous experiment, we examine how the choice of source graph affects performance. However, since the graphs have the same number of properties, the previous bottleneck (key discovery on graphs with many properties) no longer applies. Instead, new factors must be considered. Comparing Tables 5.6 and 5.7, we find no consistent rule to determine the best starting point. For instance, for $Actor$, using the smaller graph (fewer triples and entities) is faster, but for $Album$ and $City$, the opposite holds.

Table 5.8 presents a comparison of the number and length of discovered keys and transferred keys (T-K). Fewer and longer keys tend to result in faster execution. However, before executing the framework on a specific graph, we have no assumption to anticipate

the number or specificity of the keys the algorithm will produce. Therefore, beyond the number of properties and instances, we currently lack a reliable criterion to determine the optimal starting point for minimizing total execution time.

Data Linking Results

Table 5.9 reports the results of data linking using keys to match instances between DBpedia and YAGO for each dataset. The evaluation metrics are recall, precision, and F1 score, and results are presented for increasing values of the maximum relative exception threshold ex^R_{max}. Each block corresponds to a dataset, while the rows labeled *D*, *Y*, and *I* respectively denote configurations where DBpedia is used as the source graph, YAGO as the source graph, or the intersection-based approach.

As an illustration of how to read the table, consider the `City` dataset. When YAGO is used as the source and the exception threshold is 0.5%, recall is 65.7%, precision is 96.5%, and the F1 score reaches 78.2%. This means that by allowing a small exception margin of 0.5%, the system correctly identifies most city correspondences between the two graphs with high precision. Similarly, when DBpedia is used as the source, the best F1 score obtained is 75.3%, showing that the choice of source graph has a notable influence on the linking performance.

Comparing the key transfer approach to the classical intersection-based method reveals that the intersection approach is consistently worse or equal to the worst-performing transfer configuration. In some cases, the performance gains are dramatic. For example, for the `Album` dataset, F1 improves from 0.03% to 57.5%. The intersection approach can only retrieve a subset of the keys found across both graphs. Moreover, when a small exception margin is allowed (e.g., 0.5%, 2%, or 5%), the best results are consistently obtained when YAGO is used as the source graph and DBpedia as the target.

5.6 . Conclusion

This chapter introduced a key transfer strategy designed to avoid repeated key discovery on every knowledge graph involved in data linking. By mining keys once on a suitable source graph, rewriting them through property alignments, and validating them on the target, we achieved substantial runtime reductions, with observed speedups of up to a factor of 22 when the source graph was chosen judiciously. In the case of heterogeneous vocabularies, experiments on DBpedia and Wikidata indicate that the graph with fewer relations often provides a better starting point, acting as a proxy for a smaller key search space. For harmonized vocabularies, however, the optimal choice of source remains less clear and appears to depend on class distribution, data noise, and alignment coverage.

A central methodological lesson is the necessity of validating rewritten keys in the target, since unrevised or degenerate candidates can otherwise introduce errors. We also showed that allowing a small relative exception tolerance of around 0.5% consistently improves interlinking quality compared to strict discovery, as it accommodates realistic

ex^R_{max}		0			0.5%			2%			5%		
Metric		R	P	F1	R	P	F1	R	P	F1	R	P	F1
Actor	D	23.5	99.8	38.0	25.4	99.6	40.5	27.5	98.0	42.9	27.5	97.8	42.8
	Y	15.8	99.6	27.3	52.6	99.8	68.9	52.6	99.8	**68.9**	52.6	99.8	**68.9**
	I	14.4	99.6	25.2	14.4	99.6	25.2	14.4	99.6	25.2	14.4	99.6	25.2
Album	D	0.01	75.0	0.03	0.01	75.0	0.03	0.01	75.0	0.03	0.01	75.0	0.03
	Y	0.01	75.0	0.03	24.0	96.7	38.5	41.2	95.1	**57.5**	41.2	95.1	**57.5**
	I	0.01	75.0	0.03	0.01	75.0	0.03	0.01	75.0	0.03	0.01	75.0	0.03
Book	D	2.2	84.6	4.3	3.5	94.9	6.7	3.5	94.9	6.7	3.5	94.9	6.7
	Y	2.9	95.4	5.7	31.4	96.9	47.4	44.2	96.3	**60.6**	44.2	96.3	**60.6**
	I	75.0	95.4	5.7	75.0	95.4	5.7	75.0	95.4	5.7	75.0	95.4	5.7
City	D	60.5	98.5	74.7	60.9	98.5	75.3	60.9	98.5	75.3	61.0	98.4	75.3
	Y	31.3	99.6	47.6	65.7	96.5	78.2	65.9	96.3	**78.3**	66.2	95.8	**78.3**
	I	0	100	0	0	100	0	0	100	0	0	100	0
Film	D	3.2	98.5	6.2	4.2	98.1	8.1	4.2	98.1	8.1	4.2	98.1	8.1
	Y	0.72	97.9	1.42	16.9	98.4	28.9	30.5	97.3	**46.5**	30.5	97.3	**46.5**
	I	0	100	0	0	100	0	0	100	0	0	100	0
Moun.	D	0.22	100	0.44	0.22	100	0.44	0.22	100	0.44	0.22	100	0.44
	Y	0.22	100	0.44	1.57	99.5	3.1	1.57	99.5	**3.1**	1.57	99.5	**3.1**
	I	0.22	100	0.44	0.22	100	0.44	0.22	100	0.44	0.22	100	0.44
Mus.	D	3.5	70.2	6.7	3.9	67.9	7.3	3.9	67.9	7.3	3.9	67.9	7.3
	Y	1.3	71.0	2.6	1.3	71.0	2.6	4.1	66.5	**7.7**	4.1	66.5	**7.7**
	I	1.3	71.0	2.6	1.3	71.0	2.6	1.3	71.0	2.6	1.3	71.0	2.6
Univ.	D	8.9	99.1	16.3	9.0	99.0	16.4	9.0	98.9	16.4	9.0	98.9	16.4
	Y	9.0	99.1	16.3	10.1	99.1	18.4	12.4	98.9	**22.1**	12.4	98.9	**22.1**
	I	8.9	99.1	16.3	8.9	99.1	16.3	8.9	99.1	16.3	8.9	99.1	16.3

Table 5.9: Data linking performance as a function of the maximum relative exception threshold {0%, 0.5%, 2%, 5%}. *D* and *Y* refer to DBpedia and YAGO as source graphs, respectively. *I* refers to the classical intersection-based approach.

data imperfections. This demonstrates the practical benefit of combining efficiency with robustness, while also preserving the interpretability of symbolic rules.

Beyond immediate performance gains, the study supports a broader perspective in which discovered keys are treated as reusable resources rather than one-off outputs. This vision contributes not only to explainable symbolic alignment, but also to hybrid frameworks where richer entity descriptions benefit deep learning approaches. In summary, key transfer transforms key discovery from a repetitive procedure into a reusable knowledge asset, paving the way toward scalable and explainable KG interoperability. This motivation naturally leads to the next chapter, which introduces *KG Nexus* (chapter 6), a resource for storing schema alignments, provenance, and pre-discovered keys in support of reproducible and sustainable alignment knowledge.

5.7 . Future Work

We identify several directions to strengthen the scientific and practical impact of key transfer.

5.7.1 . Comparative evaluation of key discovery methods

A systematic comparison of key discovery methods in the transfer setting is required to better understand their respective strengths and weaknesses. Such an evaluation should measure not only performance in terms of runtime and interlinking quality, but also robustness to noise and sensitivity to class imbalance.

In addition, it is important to analyze whether theoretical properties of keys, such as minimality and practical monotonicity, are preserved after rewriting into the target graph. This comparative study would provide valuable insights into which discovery strategies are most reliable for key transfer and under what conditions they remain effective.

5.7.2 . Key discovery with native relative exception control

In this chapter, we defined the relative exception rate (Definition 32) but applied it only as a post hoc filter: keys are first mined under a global absolute bound x, then discarded if their observed relative rate exceeds ϵ. This approach wastes partially informative property sets that could yield valid keys after targeted augmentation.

Future work could explore the design of a miner that integrates relative thresholds directly during the search. An optional two-phase schedule can be used: phase one applies a small absolute bound x for fast screening, while phase two maintains for each candidate K an estimate of $\mathsf{supp}(K)$ and an upper bound on $\mathsf{exR}(K)$. Candidates exceeding ϵ are pruned early, while near-violations trigger guided augmentation $K \to K \cup \{p\}$. Such a level-wise implementation with relative-aware pruning and augmentation heuristics would return the complete set $\mathcal{K}_{\leq \epsilon}$ and improve discriminability on alignment tasks, while avoiding wasted computation from post hoc rejection.

5.7.3 . Confidence-aware alignment handling

Our framework currently relies on property alignments produced by language models, which are associated with confidence values. At present, these confidences are only used through a hard threshold: alignments above the threshold are accepted as ground truth, while those below are discarded. A more principled approach would retain the confidence values and propagate them throughout the key transfer process.

Such confidence-aware handling would enable conflict resolution between competing keys, giving priority to those supported by higher-confidence alignments. In addition, alignment confidence could be combined with the relative exception rate of each key to provide a richer estimate of its reliability. This joint perspective would yield a more nuanced notion of key precision and help reduce the impact of low-confidence alignments on the overall linking task.

5.7.4 . Neuro-symbolic framework

Symbolic approaches offer high interpretability by relying on explicit rules, in this case keys, to infer alignments. However, they often suffer from limited coverage, as only a restricted number of alignments can be discovered. In contrast, deep learning methods [Sun et al.(2020)] can produce a larger number of alignments but at the cost of interpretability and the need for costly seed correspondences usually hand crafted.

A promising direction is to merge both paradigms into a neuro-symbolic framework for entity alignment. Keys could provide interpretable seeds and constraints, guiding the neural models toward high-quality correspondences. In turn, the learned embeddings could extend coverage beyond what purely symbolic methods achieve. Such a hybrid approach would enable a more automated and scalable alignment process, while preserving interpretability through the explanatory power of keys.

5.7.5 . Temporally aware alignment

As discussed in Chapter 4, tasks such as Query Answering or Entity Alignment can benefit from the additional dimension introduced by temporal context. Temporal constraints make it possible to prune contradictory candidate alignments and to mine time-scoped keys that respect period-specific attribute combinations.

Incorporating temporal awareness into the alignment process would allow more precise and context-sensitive correspondences, reducing errors that arise from attributes which evolve over time. Such an extension would directly support the vision of *KG Nexus* presented in Chapter 6, where reusable alignment knowledge is enriched with explicit uncertainty, provenance, and temporal scope.

Chapter 6

KG Nexus: Bridging the Web of Data

*The work presented in this chapter has been in collaboration with **Fatiha Saïs, Joe Raad, Nicolas Denier and Anne-Catherine Letournel**.*

The results of Chapter 5 demonstrated that keys can, under well-defined conditions, be transferred across heterogeneous Knowledge Graphs, thereby reducing the computational cost of systematic key discovery while preserving linking quality. However, these experiments also highlighted the absence of a dedicated infrastructure for storing, sharing, and reusing such transferable alignment knowledge at scale. Without a centralized resource, each discovery remains isolated, limiting both reproducibility and the broader impact on the Web of Data. To overcome this limitation, the following chapter introduces *KG Nexus*, a curated and extensible platform designed to operationalize the lessons learned from key transferability by consolidating vocabularies, schema alignments, and reusable keys into a single, openly accessible hub.

6.1 . Introduction

The Web of Data envisions a globally interconnected ecosystem of structured knowledge, where disparate datasets can be seamlessly integrated and reasoned over. Central to this vision are **KGs**, which represent entities and their relationships in a machine-interpretable form using standards such as RDF and OWL. However, while the number and scale of KGs have grown significantly, their utility remains limited by fragmentation and heterogeneity across datasets.

One of the most critical challenges in this space is the problem of **cross-KG alignment**, the task of identifying semantically equivalent entities and schema elements across independently developed KGs. Differences in vocabulary, modeling conventions, and coverage lead to significant barriers in integrating information across domains. To overcome

these, two main alignment strategies have emerged: *ontology alignment*, which maps schema elements (classes, properties) across KGs, and *entity linking*, which identifies when two entities from different graphs refer to the same real-world object.

Among entity linking approaches, **key-based methods** have gained attention due to their precision and explainability [Jiménez-Ruiz(2024), Nassiri et al.(2020)]. As shown in Chapter 5, keys constitute discriminative sets of properties that uniquely identify instances of a given class and can thus be leveraged for transparent and verifiable alignment. Compared to similarity or embedding-based methods, which may act as black boxes, key-based linking provides explicit justifications that are easier to debug, validate, and trust. For example, the property $isbn$ uniquely identifies books across datasets, while a composite key such as $\{title,\ author,\ year\}$ can distinguish different editions of a publication. By relying on such keys, one can link entities across graphs with both high confidence and structured interpretability.

Key-based alignment, however, faces two persistent challenges that were thoroughly analyzed in Chapter 5.

Scalability: remains a bottleneck, discovering keys in large-scale KGs such as DBpedia or Wikidata requires exploring an exponential space of candidate combinations, and our experiments showed that discovery can take hours or days even with pruning strategies such as VICKEY [Symeonidou et al.(2017)] or SAKey [Symeonidou et al.(2014)].

