Hasso Plattner Institute
University of Potsdam
Information Systems Group

Representation and Curation of Knowledge Graphs with Embeddings

Dissertation submitted for the degree of
"Doktor der Ingenieurwissenschaften"
(Dr.-Ing.)
in the Scientific Discipline
Information Systems

Digital Engineering Faculty
Hasso Plattner Institute, University of Potsdam

By: Nitisha Jain

Potsdam, October 2022

Reviewers

Prof. Dr. Felix Naumann
Hasso Plattner Institute for Digital Engineering, University of Potsdam

Prof. Dr. Steffen Staab
Institute for Parallel and Distributed Systems, University of Stuttgart

Prof. Dr. Katja Hose
Department of Computer Science, Aalborg University

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-61224
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-612240

Abstract

Knowledge graphs are structured repositories of knowledge that store facts about the general world or a particular domain in terms of entities and their relationships. Owing to the heterogeneity of use cases that are served by them, there arises a need for the automated construction of domain-specific knowledge graphs from texts. While there have been many research efforts towards open information extraction for automated knowledge graph construction, these techniques do not perform well in domain-specific settings. Furthermore, regardless of whether they are constructed automatically from specific texts or based on real-world facts that are constantly evolving, all knowledge graphs inherently suffer from incompleteness as well as errors in the information they hold.

This thesis investigates the challenges encountered during knowledge graph construction and proposes techniques for their curation (a.k.a. refinement) including the correction of semantic ambiguities and the completion of missing facts. Firstly, we leverage existing approaches for the automatic construction of a knowledge graph in the art domain with open information extraction techniques and analyse their limitations. In particular, we focus on the challenging task of named entity recognition for artwork titles and show empirical evidence of performance improvement with our proposed solution for the generation of annotated training data.

Towards the curation of existing knowledge graphs, we identify the issue of polysemous relations that represent different semantics based on the context. Having concrete semantics for relations is important for downstream applications (e.g. question answering) that are supported by knowledge graphs. Therefore, we define the novel task of finding fine-grained relation semantics in knowledge graphs and propose *FineGReS*, a data-driven technique that discovers potential sub-relations with fine-grained meaning from existing polysemous relations. We leverage knowledge representation learning methods that generate low-dimensional vectors (or embeddings) for knowledge graphs to capture their semantics and structure. The efficacy and utility of the proposed technique are demonstrated by comparing it with several baselines on the entity classification use case.

Further, we explore the semantic representations in knowledge graph embedding models. In the past decade, these models have shown state-of-the-art results for the task of link prediction in the context of knowledge graph completion. In view of the popularity and widespread application of the embedding techniques not only for link prediction but also for different semantic tasks, this thesis presents a critical analysis of the embeddings by quantitatively

measuring their semantic capabilities. We investigate and discuss the reasons for the shortcomings of embeddings in terms of the characteristics of the underlying knowledge graph datasets and the training techniques used by popular models.

Following up on this, we propose *ReasonKGE*, a novel method for generating semantically enriched knowledge graph embeddings by taking into account the semantics of the facts that are encapsulated by an ontology accompanying the knowledge graph. With a targeted, reasoning-based method for generating negative samples during the training of the models, *ReasonKGE* is able to not only enhance the link prediction performance, but also reduce the number of semantically inconsistent predictions made by the resultant embeddings, thus improving the quality of knowledge graphs.

Zusammenfassung

Wissensgraphen sind strukturierte Wissenssammlungen, die Fakten über die allgemeine Welt oder eine bestimmte Domäne in Form von Entitäten und deren Beziehungen speichern. Aufgrund der Heterogenität der Anwendungsfälle, für die sie verwendet werden, besteht ein Bedarf an der automatischen Erstellung von domänenspezifischen Wissensgraphen aus Texten. Obwohl es viele Forschungsbemühungen in Richtung offener Informationsextraktion für die automatische Konstruktion von Wissensgraphen gegeben hat, sind diese Techniken in domänenspezifischen Umgebungen nicht sehr leistungsfähig. Darüber hinaus leiden alle Wissensgraphen, unabhängig davon, ob sie automatisch aus spezifischen Texten oder auf der Grundlage realer Fakten, die sich ständig weiterentwickeln, konstruiert werden, unter Unvollständigkeit und Fehlern in den darin enthaltenen Informationen.

Diese Arbeit untersucht die Herausforderungen, die bei der Konstruktion von Wissensgraphen auftreten, und schlägt Techniken zu ihrer Kuratierung (auch bekannt als Verfeinerung) vor, einschließlich der Korrektur semantischer Mehrdeutigkeiten und der Vervollständigung fehlender Fakten. Zunächst nutzen wir bestehende Ansätze für die automatische Erstellung eines Wissensgraphen im Kunstbereich mit offenen Informationsextraktionstechniken und analysieren deren Grenzen. Insbesondere konzentrieren wir uns auf die anspruchsvolle Aufgabe der Named Entity Recognition für Kunstwerke und zeigen empirische Belege für eine Leistungsverbesserung mit der von uns vorgeschlagenen Lösung für die Generierung von annotierten Trainingsdaten.

Im Hinblick auf die Kuratierung bestehender Wissensgraphen identifizieren wir das Problem polysemer Relationen, die je nach Kontext unterschiedliche Semantiken repräsentieren. Konkrete Semantiken für Relationen sind wichtig für nachgelagerte Anwendungen (z.B. Fragenbeantwortung), die durch Wissensgraphen unterstützt werden. Daher definieren wir die neuartige Aufgabe, feinkörnige Relationssemantiken in Wissensgraphen zu finden und schlagen *FineGReS* vor, eine datengesteuerte Technik, die eine datengesteuerte Technik, die potenzielle Unterbeziehungen mit feinkörniger Bedeutung aus bestehenden polysemen Beziehungen entdeckt. Wir nutzen Lernmethoden zur Wissensrepräsentation, die niedrigdimensionale Vektoren (oder Einbettungen) für Wissensgraphen erzeugen, um deren Semantik und Struktur zu erfassen. Die Wirksamkeit und Nützlichkeit der vorgeschlagenen Technik wird durch den Vergleich mit verschiedenen Basisverfahren im Anwendungsfall der Entitätsklassifizierung demonstriert.

Darüber hinaus untersuchen wir die semantischen Repräsentationen in Modellen zur Einbettung von Wissensgraphen. In den letzten zehn Jahren haben diese Modelle in den letzten zehn Jahren die besten Ergebnisse bei der

Vorhersage von Links im Zusammenhang mit der Vervollständigung von Wissensgraphen erzielt. Angesichts der Popularität und der weit verbreiteten Anwendung der Einbettungstechniken nicht nur für die Linkvorhersage, sondern auch für andere semantische Aufgaben, wird in dieser Arbeit eine kritische Analyse der Einbettungen durch quantitative Messung ihrer semantischen Fähigkeiten vorgenommen. Wir untersuchen und diskutieren die Gründe für die Unzulänglichkeiten von Einbettungen in Bezug auf die Eigenschaften der zugrundeliegenden Wissensgraphen-Datensätze und die von den populären Modellen verwendeten Trainingstechniken.

Darauf aufbauend schlagen wir *ReasonKGE* vor, eine neuartige Methode zur Erzeugung semantisch angereicherter Wissensgrapheneinbettungen durch Berücksichtigung der Semantik der Fakten, die durch eine den Wissensgraphen begleitende Ontologie gekapselt sind. Mit einer gezielten, schlussfolgernden Methode zur Erzeugung von Negativproben während des Trainings der Modelle ist *ReasonKGE* in der Lage, nicht nur die Leistung der Link-Vorhersage zu verbessern, sondern auch die Anzahl der semantisch inkonsistenten Vorhersagen der resultierenden Einbettungen zu reduzieren und damit die Qualität der Wissensgraphen zu verbessern.

Acknowledgements

I would like to express my deepest appreciation to Felix Naumann and Ralf Krestel, who have provided me with their guidance and support throughout the Ph.D. I have learned a lot from our discussions, not only in terms of academic research but also in paying attention to the details. I am also thankful to Fabian Suchanek, who has been a great mentor to me since before I began my Ph.D. officially. I am grateful that I could turn to him at times when I needed a fresh perspective or expert opinion on technical ideas and my questions were always met with a kind and understanding ear.

Though Ph.D. is indeed meant to be a challenging experience, for me, the last four years have proven to be especially and surprisingly arduous. Like many of us, the pandemic was a life-changing and unique experience for me, one that brought new realizations and unexpected problems, but at the same time, an opportunity to truly reflect on my core values and priorities in life. In a way, I am grateful to the challenging times for pushing me on the path to self-discovery and making me more resilient in the face of adversities. I believe that I am better equipped now to make smart choices and steer my career in the direction that is conducive to my goals. I am thankful for the rewarding moments that accompanied the hardships in this journey - moments of success and achievement, and recognition from the community, that have given me great confidence in my skills as an independent researcher.

I consider myself fortunate to have worked with many talented students and researchers throughout my Ph.D., both at HPI as well as external. The collaboration with Jan-Christoph Kalo has been perhaps the most successful one in my research career. I hope our partnership will continue in the future and bring us both further success. My colleagues at the chair contributed to a friendly and supportive work environment in the office. I especially enjoyed working with my bright students at HPI - Jan Ehmueller, Philipp Schmidt, Maria Lomaeva and Lucas Silbernagel and appreciate their contributions to my research and personal growth as an advisor. My dear friend Ting, with whom I have shared this Ph.D. experience, has been a constant source of companionship and support. Unbeknownst to her, our long conversations on work and life have always given me a deep sense of trust and comfort. I will certainly miss our regular meetings and wish her the best in life and career.

Finally, and most importantly, I want to express my deepest gratitude to my family and close friends, who truly acted as my pillar of strength and enabled me to reach this far. None of this would have been even fathomable without the unwavering support from my loving mother to pursue my dreams and ambitions all through these years. She has always been the beacon of light

for me in the dark times of extreme hardships when I was close to giving up. I am eternally thankful to my amazing husband Aashish who has been constantly by my side (both literally and virtually) through the long years of Ph.D., being my biggest cheerleader and always the proudest whenever I found success. Not only have you been my moral touchstone, but I am lucky to have you as my sounding board for technical ideas as well. I could not be more impatient to start afresh our lives together and have long meaningful conversations for the rest of our time.

Dedicated to
To my Mom, Dad, little sister and my partner Aashish
whose love and support are always with me ..

Publications List

This thesis is based on 5* out of 12 publications that have resulted from my Ph.D.

- *__Nitisha Jain__, Ralf Krestel: Discovering Fine-Grained Semantics in Knowledge Graph Relations. Proceedings of the Thirty-First ACM International Conference on Information and Knowledge Management (CIKM), 2022.

- *__Nitisha Jain__, Alejandro Sierra-Múnera, Jan Ehmueller, Ralf Krestel: Generation of Training Data for Named Entity Recognition of Artworks. Semantic Web Journal (Special Issue Cultural Heritage 2021).

- Maria Lomaeva, __Nitisha Jain__: Relation Canonicalization in Open Knowledge Graphs: A Quantitative Analysis. Proceedings of the Extended Semantic Web Conference, Posters and Demos (ESWC), 2022.

- *__Nitisha Jain__, Alejandro Sierra-Múnera, Philipp Schmidt, Julius Streit, Simon Thormeyer, Maria Lomaeva, Ralf Krestel: Generating Domain-Specific Knowledge Graphs: Challenges with Open Information Extraction. Proceedings of the International Workshop on Knowledge Graph Generation from Text at the Extended Semantic Web Conference (Text2KG@ESWC), 2022.

- *__Nitisha Jain__, Trung-Kien Tran, Mohamed H. Gad-Elrab, Daria Stepanova: Improving Knowledge Graph Embeddings with Ontological Reasoning. Proceedings of the International Semantic Web Conference (ISWC), 2021.

- *__Nitisha Jain__, Jan-Christoph Kalo, Wolf-Tilo Balke, Ralf Krestel: Do Embeddings Actually Capture Knowledge Graph Semantics?. Proceedings of the Extended Semantic Web Conference (ESWC), 2021.

- __Nitisha Jain__, Christian Bartz, Tobias Bredow, Emanuel Metzenthin, Jona Otholt, Ralf Krestel: Semantic Analysis of Cultural Heritage Data: Aligning Paintings and Descriptions in Art-Historic Collections. Proceedings of the International Workshop on Fine Art Pattern Extraction and Recognition at the International Conference on Pattern Recognition (FAPER@ICPR), 2020.

- __Nitisha Jain__, Ralf Krestel: Learning Fine-Grained Semantics for Multi-Relational Data. Proceedings of the International Semantic Web Conference, Posters and Demos (ISWC), 2020.

- __Nitisha Jain__: Domain-Specific Knowledge Graph Construction for Semantic Analysis. Proceedings of the Extended Semantic Web Conference, Doctoral Symposium (ESWC), 2020.

- **Nitisha Jain**, Christian Bartz, Ralf Krestel: Automatic Matching of Paintings and Descriptions in Art-Historic Archives using Multimodal Analysis. Proceedings of the International Workshop on Artificial Intelligence for Historical Image Enrichment and Access at the Language Resources and Evaluation Conference (AI4HI@LREC), 2020.

- Simon Razniewski, **Nitisha Jain**, Paramita Mirza, Gerhard Weikum: Coverage of Information Extraction from Sentences and Paragraphs. Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

- **Nitisha Jain**, Ralf Krestel: Who is Mona L.? Identifying Mentions of Artworks in Historical Archives. Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL), 2019.

Contents

Chapter 1

Knowledge Graphs

"The greatest enemy of knowledge is not ignorance,
it is the illusion of knowledge."
— Stephen Hawking

The advent of Big Data has led to a tremendous increase in the rate of generation of data over the past decade. While this data in and of itself is useful, the real value lies in the extraction of important nuggets of information from this data to generate and accumulate *knowledge*. This knowledge when stored in a structured and easily accessible form can power a wide variety of applications. This is where *Knowledge Graphs* serve an essential role. Knowledge graphs (KGs) are a popular form of representation of facts that are extracted from texts contained in Web pages and unstructured or semi-structured documents.

KGs have become an integral part of the Semantic Web [12], where the stored information is represented with the help of the RDF standard [96]. Each node in the knowledge graph corresponds to an entity of the semantic web, which is connected to the other nodes by means of meaningful relations that are the edges of the graph. A fact is, thus, represented in RDF language as a triple ⟨subject, predicate, object⟩ where subject and object belong to a set of entities and predicate refers to the relationship between them. Knowledge graphs rely on an underlying schema or *ontology* consisting of the concepts (that define the *type* of the entities) and the possible relationships among the concepts. The ontology holds the key to the semantic meaning of the facts in a KG and dictates the logical rules as well as restrictions for populating the KG with actual data. They play a central role in the Semantic Web by providing a set of common rules and standards for shared communication of data and knowledge [105]. Thus, ontologies encapsulate the necessary data semantics that can enable machine understanding of real-world data.

After the introduction and popularization of the first web-scale knowledge graph[1] by Google in 2012 [41], KGs have been powering not only Google's search, but also an increasing number of applications in the area of natural language processing (NLP) and information extraction (IE) [25, 41]. Several KGs have been made publicly available for academic research and applications, including Wikidata [160], DBpedia [97], Yago [145] and NELL [26]. Yago was created from crawling the Wikipedia pages for facts and using the Wikipedia categories to build the entity types. DBpedia was also built from Wikipedia

[1] prior to this, databases of facts were known as knowledge bases in the Semantic Web community

with a focus on the info-boxes to derive the DBpedia ontology. NELL has been constructed with facts from millions of Web pages based on an initial list of predicates. These KBs consist of millions of entities and relationships between them, e.g., the birthplace of a person, the actor or director of a movie, the country that a city is located in and so on. Such KBs are used for question answering, Web search, recommendation systems, personal assistants, and a variety of other AI applications [168].

Apart from the general purpose KGs that have been primarily designed and curated through academic efforts, major technology and financial companies such as Goldman Sachs, Bosch, Microsoft and Amazon have also created their own KGs that power their respective use cases, both internally and externally. In addition, KGs are also being explored for applications in specific domains such as medical, financial, cultural heritage and so on. As such, domain-specific knowledge graphs have been gathering a lot of attention for various applications (as discussed in detail in Chapter 2).

1.1 Construction of Knowledge Graphs

While they are undeniably valuable information repositories, the process of construction of knowledge graphs from textual sources is a long-standing research problem. The complexity and nuances of natural language makes the extraction of information an arduous task. Since language allows for a rich variety of expressions for conveying knowledge, automatic extraction of facts is a complicated endeavour, with several steps of information extraction, including named entity recognition (NER), entity linking, relation extraction (RE), as well as the refinement of the extracted facts for a clean and structured representation in form of a KG. While some of the steps such as NER have been steadily getting better and show state-of-the-art results comparable to manual efforts, others such as RE are more difficult and far from an established solution that is adoptable for practical use cases.

Furthermore, due to the large volumes of data that have been made available by the Web, the manual creation of a general-purpose KG is nearly an impossibility. As such, automated methods for KG construction based on machine learning techniques have been widely employed [26, 142]. These techniques populate a KG based on an ontology which is assumed to be available beforehand. The availability of a suitable ontology is a strong assumption, one that puts a constraint on the extraction of all information available in the text and might even restrict the quality and coverage of the resultant KG. This effect is especially more pronounced for domain-specific settings where the expertise to design custom ontologies is often scarce as well as quite expensive. Therefore, Open Information Extraction (Open IE) techniques, that do not rely on a pre-existing schema for fact extraction, prove to be more feasible for the construction of KGs in many scenarios.

Open information extraction. Open IE has emerged as a popular approach, where a large set of relational triples can be extracted from text without any human input or domain expertise [46] in the absence of a pre-specified list of relations. These techniques focus on the extraction of triples consisting of noun phrases and relation phrases from the text as a first step. Thereafter, these triples are *canonicalized* to ensure a consistent representation of entities and relations and to remove duplicate information. Several Open IE techniques have been proposed to build and populate knowledge graphs from

free-form texts [8, 32, 47, 56, 91, 177]. Traditional systems were either rule-based or statistical and relied heavily on pattern-based extractions. Recently, Open IE techniques based on neural networks have also been proposed [188]. While these techniques are fundamentally targeted for domain-independent extractions from heterogeneous corpora, their evaluation is predominantly performed by using Web and news datasets. Hence, their scalability and efficacy for unseen and novel datasets are not considered or properly evaluated. Research on this topic is still ongoing and an effective Open IE pipeline for the automated construction of a KG from a specific corpus is yet elusive. In the context of this open question, there is a need to investigate the challenges of employing Open IE techniques for the construction of a domain-specific knowledge graph and identify the key areas of improvement.

1.2 Knowledge Graph Curation

Modern KGs store information about millions of facts, however, whether they have been populated on the basis of an ontology or constructed automatically from texts, the resulting KGs are often bound to be incomplete in terms of fact coverage and they are rarely fully correct [54]. Due to the inaccuracies induced by different statistical or linguistic methods employed for their construction, several types of quality issues can manifest in the KGs, not only in terms of incorrect facts, but also in the form of semantic ambiguities or inconsistencies. To address these shortcomings, knowledge graph *curation* or *refinement* is an essential task that ensures the quality of the knowledge graphs before they can be deployed for downstream applications.

Knowledge graph *completion* has been the subject of particular interest in many research efforts. Popular KGs, that represent facts about the real world, are inherently incomplete due to the evolving nature of the information. Furthermore, for KGs constructed from the Web, automatic extraction techniques fail to extract all information, and the underlying sources can be incomplete themselves. In the case of KG construction with Open IE techniques, it is hard to establish a recall of 100% for the extraction of facts from raw text. Even if all the facts were being correctly extracted by Open IE, it is unreasonable to expect all the information would be explicitly stated in the text such that it can be extracted. Consider the text '*US President Barack Obama's wife Michelle Obama*' - from this, we could derive triples such as ⟨*Obama, presidentOf, US*⟩ and ⟨*Barack Obama, hasSpouse, Michelle Obama*⟩, but it still does not give us the fact that *Michelle Obama* is the *first lady* of the *US*. Such facts can only be derived through inference over the KG facts in the presence of similar information for other entities.