Schema Heterogeneity: prevents straightforward reuse of keys across datasets. While *link keys* [Atencia et al.(2021)] extend the notion of keys to heterogeneous vocabularies, our evaluation demonstrated that they amplify computational costs by requiring systematic cross-graph exploration.

Most importantly, Chapter 5 established that under well-defined conditions, keys discovered in one graph can be transferred to another by means of property alignments and validation, thereby avoiding redundant discovery. This finding underlines the potential of treating keys not as one-off artifacts but as reusable resources for scalable linking.

To build upon these improvements, we propose **KG Nexus** a modular and extensible resource designed to enable scalable, explainable entity alignment across heterogeneous Knowledge Graphs. This resource will consolidate alignment-relevant knowledge and serves as a bridge between datasets in the Web of Data. It includes:

- A curated catalog of vocabularies and ontologies, including those from LOV [Dumontier et al.(2017)];
- A collection of prominent KGs linked to the vocabularies they use, enriched with metadata such as provenance, licensing, and versioning;
- Schema-level alignments, including mappings between equivalent or related classes and properties across different vocabularies;

- Automatically discovered *intra-KG keys* for commonly used classes;
- *Transferable keys*, indicating which keys can be reused for cross-KG entity alignment, even in the presence of schema mismatches.

By centralizing these resources, KG Nexus facilitates scalable and transparent alignment workflows. It significantly reduces the computational overhead of key discovery and schema matching by providing precomputed, reusable alignment knowledge. Furthermore, it supports explainability through key-based matching and schema annotations.

The KG Nexus dataset is openly available under a Creative Commons license[1]. It can be accessed via a public Web interface[2] and a SPARQL endpoint[3], enabling seamless integration into KG alignment pipelines and applications. Designed for extensibility, it also includes tools to support the integration of new datasets and vocabularies, paving the way for a more interconnected and trustworthy Web of Data.

The remainder of this chapter is organized as follows. Section 6.2 reviews related initiatives in vocabulary repositories, link repositories, entity naming services, and domain-specific KGs, highlighting their strengths and limitations with respect to alignment and reuse. Section 6.3 then details the creation of KG Nexus, including vocabulary harmonisation, ontology alignment, dataset incorporation, and the integration of reusable keys. Section 6.4 presents descriptive statistics and analyses of the resulting resource, while Section 6.5 illustrates its practical value through a series of representative use cases. Finally, Section 6.6 concludes with a summary of contributions while Section 6.7 outlines directions for future extensions of KG Nexus.

6.2 . Related Work

Efforts to facilitate reuse and integration in the Web of Data have produced several shared resources. These can be broadly categorized into four types: vocabulary and ontology repositories, link repositories for entity alignment, centralized entity naming services, and integrated domain-specific knowledge graphs. While each of these directions has provided significant value, they also exhibit limitations that motivate the design of KG Nexus.

6.2.1 . Vocabulary and Ontology Repositories

Platforms such as Linked Open Vocabularies (LOV) [Dumontier et al.(2017)], BioPortal [Noy et al.(2009)], and AgroPortal [Jonquet et al.(2018)] provide centralized catalogs of ontologies and vocabularies across multiple domains. For instance, LOV curates over 800 ontologies to promote reuse and interoperability, BioPortal aggregates a wide range of biomedical ontologies, and AgroPortal focuses on the life sciences. These repositories

[1] https://doi.org/10.5281/zenodo.15396328
[2] http://kgnexus.lisn.upsaclay.fr/
[3] https://kgnexus.lisn.upsaclay.fr/api/nexus/

have been instrumental in simplifying ontology discovery, reuse, and dissemination. However, they mainly serve as registries of vocabularies and do not directly support schema-level alignments or instance-level matching across knowledge graphs. As a result, while they lower the entry barrier to ontology reuse, they do not provide the mechanisms necessary for cross-KG integration.

6.2.2 . Link Repositories for Entity Alignment

Dedicated services such as sameAs.cc [Beek et al.(2018)] and MetaLink [Beek et al.(2020)] collect and expose $owl{:}sameAs$ links between entities across KGs. MetaLink, for example, offers more than 550 million identity links and publishes their transitive closure as equivalence classes. These platforms rely on scalable compression techniques such as HDT [Fernández et al.(2011)] to handle billions of triples efficiently. Although such repositories are invaluable for retrieving existing identity links, they remain limited to surfacing what has already been established. They do not include schema-level mappings nor provide information that would allow the generation of new identity links in unseen datasets.

6.2.3 . Centralised Entity Naming Services

Entity Name Services (ENS), such as OKKAM [Bouquet et al.(2007)], aim to provide authoritative naming schemes for entities on the Web. By ensuring that every real-world entity is associated with a single identifier, these systems address the challenge of entity alignment at web scale. While promising in theory, centralized naming authorities face considerable adoption barriers due to the scale, heterogeneity, and autonomy of the Web of Data. Consequently, their impact has remained confined to specific domains or applications, rather than becoming a general solution for cross-KG alignment.

6.2.4 . Integrated Domain-Specific Knowledge Graphs

Domain-focused projects such as AgroLD [Larmande and Todorov(2021)] in agronomy construct large RDF knowledge graphs by integrating heterogeneous datasets under a shared ontology. AgroLD, for instance, produces hundreds of millions of triples covering a rich set of classes and properties. These initiatives demonstrate the benefits of harmonizing vocabularies within a single field, but they remain tightly coupled to domain-specific requirements and lack generalizability. They cannot directly support cross-domain alignment or on-demand reuse of schema mappings in unrelated application areas.

6.2.5 . Infrastructure-Level Platforms for RDF Data

Another line of work focuses on large scale infrastructures that aim at improving the accessibility, quality, and reuse of RDF datasets on the Web. A representative example is LOD Laundromat [Beek et al.(2014)], which provides a curated collection of cleaned, normalized, and dereferenceable RDF datasets obtained from the Web of Data. By enforcing best practices such as syntactic correctness, consistent serialization formats, and stable access mechanisms, such platforms significantly lower the technical barrier for reusing existing RDF data at scale. However, these infrastructures primarily operate at the dataset

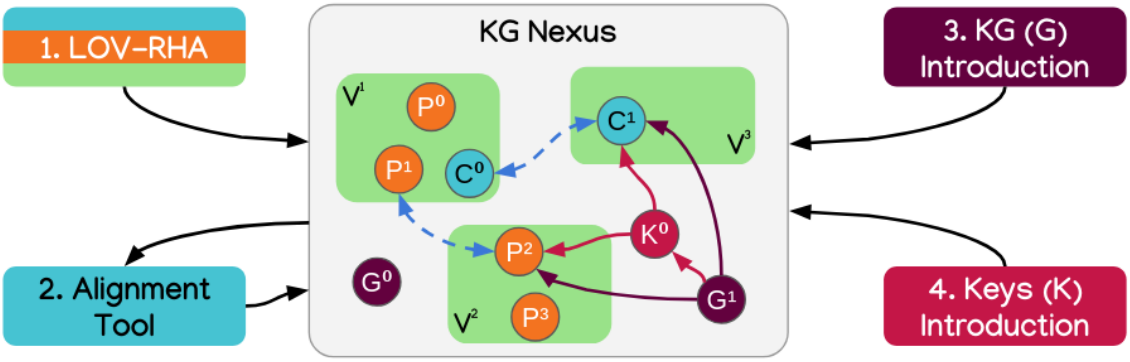


Figure 6.1: Overall workflow of KG Nexus creation.

level and do not address schema heterogeneity, vocabulary alignment, or identity resolution across knowledge graphs. As a result, while they provide an essential foundation for reliable data access, they do not offer mechanisms for integrating heterogeneous schemas or enabling cross-graph reasoning based on shared conceptual structures.

Taken together, these initiatives make vocabularies, ontologies, and precomputed links available to users, but they stop short of enabling dataset producers to extend these resources with their own data. In contrast, KG Nexus allows any data provider to incorporate a new KG by simply linking it to an existing vocabulary in the Nexus. Through the transitivity of equivalence links (e.g., $owl{:}equivalentClass$, $owl{:}equivalentProperty$), this single connection automatically integrates the new dataset into the broader Web of Data, generating new schema mappings and transferable keys. This, in turn, allows the creation of new identity links. Therefore, by aggregating vocabularies, ontologies, and the datasets that use them, along with their schema mappings and class-level keys, and by offering both a web service and a SPARQL endpoint, KG Nexus goes beyond merely surfacing existing alignments to enable extendable, scalable, and transparent integration of diverse KGs on the Web.

6.3 . Creation of the KG Nexus

The construction of KG Nexus follows a multi-step workflow, illustrated in Figure 6.1. Each step is designed to address specific challenges identified in Chapters 5 and 6.1, namely the heterogeneity of vocabularies, the lack of scalable alignment mechanisms, and the absence of reusable key repositories. By systematically combining curated vocabularies, ontology alignments, dataset footprints, and reusable keys, KG Nexus provides an extensible infrastructure that supports explainable and reproducible alignment across heterogeneous Knowledge Graphs (KGs).

The workflow is structured around four main stages:

1. **Vocabulary collection, harmonization, and augmentation.** Starting from the Linked Open Vocabularies (LOV) repository and complemented with Wikidata, KG

Nexus collects a wide range of vocabularies. However, these vocabularies exhibit strong heterogeneity and varying degrees of completeness, which hinders their direct reuse. For example, the concept of a comment can be expressed through several properties such as $rdfs{:}comment$, $schema{:}comment$, or $dc{:}description$. To resolve such inconsistencies, KG Nexus applies a series of semi-automatic curation and harmonization techniques that consolidate equivalent predicates into canonical forms. This step directly addresses the fragmentation of metadata descriptions and ensures that vocabularies can be consistently compared, aligned, and reused across different KGs.

2. **Ontology alignment generation.** To address schema heterogeneity across vocabularies, KG Nexus generates ontology alignments using existing alignment tools. These tools not only establish correspondences between elements such as properties and classes, but also provide similarity scores that quantify the strength of each mapping. By retaining these scores, KG Nexus supports more flexible integration scenarios, including n–m correspondences, and enables users to assess the confidence of automatically generated alignments.

3. **Linking vocabularies to datasets.** To operationalize schema-level alignments, KG Nexus explicitly links vocabularies to the Knowledge Graphs that employ them to describe their data. In its current version, KG Nexus integrates three unique KGs, providing concrete use cases for alignment and key transfer. Additional datasets can be incorporated through a dedicated submission process available on the Web service, which requires contributors to provide provenance and descriptive metadata. This ensures that newly integrated resources can be verified, trusted, and reused by the community. By recording these connections, KG Nexus makes it possible to propagate alignments discovered at the vocabulary level to real-world datasets, thereby bridging abstract schemas with concrete graph instances.

4. **Incorporation of keys.** Finally, to promote reuse and avoid redundant discovery, KG Nexus stores and shares keys that have been previously identified. These keys can be directly leveraged by key-based entity linking tools such as [Fiorelli and Stellato(2020), Jiménez-Ruiz and Cuenca Grau(2011), Cotovio et al.(2024)], thereby reducing the computational cost of repeated discovery. This step implements the lessons learned in Chapter 5 and builds on early work such as [Symeonidou et al.(2014)]. By treating keys as reusable resources rather than isolated outputs, KG Nexus enables scalable and explainable entity alignment across multiple graphs.

These four steps will be detailed in the forthcoming sections. All the scripts used for creating the KG Nexus are publicly available on this Github repository[4].

[4] https://github.com/NexusKG/KG-Nexus

Type of Entity	Number of predicates used in LOV
Vocabularies	346
Properties	153
Classes	293

Table 6.1: Number of predicates per type of Entity.

6.3.1 . Vocabulary Reduction, Harmonisation and Augmentation

The vocabularies collected from sources such as the Linked Open Vocabularies (LOV) repository are highly heterogeneous, both in structure and completeness. To address this, KG Nexus applies a twofold strategy: first, reducing and harmonising existing vocabularies, and second, augmenting them with vocabularies extracted from Wikidata.

Reduction & Harmonization.

As mentioned in Section 6.2, the LOV is a meta-knowledge graph designed to collect and expose vocabularies from the Linked Open Data (LOD) Cloud. These vocabularies are maintained by their original creators and curated by volunteers. However, vocabulary descriptions remain largely unharmonized. In particular, metadata attributes for properties and classes, such as labels and comments, are expressed via a variety of predicates or marginally divergent IRIs (for example, using “https” versus “http,” or capitalizing “Label” rather than “label”). Such heterogeneity complicates the precise retrieval of information. Table 6.1 illustrates this challenge by enumerating the numerous distinct predicates employed to annotate properties and classes (excluding instances) within the LOV.

Furthermore, Tables 6.2 and 6.3 detail the number of predicates used in LOV to describe selected properties and classes. While some predicates are used consistently, such as `owl:differentFrom` which is the sole predicate found for expressing difference, this is not the norm. In contrast, the comment metadata, for example, is expressed using at least 12 different predicates for properties and 13 for classes. This fragmentation makes it difficult to automatically extract semantically relevant information from the vocabularies without extensive predicate inspection.

To address this, we manually identified and grouped semantically equivalent predicates, and harmonized the data by mapping them to a single canonical predicate for each type of metadata. While only the canonical predicates are retained in the harmonized representation, we also recorded the sets of equivalent predicates to maintain traceability and allow for further analysis if needed.

For example, one such harmonization task concerns label information, which we standardize across all vocabularies using the predicate: `http://www.w3.org/2000/01/rdf-schema#label`
In practice, the following predicates are all reduced to this single representation:

- `http://xmlns.com/foaf/spec/name`
- `http://xmlns.com/foaf/0.1/name`

Information expected	# predicates
Type	2
Label	5
Comment	12
Domain	2
Range	3
Domain Includes	8
Range Includes	7
Sub Property Of	5
Equivalent Property	4
Different From	1
Inverse Of	5
Category	1
Is Defined By	2

Table 6.2: Number of predicates describing properties

Information expected	# predicates
Type	2
Label	5
Comment	13
Sub Class Of	3
Equivalent Class	1
Different From	1
Category	1
Is Defined By	2

Table 6.3: Number of predicates describing Classes

- `http://purl.org/dc/elements/1.1/title`
- `http://comicmeta.org/cbo/qlabel`

This same harmonization process is applicable to any new ontology provided by a user. If the predicates used in a user-supplied ontology are already present in our predefined mapping (as established for the LOV vocabularies), the system will automatically apply the corresponding reductions and harmonizations. Otherwise, a simple configuration file can be used to define custom mappings for previously unseen predicates. This approach ensures flexibility while maintaining consistency across all integrated vocabularies.

Vocabulary Augmentation from Wikidata.

Beyond harmonizing existing LOV vocabularies, KG Nexus also integrates vocabularies that are not explicitly published, most notably Wikidata. Since Wikidata does not provide a dedicated vocabulary file, we developed an extraction process to retrieve properties and classes directly from a Wikidata dump generated in May 2023. The process proceeds in two iterations: first, properties are collected via the predicate $wikibase{:}propertyType$; second, classes are identified through triples of the form $\langle ?subject,\ \ wdt{:}P31\,(instanceOf), ?object\rangle$. To preserve class hierarchies, we then retrieve additional classes connected by $wdt{:}P279\,(subClassOf)$, where the object represents the superclass of the previously identified class. This augmentation step ensures that Wikidata's rich and widely adopted vocabulary is consistently represented within KG Nexus, complementing the curated vocabularies from LOV.

Through these SPARQL-based iterations, we retrieved a total of 96,433 classes and

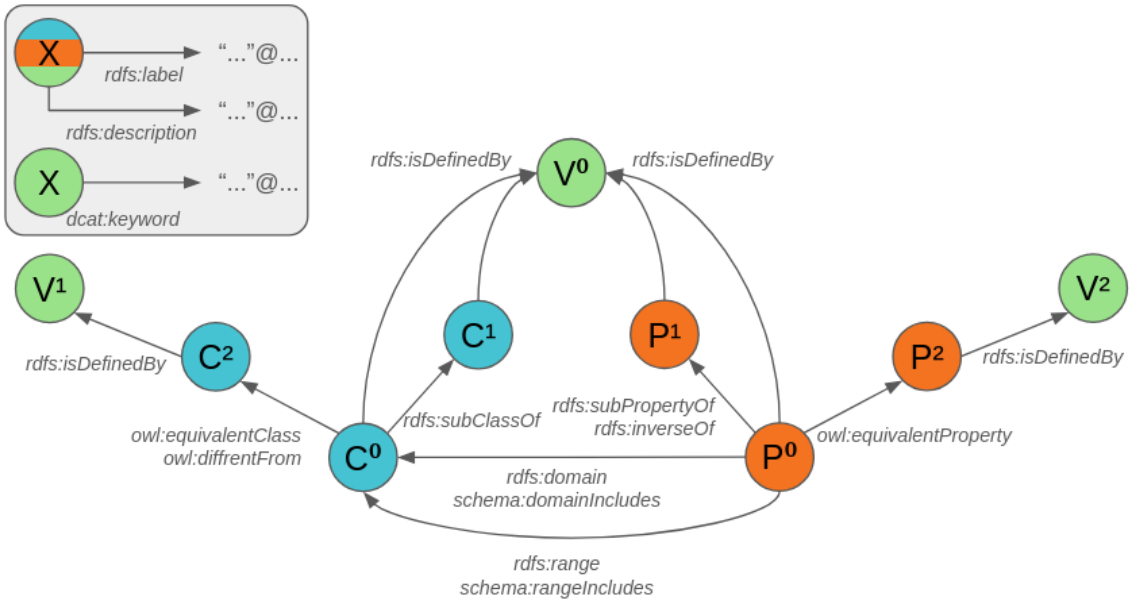


Figure 6.2: Schema of KG Nexus vocabulary representation (LOV-RHA)

11,002 properties. Additionally, we extracted all associated lexical forms in which these entities are described within the knowledge graph.

Metadata Schema for KG Nexus Vocabulary Representation.

Figure 6.2 presents the schema used to describe the metadata of vocabularies integrated into KG Nexus, constructed in Step 1 (i.e., LOV-RHA). The schema comprises three types of entities, represented by green (to designate vocabularies), blue (for classes), and orange nodes (for properties). Each property and class is linked to its corresponding vocabulary using the predicate `rdfs:isDefinedBy`. The class and property hierarchies are captured using `rdfs:subClassOf` and `rdfs:subPropertyOf`, respectively. Domain and range information is represented using `rdfs:domain` and `rdfs:range` for LOV vocabularies, and `schema:domainIncludes` and `schema:rangeIncludes` for Wikidata properties. We also retain terminological metadata, including keywords, descriptions, and language tags derived from the original ontologies.

6.3.2 . Transformer-based Ontology Alignment

To mitigate schema heterogeneity across vocabularies (or ontologies), the second step of our framework focuses on computing an initial set of mappings between properties and classes across two vocabularies, denoted as V_1 and V_2. The goal is to support federated querying by suggesting semantically similar elements from a second KG.

To achieve this, we utilize the *Universal Sentence Encoder (USE)* [Cer et al.(2018)], a pre-trained language model designed to generate embeddings for natural language text. We extract semantic information, namely *labels* and, when available, *comments*, from the

ontology elements and encode them using USE. Since USE is trained exclusively on English data, we restrict our analysis to labels and comments in English. For ontologies in other languages, a language-specific model or adapted multilingual variant should be used, as proposed in [Marchisio et al.(2023)]. This modularity allows the alignment component to evolve with advances in multilingual language models.

Once embeddings are generated, we compute *cosine similarity* scores between each pair of properties and classes across the two ontologies. These scores are computed individually for each semantic feature (e.g., label, comment), and a final similarity score is derived by averaging the available values. Optional information (such as comments) is excluded from the average if missing. This modularity allows the alignment component to evolve with advances in multilingual language models.

Only ontology element pairs whose similarity score exceeds a predefined threshold are retained in the Nexus. These alignments are encoded using *RDF-Star*[5] as *quadruples*, following the structure: $\langle C1/P1,\ kgnx{:}similarityScore,\ C2/P2,\ kgnx{:}similarityScore_X_Y\rangle$, where X and Y denote the interval in which the similarity value falls. This ensures that alignments are not only reusable but also transparent, as users can evaluate their reliability based on the associated scores.

6.3.3 . Equivalence Relationships Computation

In the previous step, we computed the similarity scores for every pair of properties and classes between the considered ontologies. We now need to identify the best alignments among these scores. Hence, we designed a conservative matching procedure that applies 1-to-1 alignment in most cases. The process begins by storing the similarity scores for all possible pairs and sorting them in descending order of similarity. A global threshold is then applied to filter out weak matches: only pairs with a score above this threshold are considered valid alignments. This threshold can be set within the interval $[0, 1]$, allowing users to adjust the trade-off between precision and recall depending on their requirements.

After this initial filtering step, we proceed greedily from the highest similarity score and align each pair while tracking previously aligned elements to avoid duplicate matches. To address structural heterogeneity, such as when a single property in one ontology corresponds to multiple properties in another, we allow $1{-}to{-}n$ alignments based on a relative threshold (RT). Specifically, if a property P aligns with P'_1 with a similarity score SV_1, and another property P'_2 has a score SV_2 such that $SV_2 \geq SV_1 - RT$, then both P'_1 and P'_2 are considered valid alignments for P. This mechanism allows to capture complex alignment scenarios, such as a geographic location described by a single property in one ontology and split into latitude and longitude in another. Similarly, multiple Wikipedia properties might represent a concept like a person's birth date. These situations exemplify complex property alignments as shown in this survey [Thiéblin et al.(2020)].

For class alignments, we apply the same general procedure, with the exception that we

[5] https://w3c.github.io/rdf-star/cg-spec/editors_draft.html

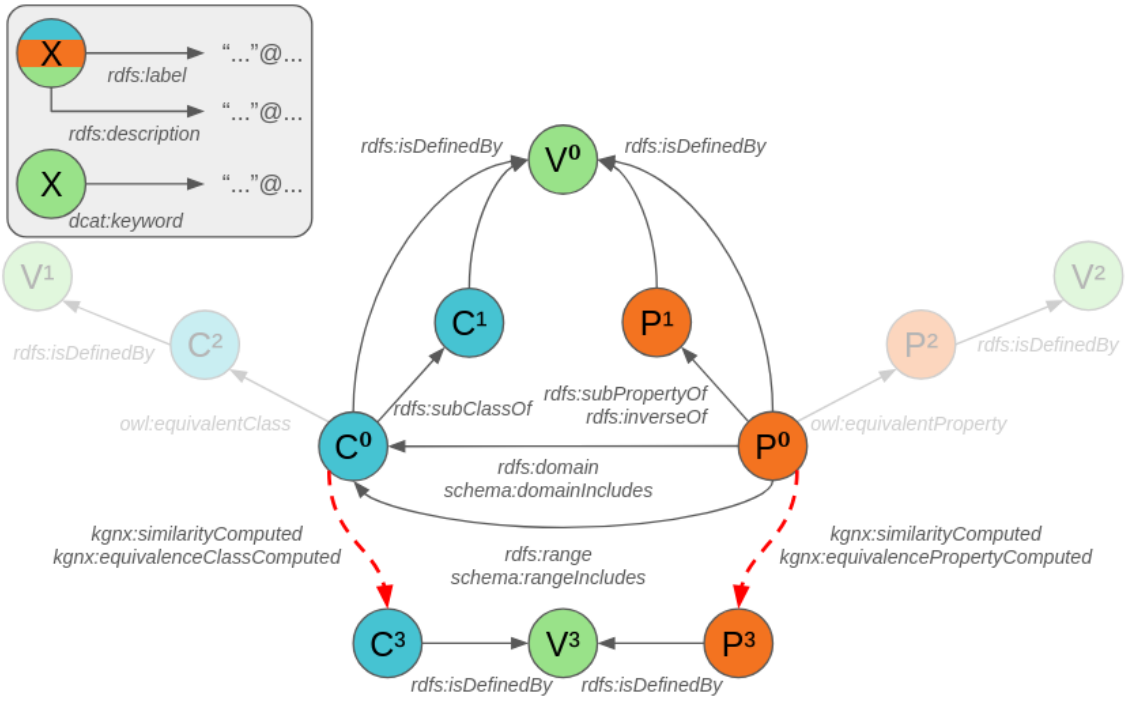


Figure 6.3: Similarity and ontology alignment representation in KG Nexus

enforce strict 1-to-1 mappings. In this case, no relative threshold is applied, as class-level semantics are typically more abstract and less granular than properties, making simpler alignment strategies more reliable.

The resulting equivalence relationships are represented within the KG Nexus using two new dedicated predicates:

- Prefix `kgnx`: $\langle http://kgnexus.lisn.upsaclay.fr/nexus/ \rangle$
- Properties: $kgnx$:$equivalencePropertyComputed$
- Classes: $kgnx$:$equivalenceClassComputed$

We intentionally avoid using standard OWL predicates such as owl:$equivalentClass$ or owl:$equivalentProperty$, as our mappings are generated automatically and not curated by domain experts. Therefore, they should be interpreted as approximate, with a degree of uncertainty. Figure 6.3 illustrates how similarity scores and equivalence mappings are modeled in the KG Nexus using the properties we introduce to encode these semantic correspondences, enabling a structured and extensible representation of ontology alignments within the KG, without generating logical inconsistencies.

6.3.4 . Bring your own Alignment Tool

While baseline alignments in KG Nexus are computed using semantic embedding such as labels and comments. While effective for initial mappings, this approach does not leverage the full richness of ontological structures available in many vocabularies.

State-of-the-art ontology alignment systems often incorporate additional signals beyond textual descriptions, such as:

- **Hierarchical relationships**, such as $rdfs{:}subClassOf$, which reflect the taxonomy of classes and help identify semantically related elements.
- **Property constraints**, including $rdfs{:}domain$ and $rdfs{:}range$, which provide structural context for properties.
- **Predefined equivalence links** between vocabularies, which may already encode trusted mappings created by domain experts.
- **Ontological context and instance-level usage**, which can be leveraged to detect patterns or overlaps not visible in metadata alone.