The most common techniques for this type of inference used to be statistical methods [127, 151]. However, representation learning techniques have recently become quite popular for KG completion. After the advent of word embeddings as a powerful means of representation learning for words, similar embeddings were proposed for KG representation as well. Just like word embeddings create dense, real valued vectors for the words in low dimensions (as compared to TF-IDF where the dimensions are as high as the number of words), *Knowledge Graph Embeddings* mainly learn the representation of the input KG by projecting entities and relations in a low-dimensional vector space, such that these vectors capture some key structural relationships between the entities and relations on a global level (entire KG). As such, knowledge graph embedding models

have been employed for KG completion by performing *fact prediction or link prediction*. This is the task of predicting facts that are true in the real world, but missing in the KB. Due to their conceptual simplicity and high scalability, knowledge graph embeddings have become one of the most popular strategies not only for KG completion, but also for other semantic tasks such as entity clustering [51] and entity typing [82]. In the following section, we outline the basic idea for KG embeddings and the way they are trained. We also present an overview of the most popular types of embedding models with examples.

1.3 Representation Learning with Embeddings

In the last decade, numerous methods for computing knowledge graph embeddings have been proposed. The methods differ from one another in terms of how they relate the entities and relations of the KG in the latent space. We explain the most basic embedding model TransE [20], which is a *translation-based model*. The embeddings are designed so that for a triple $\langle h, r, t \rangle$, the vectors $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ satisfy the relation $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ or $\mathbf{r} \approx \mathbf{t} - \mathbf{h}$ (as denoted in Figure 1.1). For example, if we know $\langle Barack, marriedTo, Michelle \rangle$ then the model would create the vector **Barack** + **marriedTo** to be close to the vector **Michelle**. An embedding with these properties has several advantages: firstly, the embedding allows us to feed entities and relations into machine learning methods that work on vectors (e.g., classification algorithms). The vectors are typically low in dimension (e.g., a few hundred), which makes them particularly suited for such applications. Secondly, the embedding provides a natural way of grouping together similar entities, such that the vectors for similar entities would lie close to one another in the vector space. In our example, we would expect *Barack* to be close in the vector space to other politicians. Most importantly, as mentioned previously, the embeddings allow for the prediction of missing links, e.g. if the entity for the spouse of *Barack* was missing, then the vector obtained from **Barack** + **marriedTo** would correspond to the vector of the correct entity i.e. *Michelle*.

In terms of implementation, knowledge graph embeddings are created by trainable machine-learning models, where they take as input a fact $\langle h,\ r,\ t \rangle$, and output a score of its likelihood of being true - the higher the score, the more likely the model believes the fact to be true. This score is typically denoted by $f(\langle h, r, t \rangle)$ or $f_{\vec{r}}(\vec{h}, \vec{t})$. To train such a model, it needs to be provided with a set of true facts from the KG, as well as negative samples associated with the facts so as to avoid over-generalization. The negative samples are typically generated by corrupting the facts from the KB, i.e., by taking a fact $\langle h, r, t \rangle$ from the KB and replacing the tail by a random entity t' (further negative sampling techniques will be discussed in Chapter 5).

TransE belongs to the class of *geometric models* which interpret relations as geometric operations in the vector space. One limitation of TransE is the inability to model symmetric relationships [166]. TransE also has problems modeling many-to-one, reflexive, and transitive relations, and to capture multiple semantics of a relation. Subsequently proposed models such as TransH [166], TransD [83], TransR [101] tried to obviate some of these issues. For instance, TransH [166] tries to alleviate the limitations of TransE by allowing an entity to have different representations in the embedding space depending on the relation it is involved with. Each relation r is represented not only by a vector $\mathbf{r}$, but also by an hyperplane (i.e. a sub-space of one dimension less than the embedding space).

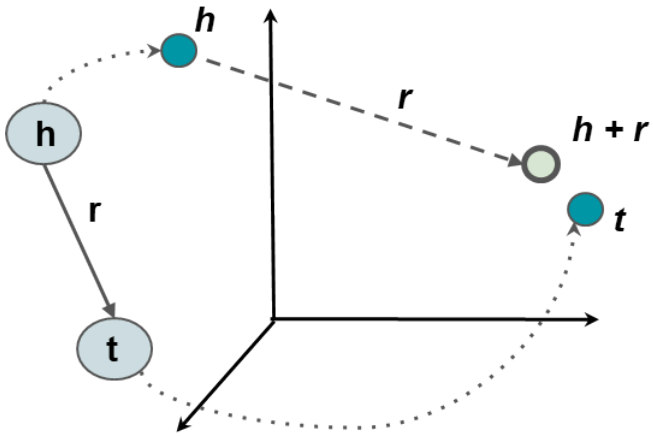


Figure 1.1: The relation between the vectors for a triple $\langle h, r, t \rangle$ in the embedding space.

Algebraically a hyperplane can be defined by a single vector, namely the vector that is orthogonal to it. Thus, each relation r is associated with a set of two vectors: r for the relation itself, and h_r for its hyperplane.

Other embedding methods in the *semantic matching* category compare the vector of the subject and the vector of the object directly in order to assess how likely the fact is to be true. In these models, the latent semantics of the entities and relations are matched to determine the plausibility score of a triple. These models perceive the link prediction task as a tensor decomposition task where the KG is a tensor(3-D adjacency matrix) that can be decomposed into entity and relation embeddings as low-dimensional vectors.

RESCAL [119] is the simplest model in this category. Entities are represented as vectors and relations become bilinear functions (simply represented as square matrices). A triple $\langle h, r, t \rangle$ is then scored by the application of the relation-specific bilinear function to the entity embeddings: $f(\langle h, r, t \rangle) = \vec{h}^t \cdot M_r \cdot \vec{t}$, where $\vec{h}$ (resp. $\vec{t}$) is the embedding of h (resp. t) and $M_r \in \mathbb{R}^{d \times d}$ is the representing matrix of r.

DistMult [176] is a variation of the RESCAL model where the relation matrices are all forced to be diagonal. This simplifies the computations, and reduces the parameter space. As a drawback, DistMult gives the same score for the triples $\langle h, r, t \rangle$ and $\langle t, r, h \rangle$. Thus, it is unable to model asymmetric relations such as *sonOf*, *actedIn* etc. Despite these limitations, DistMult has been recently shown to perform as well as many recently proposed models, presumably due to its simplicity and scalability [140]. ComplEx [153] improves upon the DistMult model by using the same diagonal constraint, but with complex-valued embedding vectors.

Deep neural architectures have also been introduced for KB embeddings, with the hope that hidden layers can capture more complex interaction patterns between entities and relations (and then estimate more complex scoring functions). ConvE [38] is a popular example of models that are based on convolutional neural networks (CNN). These can learn complex nonlinear features of the entities and relations with fewer parameters by using 2D convolutions over embeddings. ConvE has been shown to be particularly effective for complex graphs with nodes having a high number of incoming edges. The model introduced the 1-N scoring scheme where for a given triple $\langle h, r, t \rangle$ where t is to be predicted, the matching is performed with all the tail entities at the same time, leading

to speedier training. ConvE has proven to be a competitive embedding model and a popular baseline for more recent deep learning approaches.

The embedding models outlined in this section are primarily evaluated in terms of their performance for the link prediction task related to the KG completion. But embeddings also implicitly capture the semantics of the KG, and thus can be employed for tasks such as entity similarity. In this thesis, we take advantage of these embeddings for performing a novel task of KG curation in terms of refining the semantics of the relations in the KG. At the same time, it is important to question the limitation of the semantic representations in embeddings instead of assuming they would perform well for all semantic use cases. For this, a critical and quantitative analysis of the semantic capability of embeddings is imperative and thus, a part of this thesis. Furthermore, ontologies serve as the semantic guide of knowledge graphs which can also improve the semantics of embeddings when they are included in the training process of the models. The role of ontologies during the training as well as the performance benefits from this approach are also an important contribution of this thesis.

1.4 Outline and Contributions

This chapter has provided a short introduction to knowledge graphs as well as research efforts related to their construction and refinement. We have also discussed popular models for knowledge graph embeddings and their role in KG representation and completion. The remainder of the thesis is organized into individual chapters that address a specific research question in the above context and describe our contributions towards a solution.

Chapter 2 is focused on the construction of knowledge graphs, in particular, the challenges of constructing a domain-specific KG. We address the following :

(RQ 1) How can a knowledge graph be constructed automatically from domain-specific texts with the help of existing techniques for Open IE ? In particular, how can we improve named entity recognition for domain-specific entities?

The chapter first describes our efforts to extract triples from a noisy domain-specific corpus with the help of Open IE pipeline. The challenges encountered at each step of the KG construction process are discussed at length. The features and statistics of the resulting KG are also presented in this chapter. The contributions of this work were the result of a collaboration with Alejandro Sierra-Munera as well as our students Philipp Schmidt, Julius Streit, Simon Thormeyer and Maria Lomaeva. This work was published at the Text2KG workshop at the ESWC 2022 conference [80].

Furthermore, this chapter also presents our contributions towards generating training data for the named entity recognition task, particularly for domain-specific entities. We motivate the importance and challenges of identifying artwork titles from a corpus of art-historic data as well as establish the lack of annotated training data as the main reason for sub-par performance of existing NER tools. Subsequently, we discuss our proposed pipeline for automatically generating such annotated data and demonstrate the improvement in NER results. Alejandro Sierra-Munera and Jan Ehmueller have provided valuable contributions to this effort in terms of additional experimental results and an NER demo tool. This work was published as a research paper in the Semantic Web Journal in the Special Issue on Cultural Heritage 2021 [79].

Chapter 3 is devoted to the curation of knowledge graphs. In this context, we specifi-

cally address the following:
(RQ 2) How can existing knowledge graphs be refined in terms of having unambiguous semantics for relations with the help of embedding techniques for knowledge graph representation?
This chapter first discusses the issue of relation polysemy in popular knowledge graphs and the advantages of fine-grained relation semantics for downstream applications. We then describe our proposed method *FineGReS* which leverages knowledge graph embeddings to derive sub-relations having precise semantics in a data-driven and scalable manner. To demonstrate the effectiveness and utility of this method, an extensive experimental analysis is also included in this chapter. The chapter is based on the full paper at the research track of CIKM 2022 conference [78].

Chapter 4 is concerned with the exploration of the semantic representation in knowledge graph embedding models. We address the following:
(RQ 3) Do popular knowledge graph embedding models adequately capture the semantics of knowledge graph components in a way that is universally applicable for semantic tasks?
This chapter discusses the general adoption of embedding models for various semantic tasks with examples of related work and presents our investigation of the semantic representation in the embeddings. The results of the experimental analysis reveal serious shortcomings for the fine-grained semantics of entities. Further, the reasons behind the shortcomings are also explored and detailed in the chapter. The full paper published at ESWC 2021 forms the basis of this chapter. This work was done in close collaboration with Jan-Christoph Kalo (PhD student at TU Braunschweig at the time) who contributed to the design as well as the execution of the experiments.

In Chapter 5, we endeavour to address the shortcomings of KG embedding models for the prediction of missing links as presented in the previous chapter. Specifically, we look at the following question:
(RQ 4) How can we improve the semantics and overall performance of knowledge graph embeddings by integrating ontological reasoning during the training of the models?
We present our technique for negative sampling *ReasonKGE* that can detect semantically incorrect predictions being made by an embedding model via ontological reasoning and generate targeted negative samples for the next iteration of training to prevent such mistakes. The *ReasonKGE* approach is able to considerably not only improve link prediction performance but also improve the ratio of semantically consistent predictions for any underlying embedding model. The results of further experiments are also included in this chapter. This work was done during my Ph.D. sabbatical with the Bosch Centre for AI (Renningen) under the guidance of Daria Stepanova and Trung-Kien Tran with inputs from Gad-Elrab and it was published as a research paper in ISWC 2021.

Finally, Chapter 6 concludes this thesis by providing a summary of the contributions made by this thesis. This chapter also discusses the ideas for extending the proposed contributions for furthering the research in the area of knowledge graphs and their semantic representation with embeddings.

Chapter 2

Domain-Specific Knowledge Graph Construction

"If you wish to make an apple pie from scratch,
you must first invent the universe."
— Carl Sagan

As outlined in Chapter 1, automated KG construction is an ongoing research area. While there have been several efforts in this direction, there is no widely accepted and effective pipeline for the automated construction of a KG from a given corpus. The problem gets significantly compounded in domain-specific scenarios where general-purpose techniques for automated KG construction suffer serious limitations [86]. This chapter is devoted to highlighting the challenges of constructing a domain-specific KG in an automated manner with Open IE techniques. Domain-specific NER in the absence of annotation data is one of the major challenges and we show that an approach for generating large, good quality annotated datasets for NER models can be adapted for identification of domain-specific entities.

Chiefly, this chapter is based on the contributions in two publications - first is Jain et al. [80] that presents the details of an Open IE pipeline for domain-specific KG construction and the limitations thereof. The second is Jain et al. [79] that concerns with named entity recognition for domain-specific entities, specifically artworks. This chapter is structured as follows — in Section 2.1 we discuss the need for domain-specific knowledge graphs as well as the particular difficulties faced during their construction. Section 2.2 presents related work in the area of domain-specific tasks, relating to Open IE based techniques for KG construction and named entity recognition for domain-specific entities. In Section 2.3 we describe the construction of an art-historic KG with the help of Open IE techniques and the challenges that were encountered at each step. Section 2.4 provides the details of our contributions towards the NER for artworks, where we propose an automated technique for the generation of training data for NER. Finally, Section 2.5 gives a summary of this chapter.

2.1 Domain-Specific Knowledge Graphs

General purpose knowledge graphs constructed from Web sources cover a wide range of domains. As such, they cannot be expected to be comprehensive and semantically aligned to any single domain in particular. In order for knowledge graphs to be useful for a specific domain, it is essential to have a semantically-rich and comprehensive representation of the domain in the KG. This is where domain-specific KGs play an important role. For instance, the most important concepts and relations differ from one domain to the other — for the financial domain, concepts such as Bank, Loans, etc. are important to detect and classify, while for the biomedical domain, the names of Proteins, Genes etc. are important to be correctly identified. Due to this, general purpose techniques need to be adapted for the semantic representation of specific domains.

In order to motivate and explore the research problems for the construction of domain-specific KGs, we consider cultural heritage as a representative domain. We are working in collaboration with the Wildenstein Plattner Institute[1] that was founded to promote scholarly research on cultural heritage collections, where a wealth of information is buried in large collections of recently digitized art resources. In these resources, cultural objects such as artworks, auctions, art collections, artistic movements etc. are often mentioned within semi-structured or unstructured text narratives. The identification and extraction of the mentions of these cultural objects as named entities and establishing their relations can facilitate a plethora of applications such as search and browsing in digital resources, help art historians to track the provenance of artworks and enable wider semantic text exploration for digital cultural resources.

However, extraction of relevant entities as well as construction of a representative *art* knowledge graph is a non-trivial task. This is attributed to the inherent complexities with cultural heritage data as well the lack of any domain-specific gold standard annotated datasets for training and evaluation of automated techniques. Consider the task of named entity recognition (NER) - most of the recent neural network based NER models have been trained on a few well-established corpora available for the task such as the CoNNL datasets [149, 150] or OntoNotes [131]. Although these systems attain state-of-the-art results for the generic NER task, their performance and utility for identifying fine-grained entities is essentially limited due to the specific training of the models. Thus, it comes as no surprise that it has been a challenge to adapt NER systems for identifying fine-grained and domain-specific named entities with reasonable accuracy [130, 132].

Cultural heritage data poses several additional challenges - the data is extremely heterogeneous and comprises of multiple topics, multiple languages as well as numerous different text formats ranging from structured tabular data to long passages of unstructured text descriptions. Data obtained from historical archives also poses significant linguistic challenges in terms of outdated vocabularies and phrases, such that the modern natural language processing tools are unable to perform well for these texts [43]. In the absence of gold standard annotation datasets for NER as well as other natural language processing and information retrieval tasks, the adaptation of existing solutions to the art and cultural heritage domain faces significant challenges [156]. We elaborate on this further in the following sections in the context of constructing an art-historic KG in an automated manner, including the task of NER for cultural heritage entities.

[1] https://wpi.art/

2.2 Related Work

Here, we discuss the previous work related to construction of knowledge graphs in domain-specific settings, especially in the cultural heritage domain. An overview of efforts for automated KG construction with Open IE techniques is included as well. We also present a discussion of previous work on domain-specific and fine-grained NER as well as efforts related to the cultural heritage domain. Further, we mention previous work in the context of generation of annotated datasets for NER.

Knowledge graphs for cultural heritage. The construction of domain-specific KGs has been the subject of investigation in previous works for various domains, e.g. software engineering [185], academic literatures [72], and more prominently, the biomedical domain [11, 45, 178]. With the availability of digitized cultural heritage data, previous works have proposed KGs for art-related datasets [27, 73, 124, 171]. Arco [27] is a large Italian cultural heritage graph with a pre-defined ontology that was developed in a collaborative fashion with contributions from domain experts all over the country. While the Arco KG is quite broad in its coverage, Ardo [161] pertains to a very specific use case of multimedia archival records. Similarly, the Linked Stage Graph [148] was developed as a KG specifically for storing historical data about the Stuttgart State Theater. Increasingly, the principles of linked open data[2] have also been widely adopted within the cultural heritage domain for facilitating researchers, practitioners and generic users to study and consume cultural objects. Notable examples include the CIDOC-CRM [123], the Rijksmuseum collection [39], the Zeri Photo Archive[3], OpenGLAM [157] among many others. Most related to our work is the ArtGraph [28] where the authors have integrated the art resources from DBpedia and WikiArt and constructed a KG with a well-defined schema that is centered around artworks and artists. While all these works are concerned with KGs and ontologies for specific art-related corpora, they have leveraged a schema for representing the information and are not concerned with the challenges of an extraction process in its absence, which is the main focus of our work.

Open IE for KG construction. Open IE approaches for the construction of KGs extract triples directly from text, without an explicit ontology or schema behind the extraction process. Several works have been proposed in the past. TextRunner [177] relies on a self supervised classifier which determines trustworthy relationships with pairs of entities, while Reverb [47] uses syntactical and lexical constraints to overcome incoherent and uninformative relationships. ClausIE [32] relies heavily on dependency parsing to construct clauses from which the propositions will be extracted. The Stanford CoreNLP OpenIE implementation [8, 108] leveraged in this work uses dependency parsing to minimize the phrases of the resulting clauses, and was originally evaluated in a slot filling task. However, these methods suffer from a number of shortcomings in terms of their applicability to specific domains [86]. Existing techniques that exhibit state-of-the-art results on standard, clean datasets fail to achieve comparable performance for domain-specific datasets [75]. Moreover, none of the the previously proposed automated methods are directly applicable for the arts and cultural heritage domain, where unique challenges with respect to the heterogeneity and quality of data are prevalent 2.1 . In

[2]Linked Open Data: http://www.w3.org/DesignIssues/LinkedData
[3]https://fondazionezeri.unibo.it/en

this chapter, we identify and discuss the particular difficulties encountered while applying existing information extraction techniques to art-related corpora 2.3.2.

NER for cultural heritage. Named Entity Recognition, being an important step towards KG construction, has been the subject of numerous research efforts [99]. There is also prior work for domain-specific NER, such as for the biomedical domain. NER systems have been used to identify the names of drugs, proteins and genes [88, 93, 155]. But since these techniques rely on specific resources such as carefully curated lists for drug names [90] or biology and microbiology NER datasets [36, 68], they are highly specific solutions geared towards biomedical domain and cannot be applied directly to cultural heritage data. In the absence of gold standard NER annotation datasets, the adaptation of existing solutions to the art and cultural heritage domain faces many challenges, some of them being unique to this domain. Seth et al. [156] discuss some of these difficulties and compare the performance of several NER tools on descriptions of objects from the Smithsonian Cooper-Hewitt National Design Museum in New York. Segers et al. [141] also offer an interesting evaluation of the extraction of event types, actors, locations, and dates from unstructured text present in the management database of the Rijksmuseum in Amsterdam. However, their test data contains Wikipedia articles which are well-structured and more suitable for extraction of named entities. On similar lines, Rodriquez et al. [138] discuss the performance of several available NER services on a corpus of mid-20th-century typewritten documents and compare their performance against manually annotated test data having named entities of types people, locations, and organizations. Ehrmann et al. [43] offer a diachronic evaluation of various NER tools for digitized archives of Swiss newspapers. Freire et al. [50] use a CRF-based model to identify persons, locations and organizations on cultural heritage structured data. However, none of the existing works have focused on the task of identifying titles of paintings and sculptures which are one of the most important named entities for the art domain. Moreover, previous works have merely compared the performance of existing NER systems for cultural heritage, whereas our contribution aims to improve the performance of NER systems by generating domain-specific high-quality training data.