To support such advanced alignment strategies, we designed KG Nexus to be fully extensible. In particular, we provide a retrieval script that allows users to generate a configuration file specifying i) the vocabularies of interest; and ii) the types of semantic and structural information required by their alignment tool. Based on this configuration, the script retrieves and exports the relevant ontologies in OWL format, enriched with all necessary metadata. Users can then apply their preferred ontology alignment algorithm on top of this data. Finally, the results of these custom alignment processes can be reintegrated into KG Nexus by using the same schema and predicates (e.g., $kgnx{:}equivalencePropertyComputed$). By allowing users to reintegrate the outputs of their preferred alignment systems into a common schema, KG Nexus becomes more than a static repository. It acts as an open hub for comparing, aggregating, and reusing alignments produced with diverse methodologies, thereby fostering innovation and interoperability in ontology matching research.

6.3.5 . Knowledge Graph Incorporation

To bridge abstract vocabularies with concrete datasets, KG Nexus explicitly records which properties and classes are used by each knowledge graph. This connection enables schema-level alignments and keys to be propagated directly to real data, making them actionable for entity alignment tasks. We rely on the PROV-O ontology[6], which allows us to trace provenance and usage relationships. Specifically, for a given knowledge graph G_1, we extract all the properties and classes it uses. As illustrated in Figure 6.4, each of these elements is then linked to G_1 using the `prov:used` predicate. This set of links defines the footprint of G_1 within KG Nexus.

[6] https://www.w3.org/TR/prov-o/

By recording these footprints, KG Nexus makes it possible to compare knowledge graphs in terms of the vocabularies they rely on. It also facilitates the reuse of previously computed equivalences and similarity scores to support interoperability between datasets with different but related ontological bases.

6.3.6 . Key Incorporation

Building on the findings of Chapter 5, KG Nexus treats keys as central components of the alignment process. Keys discovered by tools such as SAKey [Symeonidou et al.(2014)] or Rocker [Soru et al.(2015)] are stored together with provenance metadata, including their dataset of origin, the tool used for discovery, and its parameters. This guarantees both transparency and reproducibility. In KG Nexus, a key is considered valid for a specific class and consists of a set of properties which may be datatype or object properties. As illustrated in Figure 6.4, each discovered key is linked to the knowledge graph from which it was derived using the PROV-O predicate `prov:wasDerivedFrom`. The tool used for discovery and its associated hyperparameters are also recorded to ensure traceability. Furthermore, the predicate `prov:used` is used to associate the key with both the class it applies to and the set of properties it comprises.

As shown in Chapter 5, keys are often transferable across KGs with limited loss of quality. KG Nexus makes this transfer explicit through the predicate `kgnx:wasTransferredTo`, which links a discovered key to the datasets where it can be applied. This avoids repeated discovery and significantly improves scalability for entity alignment.

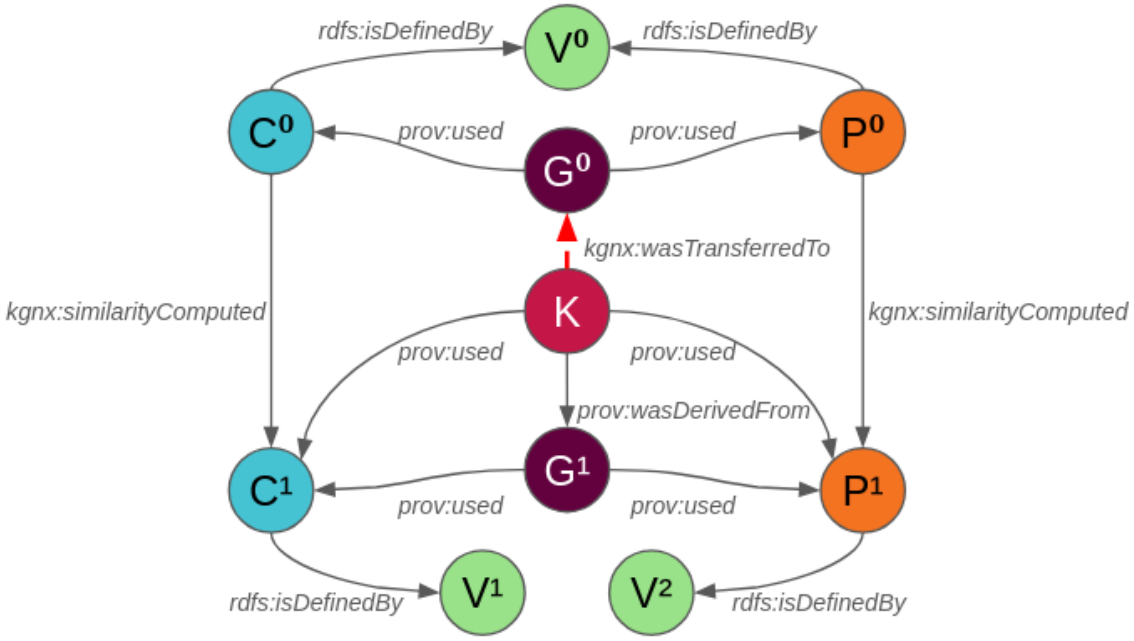


Figure 6.4: KG and Keys incorporation in KG Nexus

Although designed for keys, this incorporation mechanism is extensible to other forms of linking knowledge. For example, KG Nexus could store the best-performing similarity

measure (e.g., Levenshtein, cosine similarity, Jaro-Winkler) for a given property, or rules that have been validated across datasets. This generalization positions KG Nexus as a flexible infrastructure for capturing and reusing alignment knowledge in diverse forms.

6.4 . Data analytics

6.4.1 . Raw information of the vocabularies

To understand the coverage and diversity of KG Nexus, we first analyzed the raw information contained in the integrated vocabularies. This step is important because the scale, variety, and linguistic distribution of vocabularies directly impact the ability of KG Nexus to support alignment across heterogeneous datasets. In total, we retrieved and catalogued 940 vocabularies, encompassing 55,232 properties and 132,499 classes. The top five vocabularies by size are presented in Table 6.4, with Wikidata emerging as the largest contributor, representing 20% of all properties and 72% of all classes. This confirms its central role as a reference resource in the Web of Data and justifies its integration into KG Nexus.

Prefix Vocabulary	# Classes
wd	96,433
obo	3,675
dicom	1,592
eupont	1,578
sio	1,521

Prefix Vocabulary	# Properties
wd	11,002
dicom	7,033
dbpedia-owl	2,713
ifc	1,574
datex	1,336

Table 6.4: Top five vocabularies with respect to the number of Classes or Properties.

Another striking characteristic of KG Nexus is its linguistic diversity. While English is the most frequently used language, only 594 out of 940 vocabularies are described exclusively in English. Overall, the dataset covers 519 distinct languages or dialects, including regional variants such as American and British English. This diversity is highly valuable, as it enables KG Nexus to support cross-lingual alignments, particularly through transitive connections to Wikidata, which itself covers 496 languages (over 90% of those present in KG Nexus). Consequently, the multilingual dimension of the dataset enhances its applicability in global and multilingual contexts, going beyond resources that are limited to English-only descriptions. Consequently, vocabularies can be linked to Wikidata and, through transitive alignment, to others regardless of language differences. The top five languages are shown in Table 6.5.

Prefix Language	# Vocabularies
en	564
Undefined	502
es	61
fr	60
it	51

Prefix Language	# Description
en	91,861
Undefined	37,491
fr	28,246
uk	23,995
de	22,002

Table 6.5: Top 5 languages with respect to the number of Vocabularies or Description.

6.4.2 . Alignments Statistics

We next examined the alignments produced between vocabularies. After applying the transformer-based alignment tool, 39,654 pairs of vocabularies (9.7% of the total) were found to share at least one similarity score above the minimum threshold or an equivalence relation between properties or classes. Furthermore, more than half of all vocabularies (56.3%) are aligned with at least one other vocabulary. These results indicate that a substantial portion of the vocabulary space is interconnected, validating the role of KG Nexus as a hub where isolated resources can be linked together. At the same time, the fact that nearly half of the vocabularies remain unaligned highlights the ongoing need for extensible mechanisms, such as custom alignment tools (Sub-section 6.3.4), to expand coverage.

6.4.3 . Incorporation of Knowledge Graphs and Key Discovery

To connect schema-level information with real data, KG Nexus incorporates several large-scale knowledge graphs. Earlier versions of DBpedia (October 2016) and Wikidata (March 2021) were retrieved from RDF HDT repositories[7], complemented with more recent dumps such as Wikidata (May 25th, 2023), DBpedia (January 26, 2023), YAGO4 (January 24, 2023), and YAGO4.5 (August 3, 2023). For these datasets, keys were discovered using the SAKey algorithm, with results summarized in Table 6.6. For example, more than 143,000 keys were identified in DBpedia (2023), while over 11,000 keys were extracted from Wikidata (2023). For other datasets, transferable keys are currently reused instead of recomputing them.

These statistics illustrate both the scale and practicality of KG Nexus: rather than treating vocabularies as abstract objects, the system anchors them in concrete datasets and precomputes reusable keys. This not only reduces the computational burden of repeated discovery but also ensures that alignment pipelines can directly benefit from high-quality, ready-to-use keys.

[7] https://www.rdfhdt.org/datasets/

Dataset	# of Keys
DBpedia (2023)	1,843,312
Wikidata (2021)	9,117
Wikidata (2023)	11,496

Table 6.6: Number of keys identified per dataset using the SAKey algorithm.

6.5 . Use Cases

To illustrate the practical value of KG Nexus, we describe several possible representative use cases enabled by its integration of vocabularies, alignments, and keys. These scenarios demonstrate how KG Nexus facilitates data enrichment, consistency verification, concept analysis, and query optimization, while also supporting benchmarking and schema bootstrapping. Each use case highlights a different aspect of the platform, showing how previously isolated tasks in knowledge graph management can be streamlined and made more reproducible. Additionally, example SPARQL queries for these use cases are provided in the Github repository[8] to help users easily reproduce and build upon these scenarios.

6.5.1 . Knowledge Graph Enrichment

One of the primary applications of KG Nexus is semantic enrichment and data fusion across multiple KGs. Enrichment is crucial when datasets provide complementary but non-overlapping information, which often occurs due to schema differences. This is made feasible by storing the keys for each KG, and the mappings between vocabularies. For example, consider linking scholarly articles described in microdata extracted from WebDataCommons [Mühleisen and Bizer(2012)], typically annotated using Schema.org[9], with entities in Wikidata. While microdata are described using the class $schema{:}ScholarlyArticle$ and properties such as $schema{:}doi$ to identify articles, Wikidata uses the class $wd{:}Q18918145$ (Academic Journal Article) and alternative identifiers such as $wdt{:}P3894$ (OSTI article ID). Since Wikidata does not include a DOI for academic journal articles, users can exploit alternative stored keys in KG Nexus such as {title, author, year} and leverage mappings between these classes and properties, to establish identity links between entities, even when they use different schemas. This enables easier fusion of heterogeneous data sources including linking Web pages to their correspondent entities in the Web of Data. Furthermore, because KG Nexus integrates multiple interlinked KGs, it can facilitate the creation of large equivalence classes (i.e. set of web entities referring to the same real-world entity) that consolidate knowledge from multiple sources for improving data completeness.

[8] https://github.com/NexusKG/KG-Nexus
[9] https://schema.org

6.5.2 . Schema Consistency Verification

KG Nexus also facilitates large-scale schema-level consistency analysis by providing harmonized and interlinked vocabularies. It makes class and property hierarchies explicit and integrates mappings across ontologies, enabling systems to efficiently perform automated checks for logical inconsistencies and structural issues. For instance, works such as [Wang et al.(2021)], which detect cycles in $rdfs{:}subClassOf$ or $rdfs{:}subPropertyOf$ relations, can now leverage KG Nexus as a centralized source for these relationships, avoiding the significant overhead and complexity involved in extracting them directly from the Web. This significantly streamlines such analyses and enhances the quality assurance of the Web of Data, particularly in reasoning-intensive applications.

6.5.3 . Concept Shift Analysis

Our resource allows researchers to analyse how the same concept differs in vocabularies, enabling the study of conceptual evolution and divergence over time and across application domains. By focusing on vocabulary and ontology mappings, KG Nexus facilitates the identification of shifts in modelling practices throughout the Web of Data. For instance, consider the representation of events in two widely used vocabularies: the Event Ontology ($event{:}Event$) and Schema.org ($schema{:}Event$). Although both refer to real-world events, they diverge significantly in purpose and structure. While $event{:}Event$ is designed for rich temporal narratives, making it suitable for cultural heritage and scholarly data, $schema{:}Event$ focuses on public-facing events with attributes like ticketing, location, and performers, optimized for discoverability and Search Engine Optimization (SEO) in web searches and applications. Through KG Nexus, such differences can be systematically compared across datasets, allowing users to study how modelling choices align with application domains, and how concepts are reinterpreted across time.

6.5.4 . Federated Query Optimization

In federated query scenarios, where data is distributed across multiple endpoints, a common challenge lies in composing queries that operate over heterogeneous schemas. KG Nexus addresses this issue by providing access to vocabulary alignments that can be used for query rewriting. By leveraging mappings between equivalent properties, systems can rewrite user queries to access semantically similar data across endpoints, thereby improving recall and interoperability. For instance, a query over DBpedia requesting $dbo{:}birthDate$ can be rewritten to include $schema{:}birthDate$ or $wdt{:}P569$ from Wikidata. Such query rewriting increases interoperability and improves recall without requiring manual intervention, making federated queries both more effective and more user-friendly.