Generation of NER training data. While several prominent systems have achieved near human performance for the few most common entity types [7, 100, 107, 187], they are dependent on a few prevalent benchmark datasets that provide gold standard annotations for training purposes. These benchmark datasets were manually annotated using proper guidelines and domain expertise. E.g., the CoNNL and OntoNotes datasets, that were created on news-wire articles, are widely shared among the research community. Since these NER systems are trained on a corpus of news articles they perform well only for comparable datasets. Also, these datasets include a predefined set of named entity categories, which might not correspond in different entity domains. In most cases, these systems fail to adapt well to new domains and different named entity categories [130, 132]. Manual curation of gold standard annotations for large domain-specific corpus is expensive in terms of human labour and cost, while also requiring significant domain expertise. Hence our work complements the efforts of NER model improvements by focusing on the automated generation of training datasets for these models. In Varma et al. [158], the authors attempt to aid the creation of labeled training data in weakly-supervised fashion by a heuristic based approach. There are other works that depend on heuristic patterns along

with user input [23, 67]. In the context of generating training datasets for NER, previous works have exploited the linked structure of Wikipedia to identify and tag the entities with their type, thus creating annotations via distance supervision [5, 121]. Ghaddar and Langlais further extended this work by adding more annotations from Wikipedia [58] and adding fine-grained types for the entities [57]. However, these techniques are only useful in a very limited way for the cultural heritage domain, since Wikipedia texts do not contain sufficient entity types relevant to this domain. We propose a framework to generate a high-quality training corpus in a scalable and automated manner and demonstrate that NER models can be trained to identify mentions of artworks with notable performance gains.

In the next section, we outline our efforts for building an art-historic KG from a cultural heritage dataset with an Open IE pipeline. The features of the resulting KG and the limitations of generic techniques at each step will be discussed in detail as well.

2.3 Knowledge Graph Construction for Cultural Heritage

The art and cultural heritage domain provides a plethora of opportunities for knowledge graph applications. An art knowledge graph can enable art historians, as well as interested users, to explore interesting information that is hidden in large volumes of text in a structured manner. With a large variety of diverse information sources and manifold application scenarios, the (automated) construction of task-specific and domain-specific knowledge graphs becomes even more crucial for this domain.

In contrast to general purpose KGs, a KG for the art domain could comprise a specific set of entity types, such as *artworks*, *galleries*, as well as relevant relations, such as *influenced_by*, *part_of_movement* etc., depending on the specific task and on the specific text collection. The important entities and relations might also differ across different document types, such as auction catalogues, exhibition catalogues, or art magazines. On one hand, a general purpose, art-oriented ontology may not be well-suited and comprehensive enough for specific data collections. On the other hand, designing a custom ontology for the different art corpora would be a challenging and expensive task due to the need for significant domain expertise. In the past, several attempts have been made at creating KGs for art and related domains [27, 73, 171], with the most recent one by Castellano et al. [28]. However, there exists no systematic method for the construction of a knowledge graph based on a collection of art-related documents without a well-defined ontology. The schema-less Open IE approach is attractive for the art collections since there is no need to rely on existing ontologies to dictate the information extraction process which might restrict the scope of the entities and relations that could be extracted from the text (when the ontology is not hand-crafted for the specific dataset).

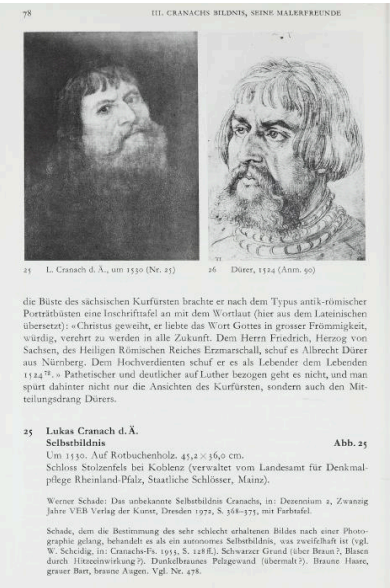
This section presents the results from our exploration of existing Open IE techniques to generate structured information. We discuss our insights in terms of their shortcomings and limited applicability when deployed for noisy, digitized data in the art domain.

2.3.1 Dataset

A large collection of digitized art historical documents was made available by our project partners as a representative cultural heritage dataset. The dataset consists of art related

Table 2.1: Types of documents in WPI dataset

Document type	Count	Ratio
Auction Catalogues	71,192	0.45
Books	42,370	0.27
Exhibition Catalogues	38,176	0.24
Others	7,054	0.04


78 III. CRANACHS BILDNIS, SEINE MALERFREUNDE

25 L. Cranach d. Ä., um 1530 (Nr. 25) 26 Dürer, 1524 (Anm. 90)

die Büste des sächsischen Kurfürsten brachte er nach dem Typus antik-römischer Porträtbüsten eine Inschrifttafel an mit dem Wortlaut (hier aus dem Lateinischen übersetzt): »Christus geweiht, er liebte das Wort Gottes in grosser Frömmigkeit, würdig, verehrt zu werden in alle Zukunft. Dem Herrn Friedrich, Herzog von Sachsen, des Heiligen Römischen Reiches Erzmarschall, schuf es Albrecht Dürer aus Nürnberg. Dem Hochverdienten schuf er es als Lebender dem Lebenden 1524[78]. « Pathetischer und deutlicher auf Luther bezogen geht es nicht, und man spürt dahinter nicht nur die Ansichten des Kurfürsten, sondern auch den Mitteilungsdrang Dürers.

25 **Lukas Cranach d. Ä.**
Selbstbildnis **Abb. 25**
Um 1550. Auf Rotbuchenholz. 45,2 × 36,0 cm.
Schloss Stolzenfels bei Koblenz (verwaltet vom Landesamt für Denkmalpflege Rheinland-Pfalz, Staatliche Schlösser, Mainz).

Werner Schade: Das unbekannte Selbstbildnis Cranachs, in: Dezennium 2, Zwanzig Jahre VEB Verlag der Kunst, Dresden 1972, S. 368–375, mit Farbtafel.

Schade, dem die Bestimmung des sehr schlecht erhaltenen Bildes nach einer Photographie gelang, behandelt es als ein autonomes Selbstbildnis, was zweifelhaft ist (vgl. W. Schmidt, in: Cranachs-Fs. 1953, S. 128ff.). Schwarzer Grund (über Braun?, Blasen durch Hitzeeinwirkung?). Dunkelbraunes Pelzgewand (übermalt?). Braune Haare, grauer Bart, braune Augen. Vgl. Nr. 478.

Figure 2.1: Example of scanned page

texts in many different languages including English, French, German, Italian, Dutch, Spanish, Swedish and Danish among others. The collection consists of different types of documents: auction catalogues, full texts of art books related to particular artists or art genres, catalogues of art exhibitions and other documents. The auction and exhibition catalogues contain semi-structured and unstructured texts that describe artworks on display, mainly paintings and sculptures. Art books may contain more unstructured text about the origins of artworks and their creators. Table 2.1 shows the proportion of the different kinds of documents in the dataset. For reference, a few sample documents from a similar collection of digitized exhibition catalogues[4] and historical art journals[5] are shown in Fig. 2.1. The pages of the catalogues and books in the WPI dataset were scanned with OCR and each page was converted to an entry stored within an elastic search index. Due to the limitations of OCR, the dataset did not retain its rich original formatting information which would have been very useful for analysis. In fact, the data suffers from many spelling and formatting mistakes that need to be appropriately handled. Fig. 2.2 shows a typical text excerpt that highlights the noise in the dataset. After OCR of the page, the page numbers are merged with the text, any formatting indicators present in the original page are lost, there are several spelling errors and it is hard to distinguish

[4]from - Lukas Cranach: Gemälde, Zeichnungen, Druckgraphik ; Ausstellung im Kunstmuseum Basel 15. Juni bis 8. September 1974, (https://digi.ub.uni-heidelberg.de/diglit/koepplin1974bd1/0084,0095)

[5]from - Studio: international art-2.1894, October 1984, (https://digi.ub.uni-heidelberg.de/diglit/studio1894/0019)

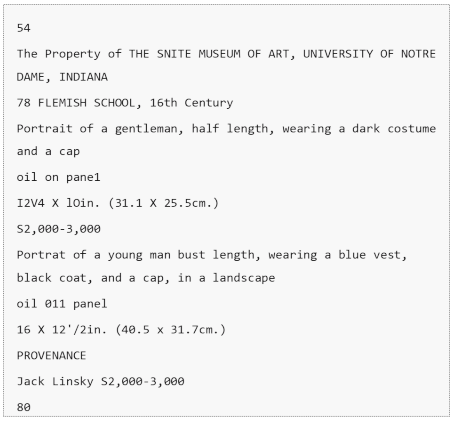


Figure 2.2: Example of digitized text

the artwork title from its description.

2.3.2 Open IE for KG construction

In this section, we describe the steps employed for the automated extraction of information (in form of triples) to construct an art-historic knowledge graph based on our underlying art-historic dataset. Fig. 2.3 shows an overview of this process. In order to restrict the size of the dataset for a proof-of-concept of our KG construction process, a subset of the entire dataset pertaining to information about the artist *Picasso* was chosen. The decision of choosing an artist-oriented subset of the collection enabled us to better understand the context and evaluate the triples that were obtained throughout the process of KG construction. The data was filtered by querying the document collection using the keyword query *'Picasso'*, resulting in 224,469 entries (where each entry corresponds to a page of the original digitized corpus) containing the term *'Picasso'*. Due to the filtering, each entry is an independent document, in the sense that the neighboring entries do not always represent the correct context. This led to some of the entries in our dataset containing incomplete sentences at the beginning or the end of a page. One such example is an entry starting with *'to say47—Picasso never belittled his work, until . . . '* where the tokens *'to say'* belong to a sentence which started in a different entry, that might no longer be a part of the dataset under consideration. It is important to note that in the same example we can see more noise from the OCR process, e.g., numbers are mixed in between words in the digitized version of the text. In general, the dataset contains full sentences, such as *'Matisse's return to the study of ancient and Renaissance sculpture is significant in itself.'*, as well as short description phrases, figure captions or footnotes such as *'G. Bloch, Pablo Picasso, Bern, 1972, vol. III, p.142'*.

Finding named entities

As a first step, it was interesting to inspect if the named entities present in the corpus could be easily identified. A dictionary-based approach to find the named entities would identify the mentions with a high precision, but at the cost of very low recall by ignoring many potentially interesting entities to be discovered in the corpus. Therefore, we chose to follow a machine learning approach to named entity recognition (NER). Generic NER tools work very well for the common entity types, such as person, location, organization and

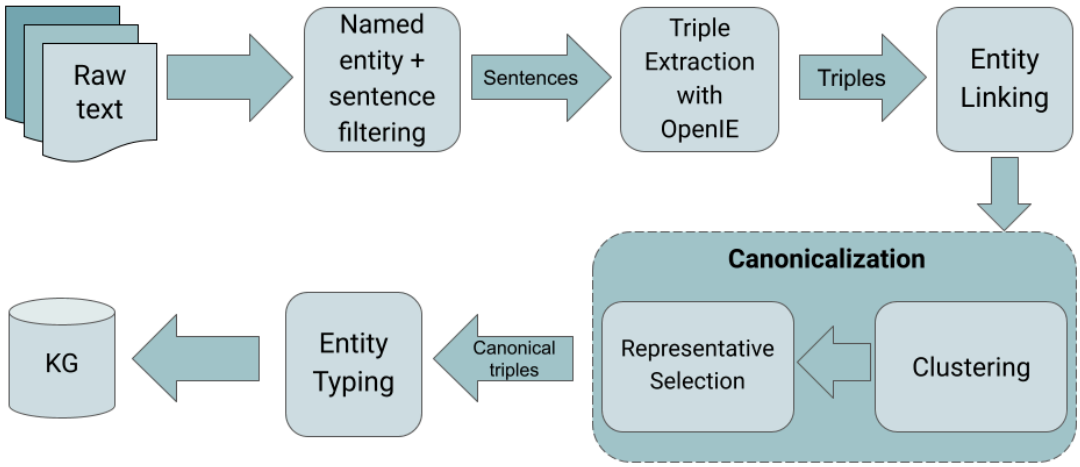


Figure 2.3: Steps for construction of art-historic KG

so on, though fine-grained or domain-specific entities are harder to identify (Section 2.4). We employed the SpaCy library[6] for finding named entities since its pre-trained models includes a *Work_Of_Art category* that could potentially identify the entities that are important in the art domain (this could encompass mentions of paintings, books, statues etc.). Excluding the cardinal entities in order to reduce noise, the SpaCy library with the pre-trained *'en_core_web_trf'* model was used to identify the following entity types - Work_Of_Art, Person, Product, ORG, LOC, GPE and NORP, which showed reasonably good results. The process of NER enabled us to filter out any sentences without any entity mention since such sentences were likely to have no useful information for the KG construction. Thus, the NER step helped with pruning the dataset for further processing, as well as improving the quality of the resulting KG.

Triple extraction

After obtaining informative sentences from the previous step, we employed Open IE tools to extract the triples from them. It is important to note that while there are some art-related ontologies proposed in previous works such as Arco [27] and ArDo [161], none of them are suitable for our corpus since they are very specific to the datasets they were designed for. Other general ontologies such as CIDOC-CRM are, on the other hand, too broad and would not be able to extract novel and interesting facts from a custom and heterogeneous corpus such as ours, where the entities and relations among them are not known before hand. In the absence of such an ontology specifically designed for the description of art-historic catalogs, open information extraction techniques for the construction of our KG enabled us to broaden the scope and utility of the extracted information.

To this end, we ran the Stanford CoreNLP OpenIE annotator [108, 133] to extract

[6] https://spacy.io/usage/v3

⟨subject, predicate, object⟩ triples from the sentences. A total of 5,057,488 triples were extracted in this process, where multiple triples could be extracted from a single sentence. Another round of filtering was performed at this stage, where any triples that did not contain a named entity in the subject or object phrase were removed. Additionally, duplicate entries and triples with serial numbers as entities were also ignored. Some examples of triples that were removed are: ⟨*we, have, good relationship*⟩, ⟨*i, be, director*⟩, ⟨*brothel, be in, evening*⟩, ⟨*drawings, acquired, work*⟩. A total of 160,000 triples remained, a valid triple at this stage looked like ⟨*P. Picasso*, *is*, *artiste*⟩.

Entity linking

Once the triples were extracted, the entity linking component of the Stanford CoreNLP pipeline [108] was used to link the entities. This component uses WikiDict as a resource, and uses the dictionary to match the entity mention text to a specific entity in Wikipedia. Since the entities in our dataset were present in multiple different surface forms, this step allowed us to partially normalize the entities and identify the unique entities. Though the number of entities was reduced as a result, the total number of triples remained the same. Note that this linking could only map entities to their Wikipedia counterpart if the entity was found as a subject or object in a triple. In many cases though, the subject and object were noun phrases instead of obvious entities, for which this kind of linking did not really work. This process was still quite useful as around 108,841 out of 337,100 entities were successfully linked to their Wikipedia form (leading to 8,369 unique entities). Some of the most frequent entities found in the dataset (along with their frequencies) were: (*Pablo_Picasso*, 11219), (*Paris*, 2178), (*Artist*, 1904), (*Henri_Matisse*, 1769), (*Georges_Braque*, 1352).

Canonicalization

One of the main challenges when constructing a KG through Open IE techniques, is that of canonicalization. Multiple surface forms of the same entity or relation might be observed in the triples extracted with Open IE techniques in the form of noun phrases or verb phrases that need to be identified and tagged to a single semantic entity or relation in the KG. Since the triples extracted from our dataset via Open IE method comprised many noisy phrases, as well as new entities, such as titles of artworks, that may not be available for mapping in existing databases, entity linking techniques would not suffice in this case. Different from entity linking (that can only link entities already present in external KGs), canonicalization is able to perform clustering for the entities and relations that may not be present in existing KGs, by labelling them as OOV (out of vocabulary) instances. In this work, we chose to perform canonicalization with the help of CESI [159] which is a popular and openly available approach for this task. The CESI approach performs clustering over the non-canonicalized forms of noun phrases for entities and verb phrases for the relations. It leverages different sources of side information for noun phrases and relation phrases such as entity linking, word senses and rule-mining systems for learning embeddings for these phrases using the HolE [118] knowledge graph embedding technique. The clustering is then performed using hierarchical agglomerative clustering (HAC) based on the cosine similarity of the phrase embeddings in vector space. In this manner, different phrases for the same entity or relation were mapped to one

canonicalized form for including in the KG. In total, we obtained 3,789 entity clusters and 3,778 relation clusters from the CESI approach that contained two or more terms.

Representative selection. An important step in the CESI approach is the assignment of representatives for the clusters obtained for the noun and relation phrases. This is decided by calculating a weighted mean of all the cluster members' embeddings in terms of their frequency of occurrence. The phrase closest to this mean is selected as the representative. However, this technique did not work well for our domain-specific and noisy dataset and many undesirable errors were noticed. For example, an entity cluster obtained from CESI was: *Olga_Khokhlova*, *olga*, *khokhlova*, *picasso*. Since *Picasso* is the most frequent entity in the dataset, it was chosen as representative by CESI, but this is clearly wrong since *Picasso* and *Olga* are different entities. There were several other errors observed, e.g., all days of the week were clustered together in one cluster. This could be a result of the embedding and contexts of the days of the week to be quite similar, hence their vectors would end up together in the vector space. In other cases, the color *blue* occasionally showed up in a cluster of phrases related to color *red*, certain dates got clustered and certain related but not interchangeable words got clustered (*kill* vs *murder* vs *shot*). In some cases, the first name was being replaced by the incorrect full name (not every *david* is *david johnson*). To mitigate the above discussed errors, we had to perform manual vetting of the clusters for verification and selection of the correct cluster representatives which took around 2-3 person hours. During this process, certain clusters, where the entities were different, were removed (such as the cluster with days of the week). After this, the entities and relations were canonicalized as per their chosen cluster representatives leading to a total of 35,305 unique entities and 33,448 unique relations in the final KG.[7]

Entity typing

Since a schema or ontology was not employed to extract the triples from text, the entities in our KG did not have any entity types implicitly assigned to them. Therefore, we attempted to identify the types of as many entities in our graph as possible. With the help of NER, we assigned the types to the entities that were recognized in the triples. A total of 14,960 entities were typed with this technique to generic types such as Person, Product, ORG, LOC, GPE, NORP and Work_Of_Art, as well as numeric types such as Date, Time and Ordinal. Note that *Work_of_Art* is quite a broad category that includes artworks but also movies, books and various other art forms. Since artworks such as paintings and sculptures are one of the most important entities in our art-historic KG, it is worthwhile to identify the mention and type of these entities. However, generic NER process is neither equipped nor optimized to correctly identify such mentions. Thus, we additionally applied dictionary-based matching. This was done by compiling a large gazetteer of artwork titles by querying Wikidata with the help of the Wikidata Query Service[8] for the names of paintings and sculptures, retrieving approximately 15,000 artwork titles (the details of further work along these lines are described in Section 2.4). In addition, we augmented

[7] It is to be noted that existing canonicalization techniques such as CESI are largely optimized for canonicalization of entities and their performance is considerably worse for relations. We also observed similar results during our analysis.

[8] https://query.wikidata.org/

Table 2.2: Statistics of the KG

Attribute	Total Triples	Unique Entities	Unique Relations	Artworks	Artists
Count	147,510	35,305	33,448	1,397	656

our dictionary with the names of the *artwork* entities from the ArtGraph dataset [28] which contains more than 60,000 artworks derived from DBpedia and WikiArt. If a match was found for an entity in our KG in the compiled dictionary, the type was assigned as *artwork* accordingly. This led to the tagging of further 1,397 entities in our KG as artworks. The dictionary-based matching for artworks was particularly useful in the cases where it was able to correctly identify entities that were wrongly assigned as the *Person* type by NER, such as *la_donna_gravida*, *portrait_of_mary_cassatt* and *st._paul_in_prison*. Similar to artworks, we attempted to additionally identify the names of artists in our triples. While NER could only tag entities as *Person*, we used a dictionary of artist names from Wikidata to identify 656 unique artist entities in our data. These included names of artists such as *Piet Mondrian*, *Edvard Munch* and *Rembrandt*.