6.5.5 . Key Quality Profiling and Benchmarking

Developers of entity matching pipelines often need to evaluate the quality and robustness of potential keys. KG Nexus provides a large collection of keys, annotated with metadata such as support, discriminability, and provenance. This enables users to identify high-quality keys by class, vocabulary, or domain which is useful for benchmarking

or designing linking strategies tailored to specific datasets. For example, before applying a rule to align instances of $foaf{:}Person$, one can retrieve keys associated with person classes in DBpedia or Wikidata and evaluate their robustness across datasets. This capability enables more informed and reproducible design of entity linking pipelines.

6.5.6 . Automatic Schema Bootstrapping for New KGs

Publishing a new KG often requires selecting appropriate classes and properties, a process that is prone to redundancy and misalignment with existing vocabularies. KG Nexus addresses this challenge through *schema bootstrapping*, by automatically suggesting related classes, properties, and reusable keys derived from existing alignments. For instance, a new agronomic dataset employing a class equivalent to $obo{:}Plant$ can be seamlessly linked to AgroPortal, Wikidata, or DBpedia. By exploiting transitive alignments and key reuse, the dataset can be rapidly integrated into the Web of Data, ensuring interoperability from the outset while reducing the manual effort required for schema design.

6.6 . Conclusion

This chapter introduced *KG Nexus*, a curated and extensible resource designed to overcome key limitations in current alignment workflows for heterogeneous Knowledge Graphs. By harmonising vocabularies, computing ontology alignments, and storing reusable keys, KG Nexus consolidates alignment knowledge that was previously scattered across isolated tools and datasets. This integration reduces computational costs, supports explainable workflows, and promotes reproducibility in KG research and applications.

The contributions of KG Nexus are fourfold. First, it provides a harmonised representation of vocabularies from diverse sources, ranging from curated repositories such as LOV to community-driven resources such as Wikidata. Second, it offers schema-level alignments derived from transformer-based embeddings, while remaining open to the integration of external alignment systems. Third, it explicitly links vocabularies to the datasets that employ them, enabling schema-level insights to be applied directly to real-world graphs. Fourth, it builds on the lessons of Chapter 5 by treating keys as reusable resources: once discovered, they are preserved with provenance and marked as transferable across datasets. In doing so, key discovery results cease to be ephemeral outputs and instead become shared knowledge assets.

The use cases presented in Section 6.5 demonstrate the versatility of the platform. They show how consolidating vocabularies, alignments, and keys within a single infrastructure can streamline enrichment, improve schema consistency verification, enable query rewriting, and support schema bootstrapping. More broadly, KG Nexus illustrates how reusable alignment knowledge can serve as a foundation for scalable, explainable, and sustainable knowledge graph integration. As such, it represents not only a practical resource for current applications but also a methodological step toward treating alignment outputs as long-lived assets in the evolving Web of Data.

6.7 . Future Work

Looking forward, several extensions can strengthen KG Nexus further.

6.7.1 . Multilingual Support

A first direction for future work is to strengthen multilingual support. Many knowledge graphs, such as Wikidata, already provide rich multilingual descriptions of entities, which can serve as bridges between underrepresented language resources and larger graphs described in English or other widely used languages. By exploiting these multilingual labels and descriptions, it becomes possible to align entities across knowledge graphs even when they are primarily documented in different languages. Beyond leveraging existing resources, the integration of language-specific or domain-adapted embedding models would also allow the semantic content of a KG to be directly projected into another linguistic space, enabling more robust cross-lingual alignments. Such translations or embeddings could be stored within KG Nexus together with provenance metadata, allowing peers to evaluate and validate them. This would ensure that information and meaning are preserved during the translation process, while also fostering community-driven quality control for multilingual interoperability.

6.7.2 . Exploration through Dashboards

A second direction is to develop interactive dashboards connected to KG Nexus in order to facilitate intuitive exploration of the data. Such interfaces would allow users to navigate across multiple sources of information, directly compare alignments produced by different tools, and visualize ontologies side by side to better understand their similarities and divergences. Beyond exploration, dashboards could also provide mechanisms for experts and community contributors to enrich KG Nexus, for example by validating or refuting candidate keys, refining schema mappings, or contributing new alignments. This interactive layer would support the use cases described in Section 6.5, while also fostering transparency, reproducibility, and collaborative validation of the knowledge contained in KG Nexus.

6.7.3 . Transversal Applications

A third direction is to extend the reuse paradigm beyond keys in order to encompass other forms of alignment and ontology knowledge that are valuable for a variety of research tasks. As discussed in Chapter 3, one promising avenue is the incorporation of temporal constraints within the Nexus. Such constraints, when combined with property alignments available in KG Nexus, would make it possible to transfer temporal knowledge from one graph to another. This transfer capability would facilitate the validation of temporal consistency across multiple graphs without requiring the costly rediscovery of temporal constraints for each new dataset. Beyond constraints, ontology-level insights such as equivalence relations, hierarchical structures, or validated alignment rules could be systematically integrated. By making these heterogeneous resources accessible through a unified and queryable platform, KG Nexus could serve not only as a repository

of reusable assets but also as a transversal research environment, supporting investigations into knowledge evolution, ontology engineering, and the broader dynamics of the Web of Data.

Chapter 7

Conclusion and Perspectives

7.1 . Conclusion

The work presented in this dissertation has explored the integration of temporal fact validation and robust linking mechanisms in Knowledge Graphs (KGs) in order to address the following research question: **how can reasoning approaches validate the temporal consistency of information across multiple and heterogeneous Knowledge Graphs**. Motivated by the limitations of current practices, this work proposed novel frameworks, benchmarks, and resources designed to strengthen the reliability, scalability, and explainability of reasoning over dynamic and heterogeneous data. Altogether, these contributions converge toward a unified objective, namely making temporal and structural knowledge in KGs both interpretable and operational at scale.

The first contribution introduced an **explainable framework for temporal fact validation**, grounded in the automatic discovery of temporal constraints between entity properties. By extending Allen's algebra from intervals to ordered sequences, we modelled complex dependencies such as `before(graduatedAsDoctor, isSurgeonIn)` and used them to verify the temporal consistency of facts. Two complementary strategies were developed: a symbolic validation mechanism, offering interpretable explanations through constraint satisfaction, and a machine-learning based aggregation method, ensuring robustness and adaptability in large-scale graphs. Experiments demonstrated that temporal consistency can be effectively captured from data, confirming the feasibility of explainable temporal validation while revealing the inherent trade-off between coverage, efficiency, and transparency. Together, these elements define a coherent methodology for the systematic evaluation of temporal fact validation, specifying how constraints are mined, how they are applied for verification, and which quantitative indicators are used to measure temporal soundness.

Building upon these findings, the second contribution proposed a **benchmark dedicated to temporal validity in Knowledge Graphs**. Unlike traditional completion tasks, this benchmark focuses on factual soundness by generating controlled temporal corruptions, such as inconsistent intervals or misordered events, and evaluating models on calibration, granularity sensitivity, scalability, and genuine temporal reasoning. The result-

ing datasets expose fundamental weaknesses of existing models, which tend to exploit temporal patterns rather than reasoning over them. Released as an open resource, this benchmark provides a standardized basis for fair comparison and reproducibility, thereby encouraging the development of models that treat temporal validity as a primary dimension of trustworthiness in KGs.

The third contribution investigated the **transferability of keys** as a means to improve scalability in data linking. We formalized a strategy that allows keys discovered in one KG to be rewritten and validated in another via property alignments. This approach eliminates redundant discovery phases, achieving substantial runtime gains while preserving alignment quality. Experiments confirmed that key transfer can generalize across heterogeneous vocabularies when supported by robust schema mappings. Beyond its practical impact, this study redefined keys as reusable knowledge assets and connecting symbolic interpretability with hybrid learning frameworks.

Finally, the fourth contribution materialized these conceptual advances through the design of **KG Nexus**, a curated and extensible infrastructure for knowledge alignment. By consolidating vocabularies, ontology alignments, and transferable keys into a unified resource, KG Nexus operationalizes the idea of alignment as a reusable and explainable component of the Web of Data. It facilitates tasks such as schema bootstrapping for new KGs, enrichment of existing datasets, and cross-graph query rewriting. More broadly, it embodies a transition from isolated alignment workflows toward a sustainable and interoperable ecosystem for KG integration.

Taken together, these contributions outline a coherent progression, from the discovery of temporal regularities to their evaluation, reuse, and operationalization within a shared infrastructure. They reveal two central challenges for Knowledge Graphs: (i) incorporating temporality into fact validation while preserving explainability, and (ii) ensuring robust, scalable, and reusable alignment across heterogeneous datasets. Addressing these challenges, the proposed methods demonstrate that symbolic knowledge, whether expressed as temporal constraints or linking keys, can be discovered and validated in both forms, and that transfer is achieved for linking keys, thereby enhancing the reliability of reasoning in complex settings. Beyond their individual scope, the different approaches illustrate how symbolic representations and learning-based methods can be combined in a complementary way, leveraging the strengths of each to achieve both interpretability and adaptability. This convergence reflects a broader movement in artificial intelligence toward more integrated reasoning frameworks, in which Knowledge Graphs play a central role. A recurring lesson throughout this research is the importance of reusability: temporal constraints, whose transferability remains to be explored in future work, benchmarks, and keys should not remain isolated outputs but evolve into long lived assets that foster reproducibility, interoperability, and cumulative progress in Knowledge Graph research.

7.2 . Active Collaboration: Deep Temporal Reasoning with Allen Algebra

During this thesis, I had the opportunity to collaborate with Assistant Professor Michael Cochez[1] during a six-month international research mobility at the L&R Team of the Vrije Universiteit Amsterdam. This collaboration initiated a new line of research that is currently under active development and aims to bridge symbolic temporal reasoning and neural representation learning.

The ongoing work focuses on designing a deep learning framework for the task of Temporal Knowledge Graph Completion (TKGC). Building upon recent advances such as [Chen et al.(2023b), Geng et al.(2023)], which demonstrate the potential of relational and meta-learning strategies for temporal reasoning, our approach diverges from models such as Timplex or TNTComplex (presented in Chapter 4) that explicitly parameterize temporal information through continuous embeddings or discrete timestamps (as shown in Chapter 4). Instead, we leverage the qualitative temporal relations defined by Allen's Interval Algebra (see Subsection 2.2.3), with the motivation of mitigating the scalability issues commonly observed when temporal dimensions are modeled as continuous variables, particularly in datasets characterized by long or irregular temporal spans.

In the proposed framework, temporal reasoning is handled implicitly through the structure of the graph. Specifically, the knowledge graph is first transformed from an entity-centric to a relation-centric representation, where entities become edges and relations are represented as nodes. This transformation allows the model to directly reason over relations rather than over entities, thereby emphasizing the temporal and structural dependencies between relational instances.

Once the graph is restructured, entities on the edges are replaced with two categories of markers. The first type, *structural markers*, preserves information about the connectivity and the role of entities within the graph through the use of two possible relations, namely $(Head\text{-}Tail)$ and $(Tail\text{-}Head)$. The second type, *temporal markers*, encodes the qualitative temporal context derived from the thirteen relations defined by Allen's Interval Algebra. This transformation results in a homogeneous relational graph where both structural and temporal dimensions are represented within the same topological space, allowing the model to reason uniformly across these two axes of information. An illustrative example of this preprocessing step is provided in Figure 7.1.

Two Relational Graph Convolutional Networks (RGCNs [Schlichtkrull et al.(2018a)]) are then employed. The first learns latent representations from the structural neighborhoods of relations, capturing the topological context and co-occurrence patterns between them. The second focuses on temporal neighborhoods, learning representations that reflect the qualitative temporal dependencies among relations. The embeddings obtained from both RGCNs are subsequently combined to form a unified representation for each unique relational instance. Finally, a classical DistMult[Yang et al.(2015)]-based scoring function, as introduced in the original RGCN formulation, is applied to compute the plausibility of candidate facts and complete the temporal knowledge graph.

[1] now full professor at Åbo Akademi University, Finland

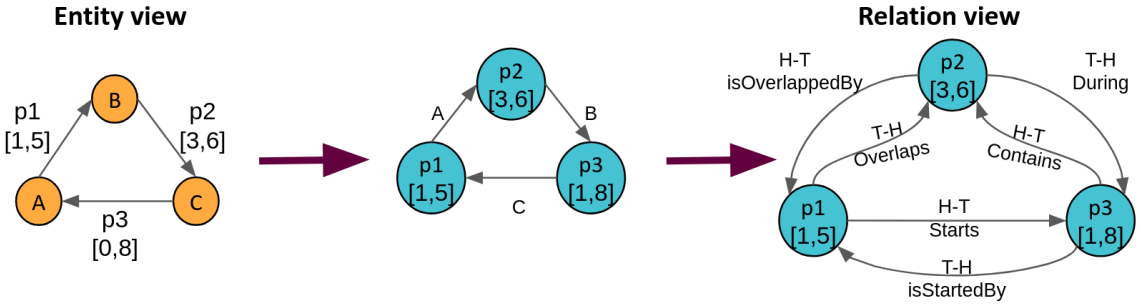


Figure 7.1: Example of transformation from an entity view to a relational view

At the current stage of development, this line of research remains exploratory. Preliminary experiments have been conducted, but the resulting performance is not yet competitive with the state of the art. For this reason, and because the framework is still undergoing active refinement, no detailed results are presented in this manuscript. Instead, the following discussion focuses on the insights gained from these early trials and on the directions identified for improving the proposed approach.