However, the process of entity typing described above is only able to identify and tag around half of the entities in our KG. Several domain and corpus-specific challenges acted as bottlenecks during this process. For example, even after filtering, some triples extracted from Open IE contained either subject or object noun phrases that were generic and did not correspond to any named entity. Examples of such phrases include *essay*, *anthology*, *periodical*, or *album* that are present in triples such as ⟨*album, be_shown_in, Paris*⟩. Without designing a custom ontology for this corpus, such entities cannot be hoped to be correctly typed.

The categorization of the relations in the KG is a particularly complicated task due to the wide variety of relations extracted from the Open IE process. Few of the most frequent relations in the KG are *will*, *be_in*, *have*, *show*, *paint*, *work* etc. We estimated that the types of the entities could be utilized to find patterns and link the most popular edges in the KG to the relations in existing graphs such as Wikidata or ArtGraph. However, preliminary analysis led to some interesting observations. Firstly, we noted the presence of multiple relations between pairs of entities in the KG. For example, *Picasso* and *June* are connected by various relations such as *will_be*, *work* and *take_trip_in* that were extracted from different contexts in the corpus and represent separate meaningful facts. Furthermore, in general, there are several different types of semantic relations between the popular entity types in our KG. For instance, two entities of the type *artist* are connected by several relations including *work*, *meet*, *know_well*, *be_with*, *friend_of* and *be_admirer_of*. While this variety indicates that a large number of interesting facts have been derived by Open IE in the absence of a fixed and limiting schema, normalizing the relations to improve the quality of the KG is a difficult task that requires further work.

2.3.3 Art-historic knowledge graph

The statistics of the KG generated from the steps as described in the previous section are shown in Table 2.2. After obtaining this refined set of triples for the first version of the art-historic KG, we performed a preliminary analysis of the graph to derive useful

insights with the help of the NetworkX[9] package. To understand the graph structure, the number of disconnected components of the graph was measured before and after the canonicalization step. It was noticed that the number of disconnected components was reduced to around 1,500 (down from 2,500) after clustering with CESI. This indicates that canonicalization of entities and relations improved the quality of the knowledge graph by removing unnecessary disconnected parts that were created through redundant triples. Additionally, we also performed node centrality on the graph using eigenvector centrality [19] and link analysis using PageRank [126]. For both the measures, the node for *Pablo Picasso* was the most central. This confirms the property of the underlying dataset which is focused on *Picasso*. Other central nodes discovered were corresponding to popular words in the corpus such as *work*, *artist*, *painting* etc. Overall, it is promising to witness that centrality analysis of the generated KG conforms well regarding the main entities and topics of the underlying corpus. A hand-picked example of a subset of the neighborhood of the entity *Picasso* is shown in Fig. 2.4.

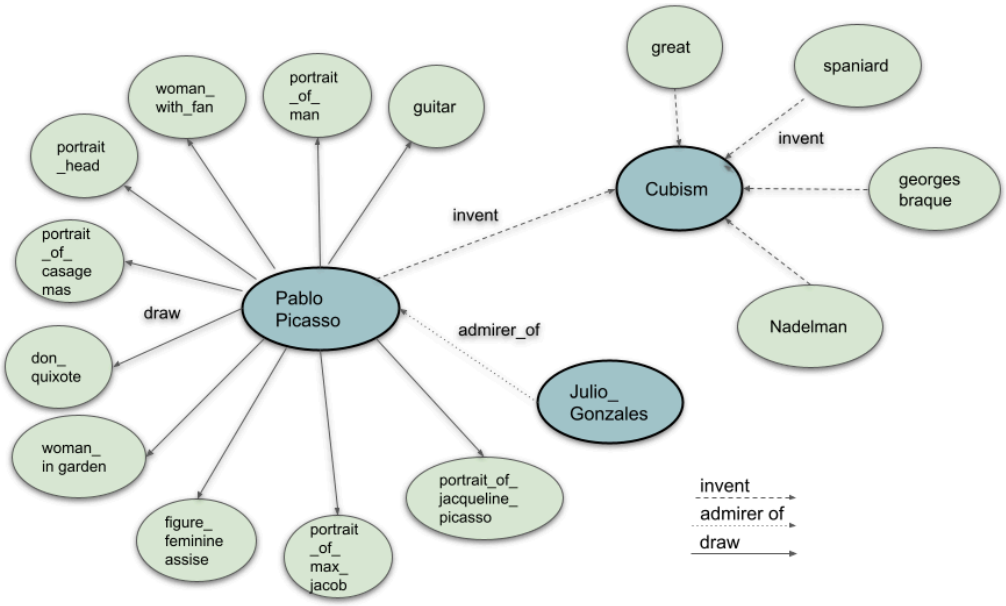


Figure 2.4: Illustration of a subset of the KG

Implementation

Taking cue from related work [28], we have encoded our KG data into Neo4j[10] which is a no-SQL graph database that provides an efficient way of capturing the diverse connections between the different entities of our knowledge graph. Additionally, the knowledge graph stored in the Neo4j database can be queried easily with the help of the Cypher language for enabling data exploration and knowledge discovery. Fig. 2.5 shows the results of a sample query that can be executed on the KG - persons and/or art styles that *Picasso* influenced or was influenced by. In some cases, interesting connections with other relevant

[9] https://pypi.org/project/networkx/
[10] https://neo4j.com

entities are also retrieved, thus providing useful cues for further exploration of the data in the KG for domain experts as well as interested users.

Evaluation

Due to the lack of any gold standard for direct comparison, the evaluation of the resulting KG proved challenging. While an absolute measure of the coverage of any KG is a non-trivial task due to the open world assumption [54], we attempted to perform limited evaluation in terms of the coverage of the KG in a semi-automated fashion. For this, we first created a subset of Wikidata [160] by querying for triples about the entity *Picasso* and used this as the knowledge graph for comparison. This is motivated by the fact that Wikidata contains high quality information about *Picasso* and the entity linking used in our pipeline performs the linking to Wikipedia (hence, Wikidata) entities. Therefore, it was likely to have a higher match between the surface forms of entities in our KG to the Wikipedia entities, as compared to other datasets such as DBpedia.

From the obtained Wikidata subset, 100 triples were randomly selected that related to information about *Picasso* as well as about museums that owned his works. Upon careful manual inspection (independently by three annotators) and resolution of conflicts with discussions, it was measured that the facts represented in 43% of these triples were also present in our KG as a direct match or in a different form with the same meaning. Notably, our KG was missing information about the museums that own Picasso's works, this is because our underlying corpus is also lacking comprehensive information on this topic. Therefore, triples relating to museums from Wikidata could not be matched. Additionally, we checked how many of our entities and entity pairs are written in exactly the same way as in the Wikidata graph. Overall, around 12% of entities and 10% of entity pairs in our graph have exact matches in Wikidata. These preliminary results are promising and point towards the need for a domain-oriented construction process for further improvement of the art-historic KG. In particular, the precision of the triples in art-historic KG is more important to the users and therefore, verification for the triples that were extracted from our dataset but are not found in Wikidata needs to be conducted by enlisting the help of domain experts. While we have performed a semi-automated evaluation for the first version of our KG, a more rigorous and thorough evaluation of the correctness of the facts is certainly imperative before this KG can be useful to a non-expert user. One way to ensure this would be to maintain the provenance and of the facts in the KG, in terms of their source document as well as their confidence measure. This could also facilitate a fair and complementary manual evaluation in terms of precision and recall which could provide further insights.

2.3.4 Observations

Section 2.3.2 described our attempt at constructing a domain-oriented knowledge graph for the art domain in an automated fashion with Open IE techniques. Due to the noisy and heterogeneous dataset that is typical of digitized art-historic collections, we encountered challenges at various steps of the KG construction process. During the very first step, it was difficult to correctly identify the mentions of artworks (i.e. titles of paintings) in the dataset due to the noise and inherent ambiguities. This domain-specific issue needs further attention in order to improve the quality as well as coverage of the resulting KG and is the main topic of discussion in the next section(2.4).

While the Open IE approach allowed for the extraction of a wide variety of entities and relations, this led to canonicalization becoming a complicated task. We observed that existing techniques for canonicalization on generic datasets, such as CESI, do not show comparable performance for domain-specific dataset. Another important aspect is the incomplete tagging of the various types of entities obtained from Open IE. Attributed yet again to the noise in the process, as well as to lack of any underlying schema, many entities could not be assigned their correct type. This task needs further exploration for the enrichment of the KG.

With regard to the implementation of the KG pipeline, while we have so far used off-the-shelf tools and libraries like SpaCy, Stanford CoreNLP and CESI, further fine-tuning is needed for the task of domain-specific KG construction. It would also be worthwhile to explore and evaluate the performance with other available tools such as Flair [2] and Blink [172] for entity recognition, linking and typing, as well as OpenIE [91] and MinIE [56] for the extraction of triples.

The evaluation of the art-historic KG is also a crucial task worth discussing. While we have performed a semi-automated evaluation for the first version of our KG, a more rigorous and thorough evaluation of the correctness of the facts is certainly imperative before this KG can be useful to a non-expert user. For this, it is necessary to closely collaborate with the domain experts.

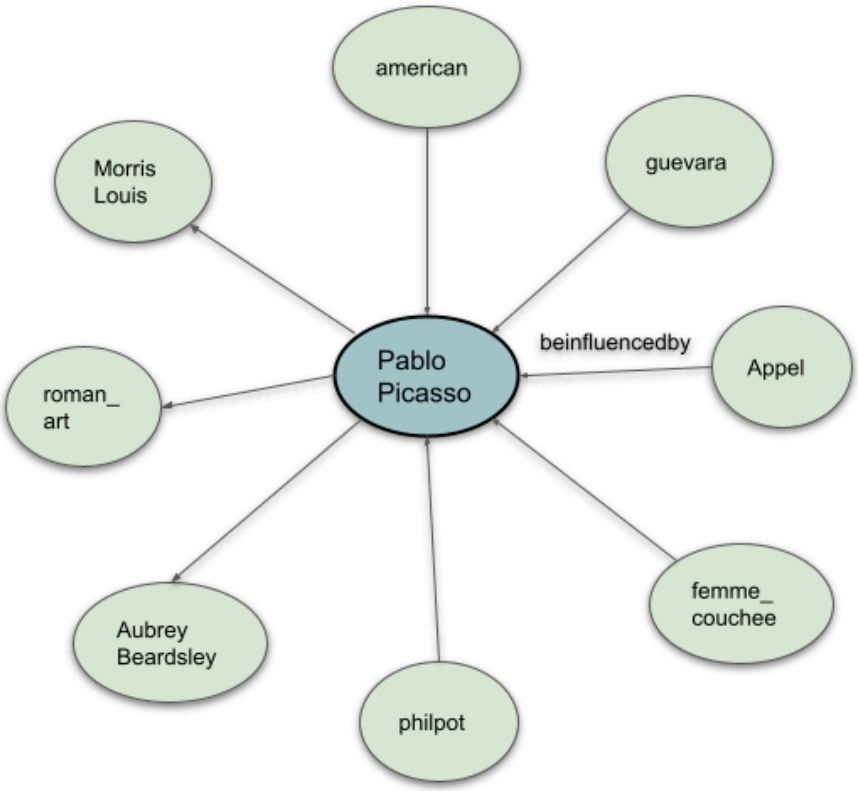


Figure 2.5: Illustration of a subset of KG, depicting the influence of and on *Picasso* (corresponding query: `MATCH p=(s)-[:beinfluenceby]-(o) WHERE s.name="Pablo Picasso" RETURN p`)

In the next section, we put our focus to one of the first challenges of domain-specific KG construction, which is NER for domain-specific entities. We discuss the particular intricacies for identifying artwork titles and motivate the necessity of generation of training data for this problem. Further, we present the merits of this approach in terms of improved NER performance.

2.4 Named Entity Recognition for Artworks

Identification of mentions of artworks seems, at first glance, to be no more difficult than detecting mentions of persons or locations. But the special characteristics of these mentions makes this a complicated task which requires significant domain expertise to tackle. Artworks in fine-art collections are typically referred to by their titles, these titles could have been assigned by artists or, in the case of certain old and ambiguous artworks, by collectors, art historians, or other domain experts. Due to the ambiguities that are inherent in artwork titles, their identification from texts is a challenging task. As an example, consider the painting titled '*girl before a mirror*' by Pablo Picasso — this title merely describes in an abstract manner what is being depicted in the painting and thus, it is hard to identify it as a named entity without knowing the context of its mention. Similarly, consider the painting with the title '*head of a woman*' — such phrases can be hard to be distinguished as named entities from the surrounding text due to their generality. Yet, such descriptive titles are common in the art domain, as are abstract titles such as *'untitled'*.

To circumvent ambiguities present in art-related documents for human readers, artwork titles are typically formatted in special ways — they are distinctly highlighted with capitalization, quotes, italics or boldface fonts, etc. which provide the required contextual hints to identify them as titles. However, the presence of these formatting cues cannot be assumed or guaranteed, especially in texts from art historical archives, due to adverse effects of scanning errors on the quality of digitized resources [87]. The formatting cues for artwork titles might vary from one text collection to the other. Therefore, the techniques for identifying the titles in digitized resources need to be independent of formatting and structural hints, making the task even more complex. Moreover, the quality of digitized versions of historical archives is adversely affected by the OCR scanning limitations and the resulting data suffers from spelling mistakes as well as formatting errors. The issue of noisy data further exacerbates the challenges for the NER task [138].

2.4.1 Types of errors in detecting artwork titles

We introduce the named entity type *artwork* that refers to the most relevant and dominant artworks in our dataset (as mentioned in Section 2.3.1) of digitized collections, i.e. paintings and sculptures.[11] In order to systematically highlight the difficulties that arise when trying to recognize the *artwork* entity type in practice, we categorize and discuss the different types of errors that are commonly encountered as follows — failure of detection of a *artwork* named entity, incorrect detection of the named entity boundaries, and incorrect tagging of the *artwork* with a wrong type. Further, there are also errors due to nested named entities and other ambiguities.

[11]The label *artwork* for the new named entity type can be replaced with another such as *fine-art* or *visual-art* without affecting the proposed technique.

Incorrectly missed artwork title. Many artwork titles contain generic words that can be found in a dictionary. This poses difficulties in the recognition of titles as named entities. E.g., a painting titled *'a pair of shoes'* by Van Gogh can be easily missed while searching for named entities in unstructured text. Such titles can only be identified if they are appropriately capitalized or highlighted, however this cannot be guaranteed for all languages and in noisy texts.

Incorrect artwork title boundary detection. Often, artworks have long and descriptive titles, e.g., a painting by Van Gogh titled *'Head of a peasant woman with dark cap'*. If this title is mentioned in text without any formatting indicators, it is likely that the boundaries may be wrongly identified and the named entity be tagged as *'Head of a peasant woman'*, which is also the title of a different painting by Van Gogh. In fact, Van Gogh had created several paintings with this title in different years. For such titles, it is common that location or time indicators are appended to the titles (by the collectors or curators of museums) in order to differentiate the artworks. However, such indicators are not a part of the original title and should not be included within the scope of the named entity. On the other hand, for the painting titled *'Black Circle (1924)'* the phrase *'(1924)'* is indeed a part of the original title and should be tagged as such. There are many other ambiguities for artwork titles, particularly for older works that are typically present in art historical archives.

Incorrect type tagging of artwork title. Even when the boundaries of the artwork titles are identified correctly, they might be tagged as the wrong entity type. This is especially true for the artworks that are directly named after the person whom they depict. The most well-known example is that of *'Mona Lisa'*, which refers to the person as well as the painting by Da Vinci that depicts her. There are many other examples such as Picasso's *'Jaqueline'*, which is a portrait of his wife Jaqueline Rogue. Numerous old paintings are portraits of the prominent personalities of those times and are named after them such as *'King George III'*, *'King Philip II of Spain'*, *'Queen Anne'* and so on. Many painters and artists also have their self-portraits named after them — such artwork titles are likely to be wrongly tagged as the *person* type in the absence of contextual clues. Apart from names of persons, paintings may also be named after locations such as *'Paris'*, *'New York'*, *'Grand Canal, Venice'* and so on and may be incorrectly tagged as *location*.

Nested named entities. Yet another type of ambiguity involving both incorrect boundaries and wrong tagging can occur in the context of nested named entities, where paintings with long titles contain phrases that match with other named entities. Consider the title *'Lambeth Palace seen through an arch of Westminster Bridge'* which is an artwork by English painter Daniel Turner. In this title, *'Lambeth Palace'* and *'Westminster Bridge'* are both separately identified as named entities of type *location*, however, the title as a whole is not tagged as any named entity at all by the default SpaCy NER tool. Due to the often descriptive nature of artwork titles, it is quite common to encounter *person* or *location* named entities embedded within the artwork titles which lead to confusion and errors in the detection of the correct *artwork* entity. Therefore, careful and correct

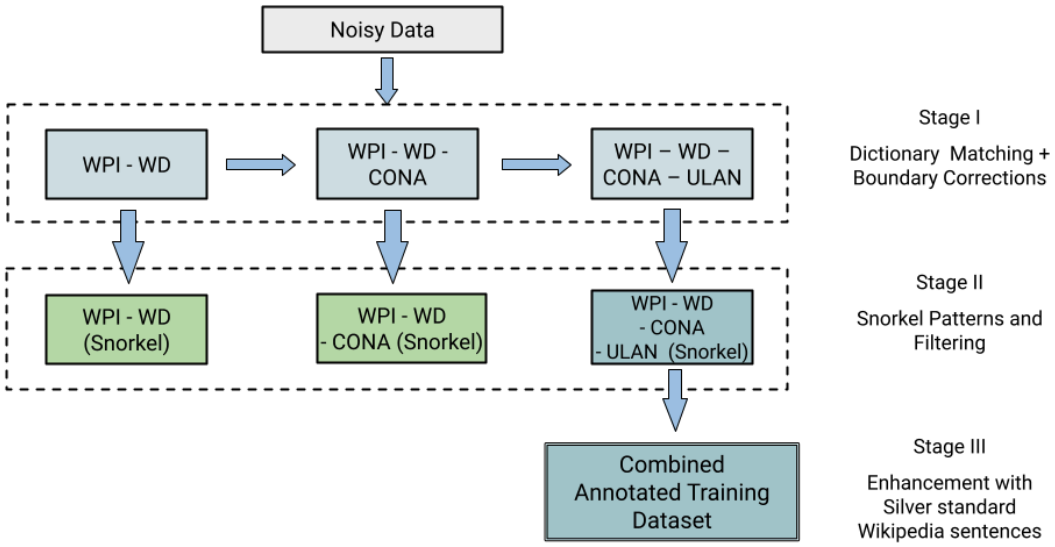


Figure 2.6: Overview of the framework showing the progressive improvements of the training datasets (as described in Section 2.4.2 and summarized in Table 2.3). Each stage illustrates the enhancements by referring to the datasets obtained with the corresponding steps.

boundary detection for the entities is imperative for good performance.[12]

The above examples demonstrate the practical difficulties for automatic identification of artwork titles. In our dataset, we encountered many additional errors due to noisy text of scanned art historical archives, as already illustrated in Fig. 2.2, that cannot be eliminated without manual efforts. Due to the innate complexity of this task, NER models need to be trained with domain-specific named entity annotations, such that the models can learn important textual features to achieve the desired results. We discuss in detail our approach for generating annotations for NER from a large corpus of art related documents in the next section.

2.4.2 Training data generation

Here, we describe our three stage framework for generating high-quality training data for the NER task without the need for manual annotations (Fig. 2.6). These techniques were geared towards tackling the challenges presented by noisy corpora that are typical of art historical archives, although they can be applicable for other domains as well. The framework can take structured or unstructured data as input and progressively add and refine annotations for *artwork* named entities. A set of training datasets is obtained at the end of each stage, with the final annotated dataset being the best performing version. While the artwork titles are multi-lingual, we focus on English texts in this work and plan to extend to further languages in future efforts. We describe the three stages of the framework and the output datasets at each stage.

[12]Details on how our approach handles this complexity are presented in Section 2.4.2.

Stage I - Dictionary-based matching for labelling artwork titles

In the first stage, we aimed to match and correctly tag the artworks present in our corpus as named entities with the help of entity dictionaries to obtain highly precise annotations. Apart from extracting the existing artwork titles from the structured part of the WPI dataset (1,075 in total), we leveraged other cultural resources that have been integrated into the public knowledge bases such as Wikidata, as well as linked open data resources such as the Getty vocabularies for creating these dictionaries. As a first step, we collected available resources from Wikidata to generate a large entity dictionary or *gazetteer* of artwork titles in an automatic way. As already mentioned in Section 2.3, to generate the entity dictionary for titles, Wikidata was queried for names of artworks, specifically for names of paintings and sculptures. Since our input dataset was inherently multilingual, there were many instances where the original non-English titles of paintings were mentioned in the texts. In order to match such titles, we added all the alternate names of the paintings and sculptures to our list belonging to the 7 major languages present in the dataset apart from English (French, German, Italian, Dutch, Spanish, Swedish and Danish). A large variety of artwork titles were obtained from Wikidata, with the shortest title belonging to a painting being just a few characters (*'C-B-1'*), while the longest title having 221 characters in total (*'Predella Panel Representing the Legend of St. Stephen ... '*). It was noticed that quite a few of the titles having only one word were highly generic, for instance, '*Italian*', *'Winter'*, *'Landscape'*, *'Portrait'* etc. Matching with such titles was contributing to errors in the annotation process, since common words in the description of the artworks were being wrongly tagged as the *artwork* named entity. In order to maintain high precision of annotations in the first stage, the titles having only one word were removed from the list even at the slight expense of missed tags for some valid artwork titles. Since several artwork titles are identical to location names such as '*Germania'*, '*Olympia'* which can lead to errors while tagging the named entity to the correct type, such titles were also ignored. Overall, around 5% of the titles were removed in this manner.[13] A combined list of approximately 15,000 titles in different languages was obtained, the majority of the titles being in English. The large variety and ambiguity observed in the titles extracted from Wikidata further confirmed that the NER for artwork titles is a non-trivial task. Due to inconsistencies in the capitalization of the words in the title found on Wikidata, as well as in the mention of titles in our dataset, the titles had to be uniformly lower-cased to enable matching. The annotations obtained from the combined WPI and Wikidata entity dictionary resulted in the first version of the training dataset, referred to as *WPI-WD*.

Furthermore, we explored the Getty vocabularies, such as CONA and ULAN, that contain structured and hand-curated terminology for the cultural heritage domain and are designed to facilitate shared research for digital art resources. The Cultural Objects Named Authority (CONA) vocabulary[14] comprises titles of works of art and architecture. Since these are contributed and compiled by an expert user community, these titles are highly precise and can lead to good quality annotations. A total of 3,013 CONA titles were added to the entity dictionary. The Union List of Artist Names (ULAN)[15] contains names of artists, architects, studios and other bodies. We mainly extracted artist names

[13] one-word titles are encountered during training in Stage III.
[14] Getty CONA (2017), http://www.getty.edu/research/tools/vocabularies/cona.
[15] Getty ULAN (2017), http://www.getty.edu/research/tools/vocabularies/ulan.

from this list (899,758 in total) and tagged them in our corpus via matching, with the motivation of providing additional context for the identification of artwork titles through pattern learning. Different versions of the dataset were generated after the iterative enhancements in annotations by the use of CONA titles and ULAN names, referred to as *WPI-WD-CONA* and *WPI-WD-CONA-ULAN* respectively.

In all cases, the simple technique of matching the dictionary items over the words in our dataset to tag them as *artwork* entities did not yield reasonable results. This was mainly due to the generality of the titles. As an example, consider the painting title *'three girls'*. If this phrase would be searched over the entire corpus, there could be many incorrect matches where the text would perhaps be used to describe some artwork instead of referring to the actual title. To circumvent this issue of false positives, we first extracted named entities of all categories as identified by a generic NER model (details in Section 2.4.3). Thereafter, those extracted named entities that were successfully matched with an artwork title in the entity dictionary, were considered as artworks and their category was explicitly tagged as *artwork*. Even though some named entities were inadvertently missed with this approach, it facilitated the generation of high-precision annotations from the underlying dataset from which the NER model could learn useful features.

Improving named entity boundaries. As discussed in Section 2.4.1, there can be many ambiguities due to partial matching of artwork titles. Due to the limitations of the naive NER model, there were many instances where only a part of the full title of artwork was recognized as a named entity from the text, thus it was not tagged correctly as such. To improve the recall of the annotations, we attempted to identify the partial matches and extend the boundaries of the named entities to obtain the complete and correct titles for each of the datasets obtained by dictionary matching. For a given text, a separate list of matches with the artwork titles in the entity dictionary over the entire text were maintained as *spans* (starting and ending character offsets), in addition to the extracted named entities. It is to be noted that the list of *spans* included many false positives due to matching of generic words and phrases that were not named entities. The overlaps between the two lists were considered, if a *span* was a super-set of a named entity, the boundary of the identified named entity was extended as per the *span* offsets. For example, consider the nested named entity from the text *"..The subject of the former (inv. 3297) is not Christ before Caiaphas, as stated by Birke and Kertesz, but Christ before Annas.."* , the named entities *'Christ'*, *'Caiaphas'* and *'Annas'* were separately identified initially. However, they were correctly updated to *'Christ before Caiaphas'* and *'Christ before Annas'* as *artwork* entities after the boundary corrections, thus resolving the particularly challenging issue of missing or wrong tagging for nested named entities. Through this technique, many missed mentions of artwork titles were added to the training datasets generated in this stage, thus improving the recall of the annotations and the overall quality of the datasets.

Stage II - Filtering with Snorkel labelling functions

Identification of artwork titles as named entities from unstructured and semi-structured text can be aided with the help of patterns found in the text. To leverage these patterns, we use Snorkel, an open source system that enables the training of models without hand

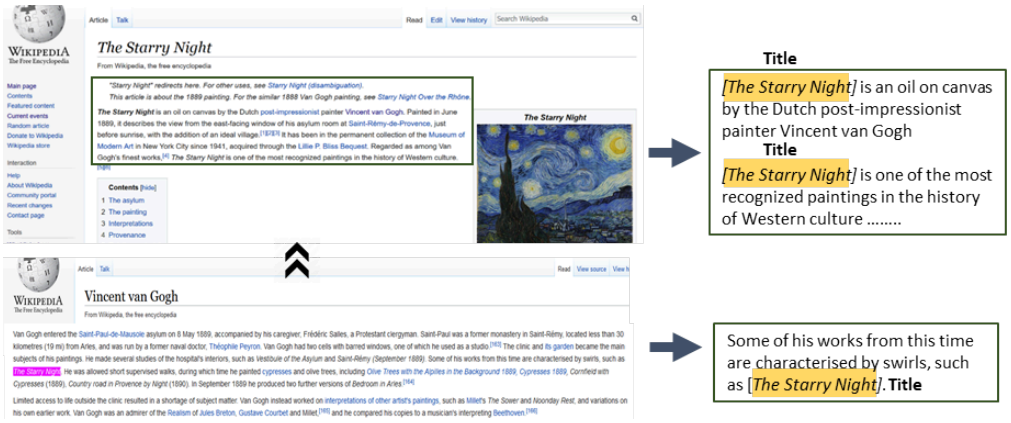


Figure 2.7: Getting annotated sentences from Wikipedia

labeling the training data [134] with the help of a set of labelling functions and patterns. It combines user-written labelling functions and learns their quality without access to ground truth data. Using heuristics, Snorkel is able to estimate which labelling functions provide high or low quality labels and combines these decisions to a final label for every sentence. This functionality is used for deciding whether an annotated sentence is of high-quality, such that it is retained in the training data while the low-quality sentences can be filtered out. Since the training dataset contained a number of noisy sentences that are detrimental to model training, Snorkel helped in reducing the noise by identifying and filtering out these sentences, while at the same time increasing the quality of the training data.

Based on the characteristics of the training data, a set of seven labelling functions were defined to capture observed patterns. For example, one such labelling function expresses that a sentence is of high-quality if it contains the phrase "*attributed to*" that is preceded by a *artwork* annotation and also succeeded by a *person* annotation. This pattern matches many sentences containing painting descriptions in auction catalogues, which make up a large part of our dataset. Another labelling function expresses that a sentence is a low-quality sentence, if it contains less than 5 tokens. With this pattern many noisy sentences are removed that were created either by OCR errors as described in Section 2.4.1 or by sentence splitting errors that were caused due to erroneous punctuation. By only retaining the sentences that are labeled as high-quality by Snorkel, the amount of training data is drastically reduced, as can be seen in Table 2.3. The resulting datasets include annotations of higher quality that can be used to more efficiently train an NER model while reducing the noise. As an example, in the case of the WPI-WD dataset (that contains annotations obtained from matching titles in the combined entity list from WPI titles and Wikidata titles), using Snorkel reduces the number of sentences to 3.2% of the original size, while only reducing the number of artwork annotations to 25.5% of the previous number.

At the end of this stage, we obtained high-quality, shrunk down versions of all three training datasets that led to improved performance of the NER models trained on them.

Stage III - Enhancements with silver standard training data

Despite efforts for high precision in Stage I, one of the major limitations of generating named entity annotations from art historical archives is the presence of errors in the

Table 2.3: Statistics of datasets

Training Dataset	Sentences	Annotations	Unique entities
Ontonotes5	185,254	1,650	-
WPI-WD	13,383,185	1,933,119	36,720
WPI-WD-CONA	13,383,185	1,951,070	37,271
WPI-WD-CONA-ULAN	13,383,185	1,875,711	36,715
WPI-WD (Snorkel)	437,026	492,192	21,838
WPI-WD-CONA (Snorkel)	436,953	496,591	22,027
WPI-WD-CONA-ULAN (Snorkel)	433,154	482,562	21,684
Wikipedia	1,628	1,835	587
Combined Annotated Dataset	434,782	484,397	22,271

training data. Since the input dataset consists of noisy text, it is inevitable that there would be errors in the matching of artwork titles as well as in the recognition of the entity boundaries. To enable an NER model to further learn the textual indicators present in the dataset for identification of artworks, in this stage we augmented our best performing training dataset with clean and well-structured silver standard[16] annotations derived from Wikipedia articles that proved very useful for NER training. To find such sentences, firstly, we searched for the Wikipedia pages of all the artwork titles in English wherever applicable; a total of 2,808 pages were found. We then extracted the relevant sentences that mentioned the artwork title from these pages. To obtain more sentences, we also leveraged the link structure of Wikipedia and mined relevant sentences from the different Wikipedia articles that, in turn, referred to a Wikipedia article of an artwork. Several previous works have utilized the anchor texts and the tagged categories present in Wikipedia articles to transform sentences into named entity annotations [70, 122, 154]. We followed a somewhat similar approach — for each Wikipedia page referring to an artwork, the back-links, i.e. the URLs of the pages that referred to this page were collected. The pages were searched for the relevant sentences that contained an outgoing link to the Wikipedia page of the artwork, while also making sure that anchor text of the outgoing link was identical to the title of the artwork. These sentences were extracted and the anchor texts of the sentences was tagged as an *artwork*, serving as accurate annotations for this category. In this stage, a total of 1,628 sentences were added as silver standard annotation data to the training set. The process is illustrated in Fig. 2.7. This data provided correct and precise textual patterns that were highly indicative of the artwork titles and led to a considerable boost in training data quality. This dataset was augmented to the best performing dataset obtained from the previous stages (*WPI-WD-CONA-ULAN (Snorkel)*) to generate a combined annotated dataset as the final result of the framework. It is to be noted that at this stage, artwork titles having a single word were also included in the annotations such that the trained model could learn from them. Overall, the final model is expected to show a more favorable performance towards multi-word titles. However, the number of false positives for one-word titles would be lower due to high quality annotations from the silver-standard annotations.

[16]The examples are not manually annotated by experts but the annotations are derived in an automatic fashion, therefore silver standard data is often lower in quality compared to gold standard data.

2.4.3 Evaluation and results

In this section, we discuss the details of our experimental setup and present the performance results of the NER models when trained on the annotated datasets generated with our approach.

Experimental setup

The input dataset to our framework consisted of art-related texts in many different languages including English, French, German, Italian, Dutch, Spanish, Swedish and Danish among others. After removing all non-English texts and performing initial preprocessing, including the removal of erroneous characters, the dataset included both partial sentences such as artwork size related entries as well as well-formed sentences describing the artworks. This noisy input dataset was transformed into annotated NER data through the three stages of our framework as described in Section 2.4.2. In order to evaluate and compare the impact on NER performance with improvements in quality of the training data, we trained two well-known machine learning based NER models, SpaCy and Flair, for the new entity type *artwork* on different variants of training data as shown in Table 2.3 and measured their performance.

Baselines

None of the existing NER systems can identify titles of artworks as named entities out-of-the-box. While previous works such as [59] and [102] consider a broad 'art' entity type, they do not include paintings and sculptures which are the primary focus of this work. Thus, these could not serve as baselines for comparison. The closest NER category to artwork titles was found in the Ontonotes5 dataset[17] as *work_of_art*. This category refers not only to artworks such as paintings and sculptures, but also covers a large variety of cultural heritage objects including movies, plays, books, songs etc. In this work, we seek to perform NER for a particular subset of this category, i.e. paintings and sculptures. Therefore, we aim to train the NER models to perform the complex task of learning the features for paintings and sculptures, while at the same time separating them from other cultural heritage objects such as book, music etc. For the lack of alternatives, we have leveraged the *work_of_art* NER category in our work for setting up a naive baseline in which the training was performed on more general annotations. With this baseline, we will compare the improvements in NER performance obtained by retraining the tools on our semi-automatically generated corpus with the specialized *artwork* entity type.

To quantify the performance gains from annotations obtained at each stage, SpaCy and Flair NER models were re-trained on each of the generated datasets for a limited number of epochs (as per computational constraints), with the training data batched and shuffled before every iteration. In each case, the performance of the re-trained NER models was compared with the *baseline* NER model (the pre-trained model without any specific annotations for artwork titles). As the underlying Ontonotes dataset does not have *artwork* annotations, the named entity type *artwork* was not applicable for the baseline models of SpaCy and Flair. Therefore, a match with the entity type *work_of_art* was considered as a true positive during the evaluations. In the absence of a gold standard

[17] https://catalog.ldc.upenn.edu/LDC2013T19

dataset for NER for artwork titles, we performed manual annotations and generated a test dataset on which the models could be suitably evaluated.

SpaCy. The SpaCy[18] library is popular for many natural language processing tasks including named entity recognition. SpaCy text processing tools were employed for tokenization and chunking of the texts before the identification of the named entities. The pre-trained English model of SpaCy has been trained on the Ontonotes5 dataset which consists of different types of texts including telephone conversations, news-wire, newsgroups, broadcast news etc. Since this dataset is considerably different from historical art document collections, the pre-trained NER model showed poor performance for named entity recognition in the cultural heritage domain, even for the common named entity types (*person*, *location* and *organization*). With regards to artwork titles, very few were identified as named entities and many among those were wrongly tagged as names of persons or locations, instead of being correctly categorized as *work_of_art*. With the pre-trained SpaCy NER model as baseline, the model was trained on the datasets for 10 epochs each and the performance evaluated.

Flair. Similar to SpaCy, Flair [2] is another widely used deep-learning based NLP library that provides an NER framework in the form of a sequence tagger, pre-trained with the Ontonotes5 dataset. The best configuration reported by the authors for the Ontonotes dataset, was re-trained with a limited number of epochs in order to define a baseline to compare against the datasets proposed in this paper. The architecture of the sequence tagger for the baseline was configured to use stacked GloVe and Flair forward and backward embeddings [3, 129]. For training the model the following values were assigned to the tagger hyper-parameters: learning rate was set to 0.1, and the number of epochs was limited to 10. These values and the network architecture were kept throughout all the experiments in order to achieve a fair comparison among the training sets.

It is to be noted that the techniques for improving the quality of NER training data that are proposed in this work are independent of the NER model used for the evaluation. Thus, SpaCy and Flair can be substituted with other re-trainable NER systems.

Manual annotations for test dataset

To generate a test dataset, a set of texts were chosen at random from the dataset, while making sure that this text was representative of the different types of document collections in the overall corpus. This test data consisted of 544 entries (with one or more sentences per entry) and was carefully excluded from the training dataset such that there was no entity overlap between the two. The titles of paintings and sculptures mentioned in this data were manually identified and tagged as named entities of *artwork* type. The annotations were performed by two non-expert annotators (from among the authors) independently of each other in 3 to 4 person hours with the help of the *Enno*[19] tool and their respective annotations were compared afterwards. The task of manual annotation was found challenging due to the inherent ambiguities in the dataset (Section 2.4.1) and lack of domain expertise. The annotators disagreed on the tagging of certain phrases as titles on multiple occasions. For example, in the text snippet "*An earlier, independent*

[18] SpaCy: https://spacy.io/
[19] https://github.com/HPI-Information-Systems/enno

watercolor of almost the same view can be dated to circa 1830 (Stadt Bernkastel-Kues; see C. Powell, Turner in Germany, exhibition catalogue, London, Tate Gallery, 1995-96, pp. 108-9, no- 23¿ illustrated in color)..", the artwork mention '*Stadt Bernkastel-Kues*' was missed by one of the annotators. The correct boundaries of the artworks was also disagreed in some cases, such as in the text "*Claude Monet, Rouen Cathedral, Facade, 1894, Oil on canvas [W.1356], Museum of Fine Arts, Boston*" - the artwork title could be '*Rouen Cathedral, Facade*' or '*Rouen Cathedral*'. It was difficult to correctly tag these artwork mentions without having expert knowledge of the art domain, especially with regard to the particular period of art. Due to these reasons, the inter-annotator agreement was quite low. The Fleiss' kappa [49] and Krippendorff's alpha [94] scores were calculated as -1.86 and 0.61 respectively. (A negative Fleiss' kappa score indicates poor agreement, while Krippendorff's alpha values for data should be above 0.667 to be considered useful). The poor inter-annotator agreement reflected by these scores reaffirmed that the task of annotating the artwork titles is difficult, even for humans. Only experts in the particular artwork collections could have perhaps identified the artworks correctly, however such expertise is rarely available or even practical. Therefore, in order to obtain the gold standard test dataset for the evaluation of NER models, the disagreements were manually sorted out with the help of web search to the best of our understanding and a total of 144 entities were positively tagged as *artwork*.

Evaluation metrics

The performance of NER systems is generally measured in terms of precision, recall and F1 scores. The correct matching of a named entity involves the matching of the boundaries of the entity (in terms of character offsets in text) as well as the tagging of the named entity to the correct category. The strict F1 scores for NER evaluation were used in the CoNNL 2003 shared task[20], where the entities' boundaries were matched exactly. The MUC NER task[21] allowed for relaxed evaluation based on the matching of left or right boundary of an identified named entity. In this work, the evaluation of NER was performed only for *artwork* entities and therefore, it was sufficient to check only for the boundary matches of the identified entities. Since there are many ambiguities involved with entity boundaries of artwork titles, as discussed in Section 2.4.1, we evaluated the NER models with both strict metrics based on exact boundary match, as well as the relaxed metrics based on partial boundary matches. The relaxed F1 metric allowed for comparison of the entities despite errors due to wrong chunking of the named entities in the text. Precision, recall, as well as F1 scores obtained for the NER models trained with different training dataset variants are shown in Table 2.4.