This work aims to evaluate whether qualitative temporal abstractions, such as those provided by Allen's relations, can offer an effective balance between expressiveness and computational efficiency. The preliminary results obtained so far, however, remain below the current state of the art. We hypothesize that this limitation may stem either from implementation issues in the current prototype or from an intrinsic weakness in the framework itself. While the conceptual formulation is sound and intuitively appealing, it may not provide sufficiently precise supervision signals for the model to learn accurate temporal boundaries, thereby limiting its predictive performance on temporal completion tasks.

To address these limitations, we plan to refine the framework by introducing a finer temporal granularity through a discretization of Allen's relations into multiple *buckets*. Indeed, a classical *before* relation may encompass vastly different temporal scales, ranging from events separated by minutes to those separated by centuries, which can obscure meaningful temporal distinctions. By subdividing such relations into categories reflecting their relative proximity (e.g., *slightly before*, *moderately before*, *long before*), we aim to enhance the expressiveness of the model and improve its ability to capture subtle temporal dependencies. We expect this refinement to yield richer temporal representations, improve overall predictive accuracy, and lead to more robust temporal reasoning across diverse datasets.

A further limitation may also arise from the architectural backbone itself, namely the RGCN. As discussed in the extended study presented in [Thanapalasingam et al.(2022)], RGCNs architecture have shown a consistent performance gap compared to state-of-the-art models for the task of Knowledge Graph Completion. This structural limitation could therefore contribute to the preliminary results observed in our framework. Nevertheless, even if performance on the completion task is limited, the model can still capture relational and temporal regularities. These representations can be reused for subsequent

tasks such as node classification or link clustering, where Relational Graph Convolutional Networks have shown strong generalization. This perspective will guide future evaluations aimed at better characterizing the scope and transferability of the learned representations.

7.3 . General Perspectives

7.3.1 . From Temporal Reasoning to Hybrid Intelligence

The results presented in this thesis highlight that temporal reasoning is not only a constraint-satisfaction problem but also a form of structured intelligence that can guide learning systems. As the boundaries between symbolic and neural approaches continue to blur, a key perspective lies in the emergence of hybrid architectures capable of combining the robustness of data-driven inference with the interpretability of explicit rules. In such systems, temporal constraints and transferable keys would no longer serve merely as post-hoc validators but as priors that shape the reasoning space of neural models. This integration would allow representations to evolve dynamically with time, enabling predictive models to learn not only from facts but also from their temporal dependencies.

Temporal Knowledge Graphs provide an ideal substrate for this hybridization. By encoding factual evolution and temporal scope, they can inform models when knowledge becomes obsolete or contextually invalid. Embedding these mechanisms within Large Language Models could foster temporally aware reasoning where generative capabilities are constrained by explicit validity conditions. More broadly, this work treats Knowledge Graphs as supports for learning. They organize facts, provide context, and add simple constraints so that model outputs remain temporally consistent and explainable.

7.3.2 . Integrating Human Expertise in the Validation Loop

As Knowledge Graphs grow in complexity, the role of human expertise remains irreplaceable. Automated reasoning can detect inconsistencies or suggest hypotheses, but only human interpretation can resolve semantic ambiguities and ensure that inferred knowledge aligns with domain understanding. The next generation of reasoning frameworks should therefore embrace interactive paradigms in which experts collaborate with algorithms through continuous feedback cycles.

In such human-in-the-loop systems, experts would not act as external validators but as co-creators of the evolving logic that governs the graph. Visualization tools could highlight temporal inconsistencies or conflicting facts, while explainable reasoning modules would allow experts to refine or reject inferred constraints. In return, these refinements would be formalized and reintroduced into the Knowledge Graph, progressively improving its precision and coverage. This iterative process would bridge the gap between machine inference and human interpretation, turning validation into a participatory form of reasoning that enhances trust, accountability, and long-term knowledge quality.

7.3.3 . Toward Sustainable and Open Knowledge Infrastructures

The sustainability of Knowledge Graph research depends on treating knowledge artifacts such as datasets, constraints, alignments, and benchmarks as shared and continuously evolving resources. Beyond methodological innovation, lasting progress in this field requires the consolidation of infrastructures that ensure interoperability, longevity, and transparent evaluation. This vision extends the collective efforts that shaped the Semantic Web through initiatives like the Linked Open Data Cloud and Linked Open Vocabularies, which established openness and reuse as central values of knowledge engineering. Building upon these foundations, our initiatives KG Nexus represents a new step in this continuum, aiming to connect vocabularies, alignments, and transferable reasoning components within coherent and community-maintained ecosystems.

Embedding such resources within FAIR-compliant platforms would ensure that they remain findable, accessible, interoperable, and reusable across projects and institutions. Yet, achieving sustainability goes beyond technical interoperability: it requires cultural and institutional recognition of maintenance, curation, and reproducibility as essential scientific contributions. Establishing shared repositories, open evaluation protocols, and clear data citation practices would not only strengthen reproducibility but also transform Knowledge Graph engineering into a collaborative science of knowledge. In this long-term perspective, Semantic Web technologies evolve from isolated tools into foundational infrastructures for trustworthy AI, where transparency, accountability, and openness are enduring design principles.

Taken together, these perspectives outline a coherent trajectory for the future of Knowledge Graphs. By extending temporal reasoning toward hybrid learning, embedding human expertise into validation processes, and grounding research practices in sustainable infrastructures, we can move toward a new generation of systems that are at once intelligent, interpretable, and durable. Such systems will not only support the reliability of artificial intelligence but also foster a collective understanding of knowledge as an evolving, temporally situated, and socially shared construct.

7.4 . Final Note

In an era where artificial intelligence mediates much of our access to knowledge, restoring trust in automated reasoning has become a collective challenge. The temporal and structural mechanisms explored in this thesis outline a path toward accountable and interpretable systems, demonstrating that grounding reasoning in verifiable Knowledge Graphs can reconcile the power of automation with the rigor of factual reliability. Yet the future of Knowledge Graph research extends beyond data integration, pointing toward intelligent systems that can justify their inferences, adapt through time, and acknowledge that knowledge is both temporally situated and culturally contextual. Recognizing that meaning emerges from diverse linguistic and societal backgrounds is essential to prevent artificial intelligence from imposing a single and static view of truth. Building such systems is therefore not only a technical pursuit but a societal endeavor that combines ethical responsibility, cultural sensitivity, and scientific rigor to ensure that our representations of the world remain as dynamic, transparent, and diverse as the world itself.

Chapter A

Résumé en Français

Cette thèse explore la validation temporelle de faits dans des graphes de connaissances multiples et hétérogènes. Elle s'inscrit dans un contexte marqué par l'évolution rapide des informations, par la difficulté croissante de vérifier des données et par la nécessité de disposer de représentations explicables et interopérables. Les graphes de connaissances temporels répondent à ces enjeux en permettant d'associer chaque fait à un intervalle temporel précis et en offrant un cadre pour raisonner sur sa validité et sa cohérence. La thèse soutient que l'association de méthodes symboliques et d'approches apprenantes constitue une voie prometteuse pour renforcer la fiabilité des systèmes de validation.

Introduction. Pour introduire ce travail, le chapitre 1 de la thèse met d'abord en évidence les limites des représentations classiques de la connaissance, souvent incapables de refléter le caractère évolutif des faits. La dynamique des événements politiques, des parcours individuels ou des phénomènes historiques souligne l'importance d'intégrer une dimension temporelle explicite. Dans ce cadre, les graphes de connaissances temporels offrent un moyen rigoureux de structurer les évolutions tout en conservant une sémantique claire. Cette réflexion s'étend également aux difficultés d'interopérabilité entre graphes hétérogènes, ce qui requiert des mécanismes de liaison robustes et réutilisables. Enfin, la thèse met en avant la nécessité de construire des ressources durables, partageables et conformes aux exigences de réutilisabilité scientifique.

Fondamentaux. Le chapitre 2 présente les fondations théoriques nécessaires au reste du manuscrit. Il décrit la structure des graphes de connaissances organisés autour d'un schéma ontologique et d'un graphe de données, puis expose diverses façons de représenter la temporalité, qu'il s'agisse d'intervalles, de points temporels ou de modèles de qualification tels qu'utilisés dans Wikidata. L'algèbre d'Allen occupe une place centrale dans cette présentation puisqu'elle offre un langage formel pour raisonner qualitativement sur les relations temporelles. L'état de l'art couvre également les approches par plongements, statiques ou temporelles, ainsi que les méthodes de liaison reposant sur les clés ou sur l'apprentissage profond. Ces éléments constituent la base conceptuelle des contributions qui suivent.

Contraintes temporelles. Une première contribution est détaillée dans le chapitre 3. Elle introduit une méthode explicable de validation temporelle fondée sur l'extraction automatique de contraintes temporelles. L'approche identifie les relations temporelles récurrentes qui apparaissent entre les propriétés d'une même entité. Ces relations, qu'elles soient simples ou composées, sont déduites à partir de comparaisons de séquences temporelles construites à partir des intervalles présents dans le graphe. Après extraction, les contraintes sont généralisées à l'échelle des classes grâce à deux indicateurs qui permettent d'absorber le bruit et les exceptions. Elles peuvent ensuite servir à confirmer ou infirmer des faits temporels, soit au travers d'une stratégie symbolique reposant sur une justification explicite, soit via une stratégie hybride capable d'ajuster automatiquement l'importance des contraintes. Les expériences montrent que ces contraintes permettent une validation robuste et interprétable.

Benchmark sur la validation temporelle. Le chapitre 4 présente la seconde contribution, qui consiste en la création d'un banc d'évaluation consacré à la validité temporelle. Contrairement aux jeux de données habituellement centrés sur la prédiction de liens, ce banc vise à évaluer la capacité des modèles à déterminer si un intervalle temporel est correct. Les faits négatifs sont générés par corruption contrôlée des données temporelles, ce qui permet d'obtenir des intervalles plausibles mais incorrects. Le banc intègre également des éléments sémantiques tels que les classes, les propriétés ou les descriptions textuelles afin de permettre une comparaison équitable entre approches symboliques et approches neuronales. Il inclut enfin plusieurs granularités temporelles et différentes densités structurelles pour refléter une diversité de contextes.

Transfert de clés d'alignement. La troisième contribution, présentée dans le chapitre 5, étudie la possibilité de transférer des clés structurelles entre graphes de connaissances. Ces clés, qui correspondent à des ensembles minimaux de propriétés permettant d'identifier de manière unique les entités d'une classe, sont découvertes dans un graphe source puis testées dans un graphe cible en s'appuyant sur des alignements de propriétés. Cette démarche évite de redécouvrir des clés dans chaque graphe et favorise la réutilisation de structures fiables. Les expériences démontrent que ce transfert peut améliorer la qualité de la liaison d'entités tout en diminuant le coût computationnel associé.

Ressource en ligne d'alignement. Le chapitre 6 présente enfin une infrastructure destinée à améliorer l'interopérabilité et la soutenabilité dans le Web de données. Cette ressource nommée KG Nexus rassemble vocabulaires, alignements et clés transférables dans un environnement unique afin de faciliter la réutilisation et la comparaison de graphes. L'infrastructure se fonde sur une harmonisation des vocabulaires, sur des techniques récentes d'alignement automatique et sur des mécanismes d'intégration de graphes et de clés. Elle ouvre la voie à divers usages tels que l'enrichissement de graphes, la vérification de schémas ou l'analyse des évolutions conceptuelles.

Conclusion. Le chapitre 7 clôt la thèse en revenant sur l'importance d'articuler temporalité, interopérabilité et soutenabilité pour construire des systèmes de validation fiables et explicables. Le manuscrit montre que la combinaison de contraintes symboliques et d'approches statistiques constitue une base solide pour le raisonnement temporel. La conclusion propose plusieurs perspectives, notamment le développement d'un raisonnement temporel profond fondé sur l'algèbre d'Allen, l'intégration de l'expertise humaine dans les processus de validation ou encore la mise en place d'infrastructures ouvertes et durables pour une intelligence artificielle fondée sur des connaissances vérifiables. L'objectif ultime est que les graphes de connaissances deviennent une mémoire stable, évolutive et auditable capable de soutenir des systèmes raisonnant dans le temps.

Chapter B

List of Publications

International Conferences

Soulard Thibaut, Saïs Fatiha, and Raad Joe. 2025. **When Facts Expire: Benchmarking Temporal Validity in Knowledge Graphs**, **Under publication** in : **CIKM 2025**.