Results

The results demonstrated definitive improvement in performance for the NER models that were trained with annotated data as compared to the baseline performance. Since the relaxed metrics allowed for flexible matching of the boundaries of the identified titles, they were consistently better than the strict matching scores for all cases. The training data obtained from Stage I, i.e. the dictionary based matching, enabled an improvement

[20] https://www.clips.uantwerpen.be/conll2003/ner/
[21] https://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Table 2.4: Performance of NER Model trained on different datasets

Training Dataset	Stage	*SpaCy*						*Flair*					
		Strict			*Relaxed*			*Strict*			*Relaxed*		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Default Unannotated (baseline)	–	.14	.06	.08	.22	.08	.12	**.22**	.04	.07	**.29**	.05	.09
WPI-WD	I	.24	.23	.23	.41	.42	.41	.03	.05	.04	.06	.09	.07
WPI-WD-CONA	I	.27	.26	.26	.43	.45	.44	.04	.08	.06	.08	.14	.10
WPI-WD-CONA-ULAN	I	.28	.26	.27	.48	.45	.46	.05	.08	.07	.09	.14	.11
WPI-WD (Snorkel)	II	.31	.28	.30	.50	.49	.50	.07	.12	.08	.12	.21	.15
WPI-WD-CONA (Snorkel)	II	.31	.31	.31	.53	.51	.52	.07	.11	.08	.13	.22	.17
WPI-WD-CONA-ULAN (Snorkel)	II	.32	.33	.33	.55	.51	.53	.09	.16	.11	.14	.24	.18
Wikipedia	III	.17	.13	.15	.38	.30	.34	.12	.34	.17	.21	**.61**	.31
Combined Annotated Dataset	All	**.46**	**.41**	**.43**	**.68**	**.62**	**.65**	.21	**.45**	**.29**	.28	.59	**.38**

in NER performance due to the benefit of domain-specific and entity-specific annotations generated from the Wikidata entity dictionaries and Getty vocabularies, along with the boost from additional annotations by the correction of entity boundaries. Further, the refinement of the training datasets obtained with the help of Snorkel labelling functions in Stage II led to better training of the NER models reflecting in their higher performance especially in terms of recall. To gauge the benefits from the silver standard annotations from Wikipedia sentences, a model was trained only on these sentences(Stage III). It can be seen that the performance of this model was quite high despite the small size of the dataset, indicating the positive impact of the quality of the annotations. The NER models re-trained on the combined annotated training dataset obtained through our framework, consisting of all the annotations obtained from the three stages, showed the best overall performance with significant improvement across all metrics, particularly in terms of recall. This indicates that the models were able to maintain the precision of the baseline while being able to find much more entities in the test dataset. The encouraging results demonstrate the importance of training on high-quality annotation datasets for named entity recognition. Our approach to generate such annotations in a semi-automated manner from a domain-specific corpus is an important contribution towards this direction. Moreover, the remarkable improvement for NER performance achieved for a novel and challenging named entity of type *artwork*, proves the effectiveness of our approach. It would be interesting to extend the techniques for named entity recognition to other important entities such as auctions, exhibitions and art styles in the corpus. Furthermore, this approach is not limited to the cultural heritage domain but can also be adapted for finding fine-grained entity types in other domains, where there is shortage of annotated training data but raw text and dictionary resources are available.

2.5 Summary

In this chapter, we discussed the importance of domain-specific KGs in the context of the cultural heritage domain that poses multiple challenges. We presented our approach to construct an art-historic KG from digitized texts in an automated manner, where existing Open IE tools were leveraged for various stages of the KG construction process. The limitations and challenges while adapting these generic tools for domain-specific datasets

were also presented in detail. Specifically, we looked into the issue of underwhelming performance of NER tools for artwork titles and motivated the need for high-quality annotations for training. We proposed techniques for generating the relevant training data for NER in a semi-automated manner. Experimental evaluations showed that the NER performance can be significantly improved by training on high-quality training data generated with our methods. This indicates that even for noisy datasets, such as digitized art archives, supervised NER models can be trained to perform well.

On the whole, while the discussion in this chapter has shown encouraging results, it has also given clear indications of the points of improvement for creating a more refined and comprehensive version of an art-historic KG which is a pre-requisite for supporting downstream tasks such as search and querying. In the next chapter, we discuss and solve a specific research problem in the context of refinement of knowledge graphs and illustrate how knowledge graph embeddings could play a vital role in this task.

Chapter 3

Discovering Fine-Grained Semantics in Knowledge Graph Relations

"I think it's much more interesting to live not knowing than to have answers which might be wrong."
— Richard P. Feynman

So far we have presented our contributions towards the construction of knowledge graphs and the challenges involved in the process. KGs obtained by automated methods, and even those curated with manual efforts, suffer from several issues in terms of completeness and correctness as outlined in Chapter 1. In order to be useful for downstream applications, it is important to identify these issues and perform refinement of the KGs to improve their quality. In this chapter, we focus on an often overlooked issue in KGs that have been constructed from textual sources - the presence of polysemous relations that convey ambiguous semantics. To tackle this, we propose a data-driven technique to discover fine-grained semantics for the refinement of existing KG relations.

The main contributions highlighted in this chapter are based on the work in Jain et al. [78]. The chapter is organized as follows — firstly, Section 3.1 introduces the issue of polysemous relations in knowledge graphs with the help of examples. Section 3.2 clearly lays down the importance of fine-grained relation semantics by discussing various application scenarios. This is followed by a discussion of previous work related to this topic in Section 3.3. Then, Section 3.4 introduces the problem statement and provides the necessary background on KG embeddings and entity types in the ontologies. In Section 3.5, we describe the details of the proposed method and present the results of our detailed empirical evaluation in Section 3.6. Finally, we conclude and discuss future work in Section 3.7.

3.1 Polysemous Relations in Knowledge Graphs

In Chapter 1, we have established that KGs represent real-world data in the form of $\langle subject, predicate, object \rangle$ triples. Here, *subject* and *object* are chosen from a set of

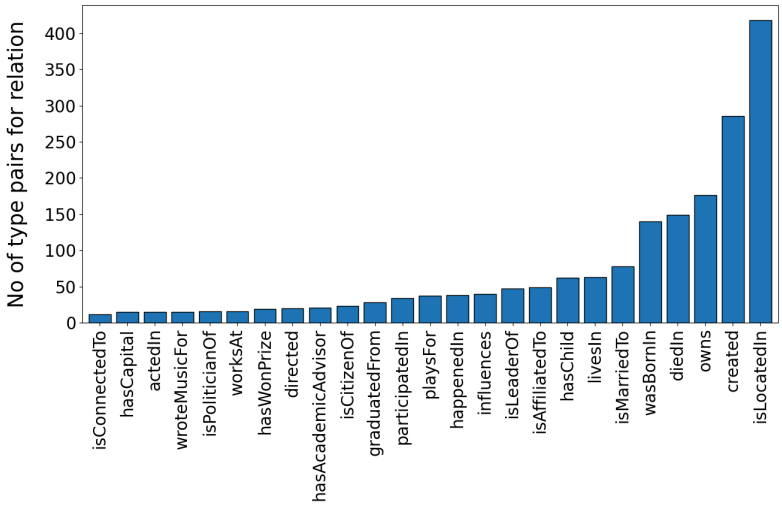


Figure 3.1: The unique type pairs associated with different relations in Yago

entities, while the *predicate* that links the entities to each other belongs to a set of relations. In textual data, the relations are often polysemous by nature, i.e., they exhibit distinct meanings in different contexts. For example, the relation '*part of*' has different semantics in '*..Sahara is part of Africa*' and '*finger part of hand*'. As the triples in KGs are derived from and represent factual information from such texts, ambiguity from texts often makes it way into the KG triples as well. Specifically, the KG relations may represent multiple meanings depending on the *context*, which is defined by the types of the entities being connected by the relations in the case of KG triples.

Relation polysemy in KGs is a particularly important issue due to the widespread application of KGs in several downstream tasks where semantics play a crucial role. However, it has received surprisingly little attention until now. In order to gauge the magnitude of the issue in popular KGs, we analysed the relations in the Yago3 [106] dataset in terms of the number of unique entity type pairs that are connected by a single relation (in the KG triples) without any further semantic specialization. The results are plotted in Figure 3.1. It can be seen that for a majority of the relations, the triples in which they occur contain subject and object entities belonging to various entity types. Among these, many relations such as *owns* and *created* exhibit very high plurality of entity types which indicates that they are quite generic with regards to their meaning. Similar insights were also derived from the NELL-995 [174] dataset, which is a subset of the 995-th iteration of NELL. Table 3.1 shows some examples of the different entity types associated with relations from these KGs.

We advocated that for such relations that are associated with a number of different entity type pairs that are semantically distant from one another, it would be prudent to replace them with sub-relations that have a more distinct meaning according to the context. The exact meanings of the sub-relations could be clearly defined based on the distinct types of the associated entities. Indeed, this underlying idea is derived from the task of word sense disambiguation in Linguistics, as advocated by Firth : 'a word is characterized by the company it keeps' [48]. In the context of KGs, one could say *'a relation is characterized by the entity types it connects'*.

However, it is to be emphasized that while being intuitive, this task is extremely tricky due to a wide variance in the types of the entities in large KGs. Let us consider the

Table 3.1: Examples of multiple semantics of relations

Yago *created*	**NELL *agentBelongsTo-Organization***
(writer, movie)	(politician, politicalparty)
(player, movie)	(country, sportsleague)
(artist, movie)	(sportsteam, sportsleague)
(officeholder, movie)	(coach, sportsleague)
(writer, fictional_character)	(person, charactertrait)
(artist, computer_game)	(televisionstation, company)
(artist, medium)	
(writer, television)	
(company, computer_game)	

relation *created* from the Yago dataset (Table 3.1). While some types such as *television* and *movie* for the *created* relation are semantically similar to one another, other types are quite different, for instance *company* and *writer*. If the relation is trivially replaced by multiple relations based on the different entity type pairs in a straightforward manner, without taking the similarity as well as frequency of these types into account, the resultant sub-relations would end up being remarkably similar to each other, thus leading to a high degree of duplication. Due to the complex hierarchy of classes (entity types)[1] in the underlying ontology, entity types often belong to different granularity levels [76], leading to a broad range of semantic similarity between them. The frequency with which a relation connects different type pairs is also widely variable. It is, therefore, a non-trivial task to decide how to define the sub-relations based on the semantics of the entity types associated with a relation, both in terms of the number of sub-relations as well the subset of entity types that the sub-relations should encompass.

Knowledge graph embedding models have shown a lot of promise for the task of knowledge graph completion and refinement as already discussed in Section 1.3. In essence, these models aim to encapsulate the structure of the KG, as well as the latent semantics of entities and relations, by embedding them in low-dimensional vector space. Previous works have shown that the vector representations obtained from these models can be used for semantic analysis in KGs [84, 85]. We extended this idea further by demonstrating that these vectors also capture relation semantics such that they can be leveraged to successfully identify polysemous relations. The proposed method used these vectors for finding representative clusters in the latent space and derive the fine-grained semantics from polysemous relations in an effective manner.

To the best of our knowledge, the task of fine-grained relation semantics had not been systematically explored in the context of KG relations. In view of this, we provide a formal definition of the task of *fine-grained relation discovery* which refers to the disambiguation of polysemous relations in knowledge graphs and motivate its importance and benefits. We propose a data-driven and scalable method *FineGReS (**Fine-G**rained **R**elation **S**emantics)* to identify multiple sub-relations that capture the different underlying semantics of the relations via clustering in the latent space. Several feature-based baselines were established to show the promise of our embedding-based solution that outperforms a

[1]The terms *class* and *entity type* will be used interchangeably in the rest of the text.

previous, related non-embedding approach in the context of open relation extraction [110] for downstream applications.

3.2 Fine-Grained Relation Semantics

Relation polysemy is quite common in knowledge graphs due to two primary reasons. Firstly, the schema for most large scale KGs that are in use today have been constructed through manual or semi-automated efforts, where the relations between the entities are curated from text. Relations are often abstracted in such KGs for simplification and avoidance of redundancies. This may result in cases where a single relation serves as a general notion between various different types of KG entities and has more than one semantic meaning associated with it. However, due to the diversity of the kinds of associations between the entities, the abstract relations may not be sufficiently representative of the underlying semantics that they are supposed to capture. In addition to this, the fact that these KGs represent real-world facts that are expressed in natural language having inherent ambiguities, contributes further to the relation polysemy in KGs. For instance, the relation phrase '*part of*' represents varied semantics based on its context of biology (*finger part of hand*), organizations (*Google part of Alphabet*), geography (*Amazon part of South America*) and many others. Even KGs that have a large number of different relations can suffer from ambiguous relations, for instance DBpedia has around 300 relations that are relatively well-defined in terms of their entity types, however there exist relations such as *award* and *partOf* that still convey ambiguity. The determination of fine-grained relation semantics in relational data is an important task which can bring substantial benefits to a wide range of use cases as discussed further in this section.

The task of *relation extraction* is essential for information extraction from texts and it continues to be challenging due to the varied semantics of the evolving language. For identifying patterns and extracting relation mentions from text, unsupervised techniques typically rely on the predefined types of relation arguments [29, 66, 143]. Given an existing KG and schema, with the goal to extract facts for a particular relation from a new corpus of text, a distant supervision approach will leverage relation patterns based on the types of entities over the text. As an example, if the relation *created* has been established between a *painter* and *artwork*, then the identification of this relation can be aided by specific patterns in text. However, if the relation *created* is generically defined between any *person* entity and any *work* entity, then the resulting text patterns for this relation will be noisy and varied, therefore may fail to identify the correct fact triples from text. Identifying the different meanings of a relation in different contexts can help with defining concrete patterns for extraction of relation phrases.

This is also useful for identification and *classification of entities* by their types in a knowledge graph. E.g. the target entity of the relation *directed* is likely to be of type *movie* or *play*. If the relations have a wider semantic range, the type of entities cannot be identified at a fine-grained level. For instance, it might be only possible to identify the entity type as *work* and not specifically *movie*, which could adversely affect the performance of further applications such as *entity linking* and *question answering*. Numerous question answering systems that use knowledge graphs as back-end data repositories (KBQA) [33] rely on the type information of the entities to narrow down the search space for the correct answers. Thus, distinct relation semantics in terms of the

types of connected entities are essential for supporting QA applications over KGs.

It is to be noted that the discovery of fine-grained relation semantics is important in the context of *KG refinement*, not being merely limited to already existing datasets, but also in general. KGs usually evolve over time and often in a fragmented fashion, where new facts might be added to a KG that do not strictly conform or can be correctly encapsulated by the existing ontology. Addition of such new facts might easily lead to noisy and abstracted semantics in previously well-defined KG relations. Relation disambiguation would therefore play a important role in identifying new fine-grained sub-relations with precise semantics. The proposed *FineGReS* method is generally applicable and could prove to be incredibly useful in all the above scenarios. Finally, it is also important to note that the approach of determining semantic sub-relations in existing KGs and their ontology can be applied to the very important open challenge of constructing *domain-specific KGs* from new corpora [86]. By way of adding novel relations to already existing ontologies, this work shows the potential and promise of aiding *ontology matching* [125] for supporting new domains.

3.3 Related Work

In this section, we discuss various works related to the semantics of relations in KGs. We also point out the approaches related to KG embeddings that have considered relation semantics and highlight the differences with our approach. In addition, we include a brief overview of the work pertaining to relation extraction from texts to put our work in the correct context.

Relation semantics. While the idea of learning embeddings for *words* by considering their multiple contextual semantics is not new [162], the contextual semantics of existing *relations* in knowledge graphs have not been studied as much. This is due to the fact that most KGs are populated on the basis of a pre-defined ontology where the relations and their semantics have already been fixed [106, 112]. Yet, issues with the relations in such KGs still persist. Kalo et al. [85] have previously presented a detailed analysis on finding and unifying synonymous relations that are found in most large KGs to reduce the number of relations for the sake of better semantics. Similar in spirit, we bring attention to the complementary problem statement of identifying the relations in KGs that exhibit more than one meaning based on different contexts and claim that they should be represented by multiple sub-relations with more precise semantics.

In other related work, Jiang et al. [84] explore the entailment between relations, e.g. the relation *creator* entails *author* or *developer* in the sense that *creator* subsumes the other relations. Similar to our work, the authors leverage the entity type information to solve the multi-classification problem of assigning the child relations to the parent ones. Our problem statement of fine-grained relation refinement is significantly more challenging and impactful in the sense that it involves the identification of novel sub-relations in an unsupervised manner.

Relations and KG embeddings. In the context of relational learning models, few works have looked into KG relations for the goal of learning better embeddings. For instance, Lin et al. [101] advocated the need for learning multiple relation vectors to

capture the fine-grained semantics, however this study was limited in scope and lacked any consideration for complex entity type hierarchies in KGs. In Zhang et al. [184], the authors create a 3-level relation hierarchy which combines similar relations as well splits relations into sub-relations, in order to improve the embeddings for relations. The proposed approach is quite rigid and opaque in terms of the actual semantics of the relations obtained from it. In fact, the number of clusters was predefined for all relations across a dataset, in contrast to the *FineGReS* method that can determine an optimal number of clusters separately for each relation based on the associated entity types. The diverse semantics of relations was also considered by Ji et al. [83] where the authors proposed two different vectors for the relations as well as entities, to capture their meanings and connections with each other. Similarly, Xiao et al. [173] discussed the generation of multiple translation components of relations based on their semantics with the help of a bayesian non-parametric infinite mixture model. However, they do not perform a systematic analysis of the relations semantics and a qualitative evaluation of their approach is missing.

In general, previous works have only discussed the semantics of KG relations in the context of KG embeddings with the primary goal of training better models that can show improvement on the link prediction (or knowledge graph completion) task. However, this work explicitly pays attention to the identification of polysemous relations in the KGs and discovery of the latent relation semantics with the overall goal of knowledge graph refinement and improvement of the quality of the relations in underlying ontology. Relational models have been leveraged as effective and promising enablers for this task instead of being the focal topic of this work. More importantly, none of the previous works have explored the challenges of deriving fine-grained relations from an existing polysemous relation in the presence of complex semantic relationships between the associated entity types, which is quite common for real-world datasets. We present a systematic and data-driven method for this task.

Relation extraction and Open IE. While this work is concerned with relations between entities, it is important to distinguish it from the task of relation extraction from texts. There are many previous approaches that identify relationships between entities in texts and perform clustering on phrases to derive the relations [18, 113, 136, 167], such approaches aim to identify relation patterns that exactly conform to a singular semantic intent. In stark contrast, we aim to find the different semantic intents that may be already present in a single KG relation. Moreover, relation extraction techniques heavily rely on the contextual cues available in the text, whereas the only context available with regard to the relations in KGs is the associated entities and their types. As such, these approaches are indeed not comparable to our work.

Research pertaining to the processing of entity and relation phrases in the context of Open Information Extraction is more relatable to our goals. Previous approaches on the canonicalization of relation phrases (that are present in Open IE triples) have attempted to establish the semantics of the relations by performing clustering over the phrases [53, 159]. Among such approaches, the closest to ours is the work by Min et al. [110] that discusses the ambiguity in the meanings of relation phrases present in Open IE triples such as $\langle Euro, be\ the\ currency\ of, Germany\rangle$ and $\langle authorship, be\ the\ currency\ of, science\rangle$. While this approach concerns with disambiguation of relation phrases in texts rather than relations in KGs, we still consider this work as a baseline approach that does not employ

embeddings for deriving the semantics and compare our embeddings-based approach against it.

3.4 Notations

In this section, we recall the basic concepts and establish the notations that will be used while explaining our approach in the remainder of the chapter.

For a knowledge graph $\mathcal{G}$, the set of unique relations is denoted as $\mathcal{R}$. A KG fact (or triple) $F = \langle e_h, r, e_t \rangle$ consists of the head entity e_h, the tail entity e_t and the relation r that connects them, where e_h and e_t belong to the set of entities $\mathcal{E}$. A given relation $r \in \mathcal{R}$ appears in several triples, forming a subset $\mathcal{G}_r$ of $\mathcal{G}$.

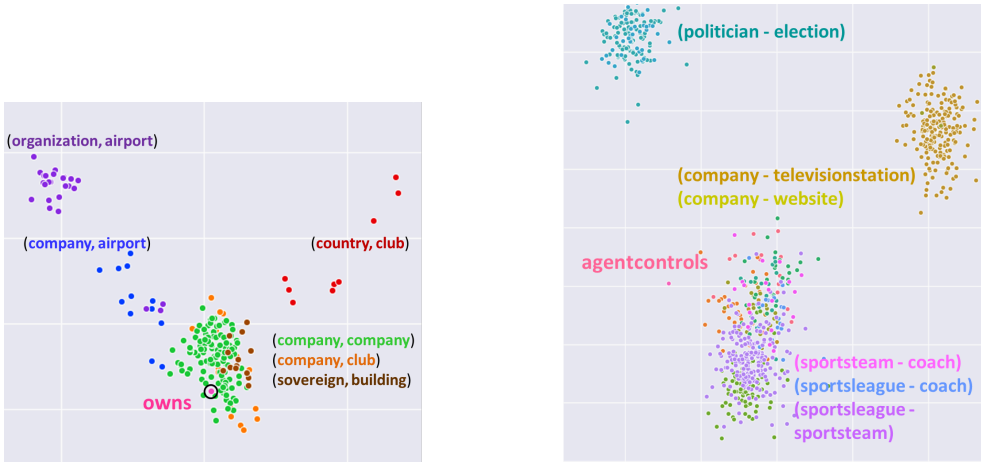
The semantic types or classes of the entities are defined in an ontology associated with a KG that defines its schema. The entities $e \in \mathcal{E}$ are connected with their types by ontological triples such as $\langle e, typeOf, t \rangle$, where $t \in \mathcal{T}$, the set of entity types in the ontology. We define a *type pair* as the tuple $\langle t_h, t_t \rangle$ where $\langle e_h, typeOf, t_h \rangle$ and $\langle e_t, typeOf, t_t \rangle$. A set of unique type pairs for a given relation r and corresponding $\mathcal{G}_r$ is denoted as $\mathcal{P}_r$. Thus we have, $\mathcal{P}_r = \{\langle t_h, t_t \rangle | \langle e_h, typeOf, t_h \rangle, \langle e_t, typeOf, t_t \rangle, \langle e_h, r, e_t \rangle \in \mathcal{G}_r\}$. The total number of such unique type pairs for relation r is denoted by $\mathcal{L}_r$.