Soulard Thibaut, Saïs Fatiha, and Raad Joe. 2025. **Explainable Temporal Fact Validation Through Constraints Discovery in Knowledge Graphs.** The Semantic Web. **ESWC 2025**. Lecture Notes in Computer Science, vol 15718. Springer, Cham. https://doi.org/10.1007/978-3-031-94575-5_13

Soulard Thibaut, Saïs Fatiha, Raad Joe, and Quercini Gianluca. 2024. **Efficient Key-based Data Linking Through Key Transfer Between Knowledge Graphs.** In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (**SAC '24**). Association for Computing Machinery, New York, NY, USA, 1642–1649. https://doi.org/10.1145/3605098.3636041

National Conferences

Soulard Thibaut, Raad Joe, and Saïs Fatiha. 2024. **Validation temporelle explicable de faits par la découverte de contraintes temporelles complexes dans les graphes de connaissances**. 35es Journées francophones d'Ingénierie des Connaissances (**IC 2024**) @ Plate-Forme Intelligence Artificielle (PFIA 2024), Jul 2024, La Rochelle, France. pp.62-71. **Short listed Best Paper** https://hal.science/hal-04650739v1

Soulard Thibaut, Saïs Fatiha, Raad Joe, and Quercini Gianluca. 2023. **Étude de transférabilité des clés pour le liage de données entre graphes de connaissances**. 34es Journées francophones d'Ingénierie des Connaissances (**IC 2023**) @ Plate-Forme Intelligence Artificielle (PFIA 2023), Jul 2023, Strasbourg, France. **Best paper** https://hal.science/hal-04152691v1

National Journal

Soulard Thibaut, Saïs Fatiha and Raad Joe. **Étude de transférabilité des clés pour le liage de données entre graphes de connaissances**, in Revue Ouverte en Intelligence Artificielle. **Under Publication**

Bibliography

[Abedjan et al.(2019)] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. 2019. *Data Profiling Challenges*. Springer International Publishing, Cham, 103–110. https://doi.org/10.1007/978-3-031-01865-7_9

[Aïmeur et al.(2023)] Esma Aïmeur, Sabrine Amri, and Gilles Brassard. 2023. Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining* 13, 1 (2023), 30.

[Al-Bakri et al.(2015)] Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset. 2015. Inferring Same-As Facts from Linked Data: An Iterative Import-by-Query Approach. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 9–15. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9508

[Allein et al.(2021)] Liesbeth Allein, Isabelle Augenstein, and Marie-Francine Moens. 2021. Time-aware evidence ranking for fact-checking. *Journal of Web Semantics* 71 (2021), 100663.

[Allen(1983)] James F Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 11 (1983), 832–843.

[Atencia et al.(2014)] Manuel Atencia, Michel Chein, Madalina Croitoru, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe, and Danai Symeonidou. 2014. Defining Key Semantics for the RDF Datasets: Experiments and Evaluations. In *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings*. 65–78. https://doi.org/10.1007/978-3-319-08389-6_7

[Atencia et al.(2021)] Manuel Atencia, Jérôme David, and Jérôme Euzenat. 2021. On the relation between keys and link keys for data interlinking. *Semantic Web* 12, 4 (2021), 547–567.

[Atencia et al.(2020)] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, and Jérémy Vizzini. 2020. Link key candidate extraction with relational concept analysis. *Discret. Appl. Math.* 273 (2020), 2–20. https://doi.org/10.1016/j.dam.2019.02.012

[Beek et al.(2020)] Wouter Beek, Joe Raad, Erman Acar, and Frank van Harmelen. 2020. MetaLink: A Travel Guide to the LOD Cloud. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12123)*, Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter

Haase, and Michael Cochez (Eds.). Springer, 481–496. https://doi.org/10.1007/978-3-030-49461-2_28

[Beek et al.(2018)] Wouter Beek, Joe Raad, Jan Wielemaker, and Frank Van Harmelen. 2018. sameas. cc: The closure of 500m owl: sameas statements. In *European semantic web conference*. Springer, 65–80.

[Beek et al.(2014)] Wouter Beek, Laurens Rietveld, Hamid R. Bazoobandi, Jan Wielemaker, and Stefan Schlobach. 2014. LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. In *The Semantic Web – ISWC 2014*, Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble (Eds.). Springer International Publishing, Cham, 213–228.

[Beretta(2018)] Valentina Beretta. 2018. *Data veracity assessment: enhancing Truth Discovery using a priori knowledge*. Theses. IMT Mines Alès. https://hal.science/tel-01914278

[Berners-Lee and Hendler(2001)] Tim Berners-Lee and James Hendler. 2001. Publishing on the semantic web. *Nature (London)* 410, 6832 (2001), 1023–1024.

[Bhuyan et al.(2024)] Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and TP Singh. 2024. Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications* 36, 21 (2024), 12809–12844.

[Bordes et al.(2013)] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

[Bouquet et al.(2007)] Paolo Bouquet, Heiko Stoermer, Daniel Giacomuzzi, et al. 2007. OKKAM: Enabling a Web of Entities. *I3* 5 (2007), 7.

[Boylan et al.(2024)] Jack Boylan, Shashank Mangla, Dominic Thorn, Demian Gholipour Ghalandari, Parsa Ghaffari, and Chris Hokamp. 2024. KGValidator: A Framework for Automatic Validation of Knowledge Graph Construction. *arXiv preprint arXiv:2404.15923* (2024).

[Brown et al.(2020)] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[Cai et al.(2022)] Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2022. Temporal knowledge graph completion: A survey. *arXiv preprint arXiv:2201.08236* (2022).

[Cai et al.(2021)] Ling Cai, Krzysztof Janowicz, Bo Yan, Rui Zhu, and Gengchen Mai. 2021. Time in a Box: Advancing Knowledge Graph Completion with Temporal Scopes. In *Proceedings of the 11th Knowledge Capture Conference* (Virtual Event, USA) *(K-CAP '21)*. Association for Computing Machinery, New York, NY, USA, 121–128. https://doi.org/10.1145/3460210.3493566

[Cai et al.(2024)] Li Cai, Xin Mao, Yuhao Zhou, Zhaoguang Long, Changxu Wu, and Man Lan. 2024. A Survey on Temporal Knowledge Graph: Representation Learning and Applications. https://doi.org/10.48550/arXiv.2403.04782 arXiv:2403.04782 [cs]

[Cer et al.(2018)] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. (2018).

[Chen et al.(2023a)] Jianhao Chen, Junyang Ren, Wentao Ding, and Yuzhong Qu. 2023a. PaTeCon: a pattern-based temporal constraint mining method for conflict detection on knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4166–4172.

[Chen et al.(2017)] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. arXiv:1611.03954 [cs.AI] https://arxiv.org/abs/1611.03954

[Chen et al.(2020)] Riquan Chen, Tianshui Chen, Xiaolu Hui, Hefeng Wu, Guanbin Li, and Liang Lin. 2020. Knowledge Graph Transfer Network for Few-Shot Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 07 (Apr. 2020), 10575–10582. https://doi.org/10.1609/aaai.v34i07.6630

[Chen et al.(2023b)] Zhongwu Chen, Chengjin Xu, Fenglong Su, Zhen Huang, and Yong Dou. 2023b. Meta-learning based knowledge extrapolation for temporal knowledge graph. In *Proceedings of the ACM web conference 2023*. 2433–2443.

[Christophides et al.(2020)] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2020. An Overview of End-to-End Entity Resolution for Big Data. *ACM Comput. Surv.* 53, 6, Article 127 (dec 2020), 42 pages. https://doi.org/10.1145/3418896

[Ciampaglia et al.(2015)] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one* 10, 6 (2015), e0128193.

[Consonni et al.(2019)] Cristian Consonni, Paolo Sottovia, Alberto Montresor, and Yannis Velegrakis. 2019. Discovering Order Dependencies through Order Compatibility. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, Melanie Herschel, Helena Galhardas, Berthold Reinwald, Irini Fundulaki, Carsten Binnig, and Zoi Kaoudi (Eds.). OpenProceedings.org, 409–420. https://doi.org/10.5441/002/EDBT.2019.36

[Cotovio et al.(2024)] Pedro Giesteira Cotovio, Lucas Ferraz, Daniel Faria, Laura Balbi, Marta Contreiras Silva, and Catia Pesquita. 2024. Matcha-dl a tool for supervised ontology alignment. *Semantic Web Journal* (2024).

[Dai et al.(2020)] Yuanfei Dai, Shiping Wang, Neal N Xiong, and Wenzhong Guo. 2020. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics* 9, 5 (2020), 750.

[d'Amato et al.(2021)] Claudia d'Amato, Nicola Flavio Quatraro, and Nicola Fanizzi. 2021. Injecting Background Knowledge into Embedding Models for Predictive Tasks on Knowledge Graphs. In *The Semantic Web*, Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Oscar Corcho, Petar Ristoski, and Mehwish Alam (Eds.). Springer International Publishing, Cham, 441–457.

[Dasgupta et al.(2018)] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, 2018-10), Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 2001–2011. https://doi.org/10.18653/v1/D18-1225

[de Groot et al.(2021)] Thomas de Groot, Joe Raad, and Stefan Schlobach. 2021. Analysing large inconsistent knowledge graphs using anti-patterns. In *European Semantic Web Conference*. Springer, 40–56.

[Devlin et al.(2018)] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[Dumontier et al.(2017)] Michel Dumontier, Pierre-Yves Vandenbussche, Ghislain A. Atemezing, María Poveda-Villalón, and Bernard Vatant. 2017. Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semant. Web* 8, 3 (jan 2017), 437–452. https://doi.org/10.3233/SW-160213

[Fan et al.(2020)] Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating Fact Checking Briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He,

and Yang Liu (Eds.). Association for Computational Linguistics, Online, 7147–7161. https://doi.org/10.18653/v1/2020.emnlp-main.580

[Fanourakis et al.(2022)] Nikolaos Fanourakis, Vasilis Efthymiou, Dimitris Kotzinos, and Vassilis Christophides. 2022. Knowledge Graph Embedding Methods for Entity Alignment: An Experimental Review. arXiv:2203.09280 [cs.DB]

[Fernández et al.(2011)] Javier D. Fernández, Miguel A. Martínez-Prieto, Mario Arias, Claudio Gutierrez, Sandra Álvarez-García, and Nieves R. Brisaboa. 2011. Lightweighting the Web of Data through Compact RDF/HDT. In *Advances in Artificial Intelligence*, Jose A. Lozano, José A. Gámez, and José A. Moreno (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 483–493.

[Fernández et al.(2007)] Norberto Fernández, José M Blázquez, Luis Sánchez, and Ansgar Bernardi. 2007. IdentityRank: Named entity disambiguation in the context of the NEWS project. In *The Semantic Web: Research and Applications: 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007. Proceedings 4*. Springer, 640–654.

[Fiorelli and Stellato(2020)] Manuel Fiorelli and Armando Stellato. 2020. A Lime-Flavored REST API for Alignment Services. In *Proceedings of the 7th Workshop on Linked Data in Linguistics (LDL-2020)* (Marseille, France, 2020-05), Maxim Ionov, John P. McCrae, Christian Chiarcos, Thierry Declerck, Julia Bosque-Gil, and Jorge Gracia (Eds.). European Language Resources Association, 52–60. https://aclanthology.org/2020.ldl-1.8/

[García-Durán et al.(2018)] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202* (2018).

[García-Durán et al.(2018)] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, 2018). Association for Computational Linguistics, 4816–4821. https://doi.org/10.18653/v1/D18-1516

[Gastinger et al.(2024)] Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. 2024. TGB 2.0: A Benchmark for Learning on Temporal Knowledge Graphs and Heterogeneous Graphs. 37 (2024), 140199–140229. https://proceedings.neurips.cc/paper_files/paper/2024/hash/fda026cf2423a01fcbcf1e1e43ee9a50-Abstract-Datasets_and_Benchmarks_Track.html

[Geng et al.(2023)] Yuxia Geng, Jiaoyan Chen, Jeff Z Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. 2023. Relational message passing for fully inductive knowledge graph completion. In *2023 IEEE 39th international conference on data engineering (ICDE)*. IEEE, 1221–1233.

[Gerber et al.(2015)] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. 2015. DeFacto—Temporal and multilingual Deep Fact Validation. 35 (2015), 85–101. https://doi.org/10.1016/j.websem.2015.08.001

[Hobbs and Pan(2004)] Jerry R Hobbs and Feng Pan. 2004. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)* 3, 1 (2004), 66–85.

[Hogan et al.(2021)] 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. 2021. *Knowledge Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01125ED1V01Y202109DSK022

[Hogan et al.(2012)] Aidan Hogan, Antoine Zimmermann, Jürgen Umbrich, Axel Polleres, and Stefan Decker. 2012. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Journal of Web Semantics* 10 (2012), 76–110.

[Islakoglu et al.(2024)] Duygu Sezen Islakoglu, Melisachew Wudage Chekol, and Yannis Velegrakis. 2024. Leveraging Pre-trained Language Models for Time Interval Prediction in Text-Enhanced Temporal Knowledge Graphs. In *European Semantic Web Conference*. Springer, 59–78.

[Jain et al.(2020)] Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. 2020. Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 3733–3747. https://doi.org/10.18653/v1/2020.emnlp-main.305

[Ji et al.(2015)] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. 687–696.

[Jiao et al.(2023)] Songlin Jiao, Zhenfang Zhu, Wenqing Wu, Zicheng Zuo, Jiangtao Qi, Wenling Wang, Guangyuan Zhang, and Peiyu Liu. 2023. An improving reasoning network for complex question answering over temporal knowledge graphs.

Applied Intelligence 53, 7 (01 Apr 2023), 8195–8208. https://doi.org/10.1007/s10489-022-03913-6

[Jiménez-Ruiz(2024)] Ernesto Jiménez-Ruiz. 2024. LogMap Family Participation in the OAEI 2024. In *Proceedings of the 19th International Workshop on Ontology Matching co-located with the 23rd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 11, 2024 (CEUR Workshop Proceedings, Vol. 3897)*, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Cássia Trojahn, Sven Hertling, Huanyu Li, Pavel Shvaiko, and Jérôme Euzenat (Eds.). CEUR-WS.org, 161–162. https://ceur-ws.org/Vol-3897/oaei2024_paper10.pdf

[Jiménez-Ruiz and Cuenca Grau(2011)] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. 2011. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web – ISWC 2011* (2011). Springer, Berlin, Heidelberg, 273–288. https://doi.org/10.1007/978-3-642-25073-6_18 ISSN: 1611-3349.

[Jin et al.(2020)] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, 2020-11), Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 6669–6683. https://doi.org/10.18653/v1/2020.emnlp-main.541

[Jonquet et al.(2018)] Clément Jonquet, Anne Toulet, Elizabeth Arnaud, Sophie Aubin, Esther Dzalé Yeumo, Vincent Emonet, John Graybeal, Marie-Angélique Laporte, Mark A. Musen, Valeria Pesce, and Pierre Larmande. 2018. AgroPortal: A vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture* 144 (2018), 126–143. https://doi.org/10.1016/j.compag.2017.10.012

[Kim et al.(2023)] Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. FactKG: Fact Verification via Reasoning on Knowledge Graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 16190–16206. https://doi.org/10.18653/v1/2023.acl-long.895

[Kontokostas et al.(2014)] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. 2014. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd international conference on World Wide Web*. 747–758.

[Lacroix et al.(2020)] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor Decompositions for Temporal Knowledge Base Completion. https://iclr.cc/virtual_2020/poster_rke2P1BFwS.html

[Larmande and Todorov(2021)] Pierre Larmande and Konstantin Todorov. 2021. AgroLD: A Knowledge Graph for the Plant Sciences. In *The Semantic Web – ISWC 2021*, Andreas Hotho, Eva Blomqvist, Stefan Dietze, Achille Fokoue, Ying Ding, Payam Barnaghi, Armin Haller, Mauro Dragoni, and Harith Alani (Eds.). Springer International Publishing, Cham, 496–510.

[Leblay and Chekol([n. d.])] Julien Leblay and Melisachew Wudage Chekol. [n. d.]. Deriving Validity Time in Knowledge Graph. In *Companion Proceedings of the The Web Conference 2018* (Republic and Canton of Geneva, CHE, 2018-04-23) *(WWW '18)*. International World Wide Web Conferences Steering Committee, 1771–1776. https://doi.org/10.1145/3184558.3191639

[Leetaru and Schrodt(2013)] Kalev Leetaru and Philip A Schrodt. 2013. GDELT: Global Data on Events, Location and Tone,. (2013).

[Ligozat(1991)] Gérard Ligozat. 1991. On generalized interval calculi. In *Proceedings of the ninth National conference on Artificial intelligence-Volume 1*. 234–240.

[Lin et al.(2015)] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[Ling et al.(2021)] C Ling, Krzysztof Janowicz, Z Rui, and G Mai. 2021. Time in a Box: Advancing Knowledge Graph Completion with Temporal Scopes. In *K-CAP'21: Proceedings of the 11th on Knowledge Capture Conference*.

[Liu and Duan(2021)] Jiangzhou Liu and Li Duan. 2021. A Survey on Knowledge Graph-Based Recommender Systems. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2450–2453. https://doi.org/10.1109/IAEAC50856.2021.9390863

[Liu et al.(2022)] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 4120–4127.

[Ma et al.(2025)] Jiatong Ma, Linmei Hu, Rang Li, and Wenbo Fu. 2025. LoCal: Logical and Causal Fact-Checking with LLM-Based Multi-Agents. In *Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 1614–1625. https://doi.org/10.1145/3696410.3714748

[Mahdisoltani et al.(2013)] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

[Mao et al.(2020)] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational Reflection Entity Alignment. In *Proceedings of the 29th ACM International*

Conference on Information & Knowledge Management (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 1095–1104. https://doi.org/10.1145/3340531.3412001

[Marchisio et al.(2023)] Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2023. Mini-Model Adaptation: Efficiently Extending Pretrained Models to New Languages via Aligned Shallow Training. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5474–5490. https://doi.org/10.18653/v1/2023.findings-acl.338

[Mudgal et al.(2018)] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. 19–34. https://doi.org/10.1145/3183713.3196926

[Mühleisen and Bizer(2012)] Hannes Mühleisen and Christian Bizer. 2012. Web Data Commons-Extracting Structured Data from Two Large Web Corpora. *LDOW* 937, 133-145 (2012), 65.

[Nassiri et al.(2020)] Armita Khajeh Nassiri, Nathalie Pernelle, Fatiha Saïs, and Gianluca Quercini. 2020. RE-miner for data linking results for OAEI 2020. In *Proceedings 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 (CEUR Workshop Proceedings, Vol. 2788)*, Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, and Cássia Trojahn (Eds.). CEUR-WS.org, 211–215. https://ceur-ws.org/Vol-2788/oaei20_paper12.pdf

[Nayyeri et al.(2022)] Mojtaba Nayyeri, Sahar Vahdati, Md Tansen Khan, Mirza Mohtashim Alam, Lisa Wenige, Andreas Behrend, and Jens Lehmann. 2022. Dihedron Algebraic Embeddings for Spatio-Temporal Knowledge Graph Completion. In *The Semantic Web* (2022). Springer, Cham, 253–269. https://doi.org/10.1007/978-3-031-06981-9_15 ISSN: 1611-3349.

[Ngomo and Auer(2011)] Axel-Cyrille Ngonga Ngomo and Sören Auer. 2011. LIMES—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[Nickel et al.(2016)] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[Nickel et al.(2011)] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data.. In *Icml*, Vol. 11. 3104482–3104584.

[Noy et al.(2017)] Natasha Noy, Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. 2017. A Survey of Current Link Discovery Frameworks. *Semant. Web* 8, 3 (jan 2017), 419–436. https://doi.org/10.3233/SW-150210

[Noy et al.(2009)] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, Christopher G Chute, et al. 2009. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research* 37, suppl_2 (2009), W170–W173.

[Paulheim(2016)] Heiko Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2016), 489–508.

[Peng et al.(2023)] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge Graphs: Opportunities and Challenges. 56, 11 (2023), 13071–13102. https://doi.org/10.1007/s10462-023-10465-9 Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 11 Publisher: Springer Netherlands.

[Pernelle et al.(2013)] Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. 2013. An automatic key discovery approach for data linking. *Journal of Web Semantics* 23 (2013), 16–30.

[Portisch et al.(2024)] Jan Portisch, Michael Hladik, and Heiko Paulheim. 2024. Background knowledge in ontology matching: A survey. *Semantic Web* 15, 6 (2024), 2639–2693. https://doi.org/10.3233/SW-223085 arXiv:https://journals.sagepub.com/doi/pdf/10.3233/SW-223085

[Qudus et al.(2023)] Umair Qudus, Michael Röder, Sabrina Kirrane, and Axel-Cyrille Ngonga Ngomo. 2023. TemporalFC: A Temporal Fact Checking Approach over Knowledge Graphs. In *The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6–10, 2023, Proceedings, Part I* (Berlin, Heidelberg, 2023-11-06). Springer-Verlag, 465–483. https://doi.org/10.1007/978-3-031-47240-4_25

[Raad et al.(2018)] Joe Raad, Wouter Beek, Frank van Harmelen, Nathalie Pernelle, and Fatiha Saïs. 2018. Detecting Erroneous Identity Links on the Web Using Network Metrics. In *The Semantic Web – ISWC 2018*, Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl (Eds.). Springer International Publishing, Cham, 391–407.

[Rezatofighi et al.(2019)] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized Intersection

Over Union: A Metric and a Loss for Bounding Box Regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 658–666. https://doi.org/10.1109/CVPR.2019.00075

[Saxena et al.(2020)] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 4498–4507. https://doi.org/10.18653/v1/2020.acl-main.412

[Saïs et al.(2009)] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. 2009. Combining a logical and a numerical method for data reconciliation. In *Journal on Data Semantics XII*. Springer, 66–94.

[Schlichtkrull et al.(2018a)] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018a. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 593–607.

[Schlichtkrull et al.(2018b)] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018b. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 593–607.

[Shiralkar et al.(2017)] Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding streams in knowledge graphs to support fact checking. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 859–864.

[Sismanis et al.(2006)] Yannis Sismanis, Paul Brown, Peter J Haas, and Berthold Reinwald. 2006. GORDIAN: efficient and scalable discovery of composite keys. In *Proceedings of the 32nd international conference on Very large data bases*. 691–702.

[Song et al.(2023)] Yiqing Song, Wenfa Li, Guiren Dai, and Xinna Shang. 2023. Advancements in Complex Knowledge Graph Question Answering: A Survey. *Electronics* 12, 21 (2023). https://doi.org/10.3390/electronics12214395

[Soru et al.(2015)] Tommaso Soru, Edgard Marx, and Axel-Cyrille Ngonga Ngomo. 2015. ROCKER: A Refinement Operator for Key Discovery. In *Proceedings of the 24th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2015-05-18) *(WWW '15)*. International World Wide Web Conferences Steering Committee, 1025–1033. https://doi.org/10.1145/2736277.2741642

[Soulard(2022)] T. Soulard. 2022. *Knowledge-based Entity Linking in Heterogeneous Knowledge Graphs at Web-Scale*. Technical Report. Université Paris Saclay. https://hal.science/hal-04046394

[Suchanek et al.(2024)] Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. 2024. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2024-07-11) *(SIGIR '24)*. Association for Computing Machinery, 131–140. https://doi.org/10.1145/3626772.3657876

[Sun et al.(2019)] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[Sun et al.(2018)] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (Stockholm, Sweden) *(IJCAI'18)*. AAAI Press, 4396–4402.

[Sun et al.(2020)] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment* 13, 12 (aug 2020), 2326–2340. https://doi.org/10.14778/3407790.3407828

[Syed et al.(2019)] Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2019. Unsupervised discovery of corroborative paths for fact validation. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*. Springer, 630–646.

[Symeonidou et al.(2014)] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. 2014. SAKey: Scalable Almost Key Discovery in RDF Data. In *The Semantic Web – ISWC 2014*, Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble (Eds.). Vol. 8796. Springer International Publishing, 33–49. https://doi.org/10.1007/978-3-319-11964-9_3 Series Title: Lecture Notes in Computer Science.

[Symeonidou et al.(2017)] Danai Symeonidou, Luis Galárraga, Nathalie Pernelle, Fatiha Saïs, and Fabian Suchanek. 2017. Vickey: Mining conditional keys on knowledge bases. In *Lecture Notes in Computer Science* (Vienne, Austria, 2017-10) *(Lecture Notes in Computer Science, Vol. 10587)*. Springer, 661–677. https://hal.science/hal-01647597

[Tappolet and Bernstein(2009)] Jonas Tappolet and Abraham Bernstein. 2009. Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *The Semantic Web: Research and Applications*, Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 308–322.

[Tchechmedjiev et al.(2019)] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A knowledge graph of fact-checked claims. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, 309–324.

[Thanapalasingam et al.(2022)] Thiviyan Thanapalasingam, Lucas van Berkel, Peter Bloem, and Paul Groth. 2022. Relational graph convolutional networks: a closer look. *PeerJ Computer Science* 8 (2022), e1073.

[Thiéblin et al.(2020)] Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. 2020. Survey on complex ontology matching. *Semantic Web* 11, 4 (2020), 689–727.

[Trouillon et al.(2016)] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[Vrandečić and Krötzsch(2014)] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57, 10 (Sept. 2014), 78–85. https://doi.org/10.1145/2629489

[Wang et al.(2023)] Jiapu Wang, Boyue Wang, Meikang Qiu, Shirui Pan, Bo Xiong, Heng Liu, Linhao Luo, Tengfei Liu, Yongli Hu, Baocai Yin, and Wen Gao. 2023. A Survey on Temporal Knowledge Graph Completion: Taxonomy, Progress, and Prospects. https://doi.org/10.48550/arXiv.2308.02457 arXiv:2308.02457 [cs]

[Wang et al.(2021)] Shuai Wang, Joe Raad, Peter Bloem, and Frank Van Harmelen. 2021. Refining transitive and pseudo-transitive relations at web scale. In *European Semantic Web Conference*. Springer, 249–264.

[Wang et al.(2014)] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.

[Wilkinson et al.(2016)] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9.

[Wu et al.(2024)] Di Wu, Hsien-Tseng Wang, and Abdullah Uz Tansel. 2024. A survey for managing temporal data in RDF. *Information Systems* 122 (2024), 102368. https://doi.org/10.1016/j.is.2024.102368

[Xiao et al.(2022)] Renjie Xiao, Zijing Tan, Haojin Wang, and Shuai Ma. 2022. Fast Approximate Denial Constraint Discovery. *Proc. VLDB Endow.* 16, 2 (oct 2022), 269–281. https://doi.org/10.14778/3565816.3565828

[Yang et al.(2015)] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6575

[Yang et al.(2014)] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[Zhang et al.(2024)] Yuchao Zhang, Xiangjie Kong, Zhehui Shen, Jianxin Li, Qiuhua Yi, Guojiang Shen, and Bo Dong. 2024. A survey on temporal knowledge graph embedding: Models and applications. *Knowledge-Based Systems* 304 (2024), 112454. https://doi.org/10.1016/j.knosys.2024.112454