As discussed in Section 1.3, knowledge graph embeddings have gained immense popularity and success for representation learning of relational data. They provide an efficient way to capture latent semantics of the entities and relations in KGs. The main advantage of these techniques is that they enable easy manipulation of KG components when represented as vectors in low dimensional space. E.g. in *TransE* [20], for a triple $\langle h, r, t \rangle$ the vectors $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ satisfy the relation $\mathbf{h} + \mathbf{r} = \mathbf{t}$ or $\mathbf{r} = \mathbf{t}$ - $\mathbf{h}$. In this work, we leverage the representational abilities of the embeddings to obtain the semantic vectors for relations expressed in terms of the entities associated with them. For vectors $\boldsymbol{e_h}$, $\mathbf{r}$ and $\boldsymbol{e_t}$ as obtained from an embedding corresponding to a KG triple $\langle e_h, r, e_t \rangle$, we define a vector $\boldsymbol{\Delta_r}$ which is a function of $\boldsymbol{e_h}$ and $\boldsymbol{e_t}$. Further, every $\boldsymbol{\Delta_r}$ vector is mapped to a type pair $\langle t_h, t_t \rangle$ corresponding to the entities e_h, e_t.

Problem definition. Given a relation $r \in \mathcal{R}$ in $\mathcal{G}$, the set of vectors $\{\boldsymbol{\Delta_{r_1}} \boldsymbol{\Delta_{r_2}} ... \boldsymbol{\Delta_{r_{\mathcal{G}_r}}}\}$ for the graph $\mathcal{G}_r$ and the set of type pairs for this relation as denoted by $\mathcal{P}_r$, the goal is to find an optimal configuration of clusters $\mathcal{C}_{opt} = \{\mathcal{C}_1, \mathcal{C}_2 ... \mathcal{C}_N\}$, where the $\boldsymbol{\Delta_{r_i}}$ vectors are uniquely distributed among the clusters i.e. each $\boldsymbol{\Delta_{r_i}} \in \mathcal{C}_j$, $i = 1...|\mathcal{G}_r|$, $j = 1...N$, s.t. an objective function $\mathcal{F}(\mathcal{C}_{opt})$ is maximized. Further, each cluster $\mathcal{C}_j$ represents the semantic union of a *subset of type pairs* from $\mathcal{P}_r$ such that $\exists$ $\boldsymbol{\Delta_{r_i}} \in \mathcal{C}_j$ where $\boldsymbol{\Delta_{r_i}}$ is mapped to one of the type pairs in this subset. Thus, the optimal configuration of clusters corresponds to the optimal number of sub-relations and their fine-grained semantics as defined by the type pairs that they represent. The proposed *FineGReS* method can derive this optimal configuration for the relations of a KG.

3.5 *FineGReS*

In this section, we describe the design and implementation details of the proposed *FineGReS* method for a relation that can be easily scaled to any number of relations in the dataset.



(a) *owns* relation in Yago (b) *agentcontrols* relation in NELL

Figure 3.2: Visualization (after PCA reduction) of relation vectors with associated type pairs

3.5.1 Semantic mapping for facts

For every unique relation r in $\mathcal{G}$, we firstly find the subset of triples $\mathcal{G}_r$ where r appears. To understand the semantics of the entities associated with r, the entities are mapped to their corresponding classes as defined in the underlying ontology. By doing so, we obtain a list of entity type pairs $\langle t_h, t_t \rangle$ for the relation. Note that several entities in $\mathcal{G}_r$ might map to the same type and therefore, a single type pair tuple would be obtained several times. Therefore in the next step, we identify the unique type pairs for a relation r as the set $\mathcal{P}$[2]. At this stage, every triple in $\mathcal{G}_r$ is associated with a type pair $\langle t_h, t_t \rangle \in \mathcal{P}$ that represents the semantics of this triple. For example, for the *created* relation, a triple $\langle DaVinci, created, MonaLisa \rangle$ would be mapped to $\langle artist, painting \rangle$ as per the types of the head and tail entities.

3.5.2 Vector representations for relations

For representing the semantics of r in terms of the associated entities, we leverage pre-trained KG embeddings. As proposed in previous work [84], we derive a representation for the relation from $\boldsymbol{e_h}$ and $\boldsymbol{e_t}$ vectors corresponding to every triple in $\mathcal{G}_r$. In this way, for every relation r, a set of vectors $\boldsymbol{\Delta_r}$ is obtained from the KG embeddings, in addition to the actual $\mathbf{r}$ vector that the embedding already provides. These $\boldsymbol{\Delta_r}$ vectors are then mapped to the corresponding type pairs (according to the types of the underlying entities). With this, each unique type pair is, in turn, mapped to and represented by a subset of $\boldsymbol{\Delta_r}$ vectors. The $\boldsymbol{\Delta_r}$ vectors encode the combined information conveyed by the head and tail entity types and represent the relationship between the entities, thus encapsulating the latent semantics of the relations in different triples. The $\boldsymbol{\Delta_r}$ vectors serve as the data points for the clustering (with the associated type pairs being their labels).

Relation semantics. While it is believed that KG embeddings are able to capture relation similarity in the embedding space, i.e., relations having similar semantics occur

[2]We denote $\mathcal{P}_r$ as $\mathcal{P}$ when the relation r is clear from the context.

close together in the vector space [40, 85], we found that relations having multiple semantics (based on the context of their entities) are, in fact, not represented well in the vector space. For polysemous relations, the vectors obtained for a single relation (from the different facts that it appears in) form separate clusters in the vector space that do not overlap with the actual relation vector $\mathbf{r}$ obtained from the embeddings. This happens due to the fact that multiple entity pairs connected by the same relation are semantically different from one another. Figure 3.2 shows examples from the NELL and Yago datasets where this behaviour of the embedding vectors for relations is clearly visible. We leverage this semantically-aware behaviour of the embedding vectors to determine meaningful clusters of $\boldsymbol{\Delta_r}$ vectors that represent the distinct latent semantics exhibited by different entity type pairs connected by the same relation, as described next.

3.5.3 Clustering for fine-grained semantics

For each relation r, the total number of unique type pairs $\mathcal{L} = |\mathcal{P}|$ is theoretically the maximum number of possible semantic sub-relations or clusters that could be obtained for r. This will create a different sub-relation for every different type pair. However, in practice, it is rare that all the type pairs would have completely different semantics. For example, the *created* relation in Yago has type pairs $\langle artist, painting \rangle$ and $\langle artist, music \rangle$ that have the same head entity type, while the type pair $\langle organization, software \rangle$ conveys quite a different meaning. While a single relation is not sufficient to be representative of the semantics of all triples that it appears in, at the same time, a naive assignment of sub-relations pertaining to all unique type pairs would also be inefficient and lead to a large number of unnecessary sub-relations.

The *FineGReS* method aims to find an optimal number and composition of clusters $\mathcal{C}_{opt}$ for the type pairs that can convey distinct semantics of the relations based on the data, by combining similar type pairs while separating the dissimilar ones. Each of the clusters having one or more than one semantically similar type pairs represents a potential sub-relation. In order to obtain this configuration, various compositions of the clusters need to be analysed for optimality. For this, clustering is performed in an iterative manner with a predefined number of clusters and combinations of type pairs within each cluster for the iterations. Since it is not feasible or practical to consider an exhaustive number of possible clusters, *FineGReS* leverages the *semantic similarity of type pairs* to narrow down the search space for obtaining the optimal clusters. First, the vector representations for the types are derived. Subsequently, the similarity scores between all combinations of the unique type pairs (t_{h_i}, t_{t_i}), (t_{h_j}, t_{t_j}) are obtained by calculating the similarity scores between the vectors corresponding to the head entity types t_{h_i} and t_{h_j} as well as the tail entity types t_{t_i} and t_{t_j} and then taking their mean value.

Iterative clustering. The iterative clustering begins with $\mathcal{L}$ clusters, with each cluster corresponding to one type pair for the relation in the first iteration. At this point, the cluster labels for the data points ($\boldsymbol{\Delta_r}$ vectors) are denoted by individual type pairs directly and serve as the 'ground truth' for evaluation. Next, the similarity scores of all the type pair combinations are calculated, and the two type pairs that are most similar are considered as candidate pairs to be merged together and placed in a single cluster for the second iteration. To generate the cluster labels, the data points corresponding to the candidate type pairs are assigned the same distinct label (that could be generated e.g. by

combining the individual label names). The number of clusters is given as $\mathcal{L}$ - 1 during the second iteration of clustering, and the cluster labels consist of $\mathcal{L}$ - 2 original type pairs and the one merged type pair. If two combinations of type pairs have the same similarity score in any iteration, ties are broken arbitrarily. This process of selecting the most similar type pairs as candidates for merging in the next iteration to reduce the number of clusters is repeated until all type pairs have been gradually merged back together in a single cluster. In some iterations, the most similar type pairs could be already in the same cluster, so the next most similar pairs are considered until candidates to merge are found. This ensures that the number of clusters always shrinks in subsequent iterations until eventually all clusters are merged back and the algorithm converges. At each iteration, the quality of the clusters is calculated (as detailed in 3.6.2) and this is regarded as the function $\mathcal{F}(\mathcal{C}_{opt})$. The results from the iteration having the maximum value of this function is chosen as the optimal configuration of clusters $\mathcal{C}_{opt}$. The complexity of this algorithm for a relation is proportional to the number of unique type pairs in the dataset and in practice, the run time of iterative clustering process ranges between a few seconds to a few minutes per relation. It is to be noted that while this approach discovers the sub-relations, their labeling is a separate task on its own. In this work, we simply use the type pairs to derive representative labels, e.g. a sub-relation of *created* that connects *company* with *computer_game* could be named as *created-company-computer_game* and so on. However, a proper naming scheme for these relations is concerned with the task of ontology design and is out of the scope of the current work.

3.6 Experiments

We evaluate the effectiveness of our proposed method by performing a series of experiments with several feature-based baselines and a non-embedding baseline approach, as well as variations of the *FineGReS* technique with different embedding models and clustering techniques. In Section 3.6.2 we evaluate if the *FineGReS* approach finds meaningful and useful fine-grained relation semantics as compared to baselines. Section 3.6.3 explores whether the sub-relations obtained from *FineGReS* really reflect what the users need in terms of semantics. Finally, we perform an analysis of the benefits of fine-grained relation semantics for a KG-based application such as entity classification in Section 3.6.4. The experiments are also supported by a qualitative analysis and detailed discussion of the results.

3.6.1 Experimental setup

Datasets. We prepared datasets derived from Yago3 and NELL-995 knowledge graphs for the experiments. For the Yago3 dataset, the entity types (concepts) of all entities were extracted from the accompanying ontology and ranked in terms of frequency. Yago ontology is composed of concepts that are derived from Wordnet as well as from Wikipedia categories (Wikicat). Since the Wikicat concepts are often fine-grained sub-classes of Wordnet concepts, we only consider Wordnet concepts for obtaining non-overlapping clean set of concepts. We considered the top 53 frequent concepts for creating our dataset as the frequencies dropped considerably thereafter. The accounted concepts each had atleast 10,000 entities associated with them. Thereafter, we extracted the facts triples

Table 3.2: Performance of *FineGReS* clusters in comparison with feature-based baselines *(best values in bold, best for a model underlined)*

		TransE						DistMult					
		KMC		HAC		OPC		KMC		HAC		OPC	
		Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
Yago	*subject*	.359	.258	.358	.242	.0013	.0001	.426	.260	.434	.258	.0007	.00003
	object	.213	.167	.187	.112	.0009	0	.248	.136	.339	.167	.0007	.00007
	pair	.153	.095	.162	.064	.0012	**.0002**	.140	.266	.058	.095	.0007	.00006
	$FineGreS_{entity}$	.472	.337	**.527**	**.339**	**.0014**	**.0002**	.519	.321	.532	.337	**.0008**	**.00008**
	$FineGreS_{concept}$	**.537**	**.357**	.525	.329	.0011	**.0002**	**.597**	**.376**	**.582**	**.347**	**.0008**	**.00008**
NELL	*subject*	.217	.125	.256	.159	.006	.0026	.281	.155	.255	.151	.004	.0009
	object	.302	.209	.286	.176	.006	.0017	.291	.204	.323	.204	.005	.0006
	pair	.128	.083	.132	.070	.005	.0033	.089	.051	.137	.079	.005	.0013
	$FineGreS_{entity}$	.345	.178	.467	.210	.007	.0034	**.686**	.194	.454	.207	**.006**	**.0014**
	$FineGreS_{concept}$	**.576**	**.379**	**.711**	**.434**	**.008**	**.0039**	.376	**.387**	**.719**	**.431**	**.006**	**.0014**

from Yago3 that were comprised of subject(head) and object(tail) entities associated with the chosen concepts. This resulted in a set of 1,492,078 triples, which were augmented with the corresponding types of entities. The final dataset consists of 31 relations and 917,325 unique entities. Note that only the data points from the relations having multiple type pairs (after filtering out the ones having too few triples to avoid errors from incorrect entity type mapping) associated with them were considered for clustering.

A similar process was followed for the NELL-995 dataset. In this dataset, the type information is embedded with the entities and thus could be directly extracted from the data triples. Similar to the above heuristics, the types of the entities were restricted to the most frequent types (top 41) found in the dataset (with the less frequent types being replaced by their more frequent supertypes when found in the NELL ontology[3]). The numerical entities were removed from the dataset since they did not have an associated type. The final dataset consists of a total of 200 relations and 75,492 entities along with their corresponding types, and 154,213 triples in total.

Finding Type Similarity. The process of iterative clustering is guided by the semantic similarity of the different type pairs for a given relation and therefore, obtaining the representations for the entity types is an important step in the *FineGReS* method. Here, we describe two different strategies to derive these representations — *concept-based embeddings* and *entity-based embeddings*.

Concept-based type representations— In order to directly obtain vector representations of the entity types, we use the pre-trained ConVec embeddings [44] that are publicly available.[4] These 300-dimensional embeddings were obtained by training over a dataset of 1.5 million words including the Wikipedia *concepts* and thus represent the semantics for the entity types quite well. While the ConVec embeddings work well in most cases, sometimes the entity types are multi-word phrases, especially in the case of NELL dataset. In addition, the NELL ontology is quite large with a much wider vocabulary due to the continuous learning paradigm of NELL. As such, in order to obtain the vector representations that are not found in ConVec and to calculate the similarity scores between the entity types (including both words and phrases), we leveraged the pre-trained *Sentence-BERT* [135] models from the HuggingFace library [170].

[3]Available at - http://rtw.ml.cmu.edu/rtw/resources
[4]https://github.com/ehsansherkat/ConVec

Entity-based type representations — The entities associated with the types can also provide meaningful representations for the entity types. For each type, we first obtain the vectors for the corresponding entities from Wikipedia2Vec tool [175]. Since the Wikipedia2Vec embeddings were derived from the mentions of the entities on the entire Wikipedia corpus, they effectively encapsulate the textual semantics of the entities. The vector representation for the entity type was then obtained by taking the average of the entity embedding vectors. Once the type vectors were obtained from either of the above strategies, the *cosine* similarity measure was used for calculating the similarity matrix between the entity types pairs.[5]

Knowledge graph embeddings. We perform our experiments on the following widely used KG embedding models — *TransE* [20] and *DistMult* [176]. These models are chosen to serve as prominent examples of embeddings using translation distance and semantic matching techniques respectively. We use the model implementations from the LibKGE library [22] for Yago3-10 dataset and from the OpenKE library [64] for NELL-995 dataset.

Clustering techniques. Several different clustering algorithms were employed to obtain the clusters in the vector space — KMeans clustering (KMC), Optics (OPC) and Hierarchical Agglomerative clustering (HAC).

Baselines. To the best of our knowledge, there is no existing research that has leveraged knowledge graph embeddings to discover fine-grained semantics of relations in large KGs. As such, we establish several baselines in this work for analysis and comparison of our proposed approach as well as future works.

Feature-based — To derive sub-relations from a polysemous relation in the KG, several simplistic configurations were explored. The semantics can be driven by the entity types of solely subject or object entities. The different type pairs can also be a criteria for new sub-relations. Hence, we define the baselines as —
pair - Sub-relations obtained on the basis of every unique type pair that is associated with a relation, this setting corresponds to the maximum number of sub-relations.
subject - Sub-relations created by grouping the type pairs by subject entity types i.e. each sub-relation represents all type pairs associated with a common subject type and different object types.
object - Similar to *subject*, but grouping instead by the object entity types.

Non-embedding baseline — Few previous works have discussed the ambiguity in the meanings of relation phrases present in Open IE triples [53, 110]. Particularly, in Min et al. [110] the authors propose *'Type A'* relations where the same relation phrase is associated with different types of subject and object entities, hence denoting different semantics. Such polysemous relation phrases are indeed identified as distinct relations through a variant of the Hierarchical Agglomerative Clustering(HAC) technique. Note that this approach heavily relies on the textual context of the entities and relations which is missing in KG triples. Nevertheless, as this is the closest related approach to our work, we consider it as a non-embedding baseline that is purely text-driven. To best

[5]We also tried the *euclidean* similarity measure and it shows very similar results. For the rest of the paper, we only refer to results from the *cosine* similarity scores.

Table 3.3: Performance of *FineGReS* compared to non-embedding baseline

	Yago		NELL	
	Micro	Macro	Micro	Macro
Baseline	.439	.275	.442	.261
TransE	.537	.357	.711	**.434**
DistMult	**.597**	**.376**	**.719**	.431

Table 3.4: Examples of *FineGReS* sub-relations

Dataset - Relation (Setting)	Count	*FineGReS* Sub-Relations
Yago - *owns* (*TransE*-HAC)	3	$\{\langle company, airport\rangle\ \langle organization, airport\rangle\}$, $\{\langle sovereign, building\rangle\}$, $\{\langle company, club\rangle\ \langle company, company\rangle, \langle country, club\rangle\}$
Yago - *created* (*TransE*-OPC)	4	$\{\langle artist, medium\rangle\ \langle officeholder, movie\rangle\}$, $\{\langle writer, fictional\_character\rangle\}$, $\{\langle writer, movie\rangle\ \langle writer, television\rangle\ \langle artist, movie\rangle\ \langle artist, computer\_game\rangle\ \langle player, movie\rangle\}$, $\{\langle company, computer\_game\rangle\}$
NELL-*agentCompetesWith* (*TransE*-KMC)	5	$\{\langle company, person\rangle\ \langle website, person\rangle\ \langle person, person\rangle, \langle sportsteam, sportsteam\rangle\}$, $\{\langle person, company\rangle, \langle person, website\rangle\}$ $\{\langle animal, animal\rangle, \langle bird, animal\rangle\}$, $\{\langle bank, bank\rangle\}$, $\{\langle mammal, politicsissue\rangle\}$
NELL-*subpartOfOrganization* (*DistMult*-KMC)	8	$\{\langle sportsteam, sportsteam\rangle\ \langle stateorprovince, sportsteam\rangle\ \langle university, sportsteam\rangle\ \langle city, sportsteam\rangle\ \}$, $\{\langle organization, organization\rangle\}$, $\{\langle televisionstation, city\rangle\}$, $\{\langle company, company\rangle\ \langle televisionstation, company\rangle\}$, $\{\langle sportsteam, sportsleague\rangle\}$, $\{\langle televisionstation, website\rangle\}$ $\{\langle televisionstation, televisionnetwork\rangle\}$, $\{\langle bank, bank\rangle\}$

implement this baseline approach from Min et al. [110], the entity similarity was derived from text-driven entity embeddings [175] instead of the KG embedding models (as done in our approach). These text-driven embeddings encapsulate the textual context as well as sentence-level lexical patterns available in Wikipedia texts via word-based skip gram and anchor context models. An entity similarity matrix (corresponding to the entity similarity graph in [175]) was thus constructed from these entity embeddings and the clustering was performed based on the pairwise similarity values from this matrix to obtain relations with distinct semantics.

3.6.2 Evaluation of *FineGReS* relation semantics

Following previous works related to relation phrase clustering [53, 159], we employ micro and macro metrics to evaluate the quality of the clustering in terms of precision, recall and F1. Table 3.2 reports the weighted (as per the number of data points) F1 metrics for the datasets obtained by the feature-based baselines and the *FineGReS* method in the different settings of KG embeddings and clustering techniques. Note that $FineGreS_{concept}$ and $FineGreS_{entity}$ correspond to the different variations of the *FineGReS* approach in

terms of obtaining type representations (refer to Section 3.6.1). Additionally, we also compared the best performing setting of the *FineGReS* method in case of *DistMult* and *TransE* models to the non-embedding baseline and present the results in Table 3.3.

Observations. It can be seen that in all settings the clusters obtained by the proposed *FineGReS* method outperform the baselines in terms of both micro and macro metrics on Yago and NELL datasets. Clustering with *kmeans* and *hierarchical agglomerative* techniques show better results in comparison with *optics* which is a density-based clustering technique.[6] Overall, the results provide strong evidence in support of the efficacy of our method for finding optimal configurations of clusters for the relations, from which sub-relations with well-defined semantics can be derived. Furthermore, it is observed that $FineGreS_{concept}$ performs better than $FineGreS_{entity}$ in majority of the cases for both Yago and NELL datasets. We conjecture this is due to the fact that the semantics of the entity types directly obtained from ConVec vectors (see Section 3.6.1) are more precise, whereas, the semantics derived from the vectors of the entities associated with the types are prone to noise and errors. Therefore, the type similarities would be more semantically aligned in the case of $FineGreS_{concept}$, thereby leading to superior performance of the method. Another important insight from the results is that while it is indeed favorable to replace a polysemous relation with multiple sub-relations, it is certainly not a trivial task to obtain these sub-relations by simply defining their semantics in terms of unique type pairs. The *pair* baseline that corresponds to such sub-relations can be seen to score consistently lower in all settings. The *subject* and *object* baselines fair better in this regard, though the proposed *FineGReS* approach is clearly the most optimal. From Table 3.3, it can be seen that the non-embedding baseline was outperformed by *FineGReS* with the exception of *TransE* giving better result for NELL dataset. As mentioned, this baseline benefits from textual context which is lacking for our approach and therefore, a fare comparison is hard to perform. Still, the results indicate that KG embeddings are able to represent the semantics of the relations and identify fine-grained relation semantics in large KGs, even in the absence of additional cues or background knowledge.

Qualitative results. Table 3.4 shows a few representative examples of the sub-relations, along with their count, obtained by *FineGReS* in different settings for Yago and NELL. It can be seen that semantically different entity type pairs have been clearly separated out as distinct sub-relations, e.g. the $\langle sovereign, building \rangle$ pair for *owns* relation where *sovereign* is semantically distant from other types or *agentCompetesWith* where $\langle bank, bank \rangle$ is a separate sub-relation. Other sub-relations have multiple type pairs associated with them based on their semantic proximity. Note that in a few cases, the optimal configuration for a relation could indeed correspond to the *pair* or *subject*/*object* baseline depending on the associated type pairs. The *FineGReS* method is able to automatically determine this optimal configuration of the sub-relations for each relation relying solely on the triples in the KG dataset and the associated entity type information.

3.6.3 Manual evaluation with Yago

In order to estimate the usefulness of the fine-grained sub-relations obtained from *FineGReS*, we performed a limited manual evaluation and analysis on the Yago dataset.

[6] This was also observed by previous work in the context of KG embeddings [76].

Three annotators were given the different type pairs associated with 15 candidate relations in Yago having more than two distinct type pairs) and asked to independently identify any potential sub-relation clusters by assigning labels to the type pairs. The relations for which atleast two annotators agreed on the label assignments were taken into consideration as the true values. These were then compared with the labels obtained from the top k best performing *FineGReS* settings from Table 3.2 for each relation and the *Hits@k* metric was calculated. Essentially, we measure how often the sub-relations identified by human annotators for each relation were also found by the proposed technique among the top k performers. The values of *Hits@1* and *Hits@3* were found to be *0.33* and *0.66* respectively, indicating that the sub-relations discovered by *FineGReS* indeed resembled the semantics that the human annotators had identified to be useful for many of the relations. The manual evaluation proved to be challenging due to the subjective nature of this task, where humans could not always identify the precise semantics of potential sub-relations in the absence of additional context. Embeddings derived from relational learning models are superior in this regard as they are able to encapsulate the latent semantics of the KG relations, hence they are well-suited to the task of fine-grained relation discovery.

Discussion. A closer inspection of the sub-relations obtained from *FineGReS* revealed further interesting insights. First of all, due to the data-driven nature of the proposed approach, where only KG triples serve as data points, the results are worse for relations with a smaller representation in terms of the number of triples in the dataset, as compared to the relations with a larger number of triples. This is quite expected as the clustering algorithms fail to identify good clusters in the vector space when there are very few data points available. Along the same lines, it is important to point out that if a type pair has few data points but it is semantically distinct from the others, it is still identified as a separate cluster, e.g. in the case of $\langle bank, bank \rangle$ type pair for the *agentCompetesWith* relation in NELL (Table 3.4). This way, the semantics of the type pairs for a relation play a decisive role in the clustering, rather than the number of data points (i.e. frequency with which the relation connects the different type pairs). Furthermore, it was seen that the proposed approach is rather too aggressive for some relations, where there might be different entity types associated with the relation but they still represent the same semantic. Especially in Yago, while relations such as *lives_In* and *married_To* convey a clear meaning, due to the hierarchical ontology structure, the entities associated with these relations form different type pairs such as *(officeholder, country)* and *(scientist, site)* in the case of *lives_In* relation. Therefore, the *FineGReS* approach discovers separate sub-relations for these relations despite the same semantic. In the same dataset there is another relation *participatedIn* where the types *officeholder* and *scientist* play distinctly different roles and indeed belong to separate sub-relations. As the mapping of the entities to their types is performed consistently for all the triples in the dataset, and not on a per-relation basis, the proposed method cannot distinguish the cases where the entities such as *officeholder* and *scientist* should be abstracted to represent the *person* type, as a human annotator would understand. Related to this discussion, it is noteworthy that the assignment of the types to the entities can be differently performed in the underlying dataset depending on the required level of granularity. The proposed method can, in principle, work at different levels of fine-grained semantics as dictated by richness of type assignment of the entities in the hierarchy of the ontology or as desired by a downstream

Table 3.5: Performance comparison for entity classification task for Yago and NELL (*R refers to original relations, Base refers to the non-embedding baseline*)

							FineGReS	
		R	*pair*	*subject*	*object*	*Base*	*TransE*	*DistMult*
Yago	P	.893	.916	.906	.918	**.926**	.923	**.928**
	R	.908	.925	.921	.935	.938	**.941**	**.942**
	F1	.894	.914	.909	.924	.926	**.931**	**.931**
NELL	P	.643	.692	.696	.665	.567	**.705**	**.713**
	R	.689	.727	.729	.703	.645	**.736**	**.747**
	F1	.650	.701	.713	.683	.584	**.715**	**.726**

application.

3.6.4 Entity classification use case

In order to empirically evaluate the *FineGReS* method in terms of the usefulness of the derived sub-relations, we consider the popular use case of entity classification which is an important task for KG completion [116]. It is modeled as a supervised multi-label classification task, where the entities are assigned to their respective types. Previous works have performed type prediction for entities in KGs based on statistical features [127], textual information [89] as well as embeddings [17]. Taking cue from the same, we design a simple architecture with a CNN classifier [181] for the multi-label classification task which can jointly classify both the entities in a given triple to their respective types.[7] The model consists of a convolutional layer with feature detector, and ReLu activation, this is followed by a max pooling layer and dropout layer to reduce over-fitting. The output is passed through a fully connected layer with softmax activation to obtain the probability of the different classes for being the predicted type for the entities. The Adam optimizer was used with the learning rate set to 0.0001. The experiments were run on a server with Intel X86 CPU and using a single NVIDIA GTX1080 GPU with 11GB RAM. The dataset for the classification task was obtained by replacing the original polysemous relations in the KG dataset with their corresponding fine-grained sub-relations in the relevant triples, obtained from the best performing setting of the *FineGReS* method as well as from the baseline techniques described in Section 3.6.1. The performance of entity classification measured in terms of weighted precision, recall and F1 scores (averaged over 10 runs) is shown in Table 3.5 for Yago and NELL. The main objective is to measure the improvement in performance when the relations in the triples of the KG are dictated by well-defined, fine-grained semantics as opposed to ambiguous semantics. The results confirm that entity classification task indeed sees an improvement when the underlying dataset is comprised of relations with fine-grained semantics obtained from *FineGReS* method, in comparison to the original polysemous relations (denoted as *R* in the tables), as well as the relations obtained from other feature-based and non-embedding baselines. In particular, the gains seen over the *pair* setting are indicative of the superiority of the *FineGReS* method in terms of not merely finding *any* set of sub-relations but finding the

[7]The setup is intentionally simple in these experiments so as to draw attention to the effect on performance from different configurations of relations and pseudo sub-relations in the KG dataset. It could arguably be replaced by any state-of-the-art technique.

optimal configuration of the sub-relations that best represent fine-grained semantics for the relations.

3.7 Summary

In this chapter, we have presented the task of fine-grained relation discovery for knowledge graph refinement, which is an important problem that has not been fully explored. The proposed scalable and data-driven method *FineGReS* automatically determines an optimal configuration for deriving fine-grained sub-relations by taking advantage of the latent relation semantics represented by KG embedding models. This technique does not rely on additional background knowledge and thus it can be employed for arbitrarily large and heterogeneous KGs. We established several baselines and conducted extensive empirical evaluation that demonstrated the difficulty of this task and the efficacy of the proposed method for learning fine-grained relation semantics. The improved performance for the task of entity classification strongly indicates the promise of this approach. Since the method relies on the type information of the entities, *FineGReS* can currently be applied only to the KGs accompanied by their ontologies. It would be interesting to extend the approach to derive relation semantics from other sources, such as text.

While knowledge graph embeddings have been successfully leveraged for finding fine-grained relation semantics in the proposed *FineGReS* approach, limitations regarding the semantic representation of KG entities were also discovered that warrant closer inspection. The next chapter presents a discussion and systematic evaluation of these shortcomings in KG embeddings.

Chapter 4

Semantic Representation in Knowledge Graph Embeddings

"I would rather have questions that can't be answered than answers that can't be questioned."
— Richard Feynman

In the last chapter, we have seen how knowledge graph embeddings can be used for deriving the semantics of the relations in the underlying KG datasets. In this chapter, we take a closer look at the embeddings models themselves, particularly, how well the latent KG semantics are actually encapsulated by the embeddings. While embeddings have indeed been used for various semantic tasks in the past, it is important to perform a critical and quantifiable analysis of the semantic representations in popular embedding models to understand their scope and limitations. Such an analysis has been missing from previous works and is the main focus of this chapter. Our systematic analysis shows that though it seems intuitive to leverage KG embeddings for semantic interpretability (just like word embeddings successfully have been), this is not always the case. The performance of embeddings is, in fact, limited in reality and heavily dependent on the dataset characteristics.

The chapter is primarily based on the work published in Jain et al. [76]. It is structured as follows — Section 4.1 discusses the semantics of knowledge graph embeddings and motivates the need for a critical analysis. Section 4.2 provides an overview of the related works that have leveraged embedding model for various tasks as well as those that have analysed different aspects of these models. Section 4.3 explains the design of our experimental analysis in terms of dataset preparation and techniques. The experimental setup and the results are presented in Section 4.4. Then, Section 4.5 contains a detailed analysis of the shortcomings that were observed and the lessons learnt from this analysis. Finally, we conclude this chapter in Section 4.6 with a summary of the major insights and points for improvement.

4.1 Knowledge Graph Embeddings and Semantics

As previously mentioned, the fundamental idea behind latent embedding models or knowledge graph embedding models (used interchangeably throughout this chapter) is

the representation of entities and relations by low-dimensional dense vectors that can capture the semantics and interactions within the knowledge graph. In the past decade, latent embedding models have garnered considerable attention due to their success on the link prediction task. Following the introduction of the *TransE* embeddings by Bordes et al. in 2013 [20], a flurry of different models have been proposed in the recent years, as summarized by Wang et al. [165]. Many different models have been proposed that achieve state of the art performance for the task of triple completion in knowledge graphs, on which these models have been trained and evaluated.

It is important to note that while these models were originally proposed for knowledge graph completion, the intense popularity and frequency of novel ideas towards better KG embedding models has encouraged the research community to exploit these embeddings for other tasks as well. Since the basic premise of KG embeddings is centered around the semantic relationships between various entities, there is a widespread notion that embeddings must be able to capture the semantics and features of KG components very well. As such, embeddings have been used for many similarity-based tasks including entity similarity [146], and conceptual clustering [51, 52, 163]. Moreover, several previous works have attempted to leverage KG embeddings for performing reasoning with rules [69, 176, 180].

While the results look promising, none of these previous works have performed a detailed analysis of the benefits of the embeddings across different datasets as well as across different entities within a single dataset. In some cases, a measurement of the consistency and scalability of the proposed embedding-based approach for different real-world datasets is largely lacking. The oversight of the limitations of KG embeddings and emphasis on the success for the simpler cases might prove misleading to research community. There is a need for addressing the above issue by studying the characteristics of the latent vectors obtained from several KG embedding models and *quantitatively* measuring their ability for semantic representation. With the aid of a systematic evaluation, it needs to be ascertained that when embeddings are trained on the KG entities to learn their semantic features, whether this learning is uniform or the quality of semantic representation varies largely across different entities within the dataset. Evidence of non-uniform quality would raise doubts about the applicability of KG embeddings not only for semantic reasoning, but also for triple completion and link prediction.

4.2 Related Work

KG embeddings have been used for a variety of applications over the years. We provide an overview of the related works that follow embeddings-based approach and discuss them in the context of semantics in the embeddings. We also mention previous works that have critically analysed embedding methods.

Entity typing. Finding missing type information for entities in KGs has been a long standing problem. Early techniques usually relied on probabilistic methods for predicting the class membership of entities based on their properties [127]. More recently, KG embeddings have been used together with classification algorithms. As an example, Nickel et al. use RESCAL to predict new type information in a small Yago dataset and show good results on high-level classes such as *persons*, *locations* and *movies* [120]. Moon

et al. propose a new embedding technique for performing entity typing [114]. In the example illustration for clustering shown in this paper, it can already be observed that the embedding technique seems to be problematic at distinguishing fine-granular classes such as *author* and *actor*. To a certain degree, their results show that entity typing with KG embeddings is far from being an ideal solution. More recently, an improved embedding technique for entity typing has been proposed [186]. Similar to us, the authors perform an evaluation of embeddings on Freebase and Yago for the entity typing task. While the results already reveal some problems when using entity embeddings for typing, a larger analysis is not performed. In contrast, our work undertakes a detailed analysis of the limits of entity typing when using KG embeddings and shows how classical techniques (e.g. *SDType* [127]) are often superior.

Entity clustering. Besides link prediction, entity clustering is another popular application of KG embeddings. Gad-elrab et al. [51], perform a limited analysis of several clustering algorithms on fine-grained classes. In a related work, the authors leverage rules and embeddings in conjunction to derive explainable clusters from the dataset [52]. However, the results have been shown to work well only for relatively easy relational datasets having well-defined relations between the entities and for small, targeted subsets of Yago. A scalability analysis of these techniques for actual knowledge graphs where their applicability would be most useful is missing. Another related work is presented by Jain et al. [82] where the authors incorporate type information of entities to design better embedding models and demonstrate their results on entity clustering. However, clustering results are illustrated only for limited classes such as persons, organizations and locations without any details on the performance across all classes in the dataset.

Another branch of research concerns with using path-based graph embeddings to perform node classification and clustering tasks [74]. Generally, these techniques aim at creating node (or entity) embeddings using longer paths, instead of relying only on triples like common KG embeddings. However, these techniques are usually evaluated on datasets that do not share the characteristics of knowledge graphs in terms of having fine-grained entity types. Still, as a representative for path-based embeddings, we also evaluate RDF2Vec [137] in this work.

Other applications. Besides knowledge graph completion, KG embeddings have been employed in a number of other settings. Similar to previous tasks, it is crucial that KG semantics are captured properly for embeddings to scale well for arbitrary real-world datasets. Embedding approaches have been explored in the context of rule mining on KGs by many previous works with seemingly good results. Existing techniques have either attempted to mine rules directly from the embeddings [176], or use embeddings to support rule mining for confidence computation [69, 180] such that rules of higher quality can be mined. The latter works have not studied or quantified the benefits of embeddings on their work or explored which entities are positively impacted by them.

Furthermore, embeddings are often used to measure the semantic similarity of the entities and relations to perform data integration via entity or relation alignments [30, 85]. An overview of several entity alignment techniques which are based on embeddings is presented in [146]. In our work, embeddings based approaches are compared to classical non-embedding approaches showing no real advantages. This result may already imply that entity semantics is not represented properly in embeddings.

Criticism of KG embedding models. For several years, a large variety of knowledge graph embeddings has been developed to perform link prediction to cope with incomplete information in KG. However, recent works have also put the efficacy of KG embeddings techniques under scrutiny [4, 139, 140]. A re-evaluation of knowledge graph embedding methods shows several quality problems in the evaluation of KG embedding models as well as the carefully curated benchmark datasets that have been universally used for performance comparison [4]. Akrami et al. demonstrate that existing datasets show several redundancies and cross-product relations. Redundancies in the datasets lead to heavy data leakage thereby making them unrealistically simple in contrast to real-world KG. Furthermore, cross-product relations, connecting all entities to all other entities are frequently used. The authors point out that predictions for these relations is trivial and leads to overestimating the performance of embedding techniques. They show that cleaning the datasets from these defects significantly reduces the link prediction quality of KG embeddings. In another study, the performance gains claimed by newer and more complex models in comparison with the first KG embedding models has also been questioned [140].

The above mentioned works have evaluated and criticized the KG embedding models primarily in terms of their performance on the link prediction task. In this chapter, we focus instead on the utility of the KG embeddings for providing semantic interpretations (or rather, the lack thereof). Our work extensively analyses the problems of current embedding models in terms of their semantic representation, casting doubt on their overall usability in complex real-world KG settings.

4.3 Analysis of the Semantics in Embeddings

In this section, we explain our approach to perform a systematic evaluation of the embeddings for checking their semantic soundness. We also elaborate on the design of our experiments based on popular benchmark datasets.

4.3.1 Categorization of entities

KG embeddings are trained to capture the structural information of the underlying dataset. Ideally, if latent embeddings were able to embody all the latent features of entities, then entities with similar features would be similar in the vector space as well. That is, entities belonging to a particular *type*, and therefore having similar features would result in similar vectors [163]. Inversely, the embeddings that are close to each other in the vector space would correspond to entities having similar types or features [114]. This implies that it should be possible to identify the entities belonging to a particular type from the KG embeddings. Therefore, in this work we focus on verifying whether the entities can be categorized or assigned to their respective types from their corresponding latent vector representations.

While this is similar to the task of *entity typing* as discussed in Section 4.2, in this work we chose to follow a comparatively straightforward approach to analyse whether the embeddings in high dimensional space can indeed express the similarities between entities belonging to the same class or concept. We perform a systematic investigation with two distinct sets of *classification* and *clustering* experiments for the entity embeddings in the vector space.

Both these methods are suitable for semantic analysis as they can identify salient features of the embeddings, if any. These can be used to assign the correct class label to the entities in the case of classification, and segregate the entities into separate clusters as per their classes in the case of clustering. If latent embeddings are able to capture the connotations of entities, then this should be reflected in the performance of classification and clustering results obtained by using the embedding vectors as representation. The intentional choice of these techniques is also, in part, to their simplicity, which will enable us to lay the focus on the quality of the embeddings instead of the quality of the evaluation technique itself.

Classification. With the aid of the supervised approach of classification, we hope to discover the salient semantic features that the latent embeddings are assumed to have learned and use these features to identify the correct class labels for entities. Since an entity can belong to multiple classes in a KG, this entity typing task is a multi-label classification problem where one or many class/type labels can be assigned to an entity. For our experiments, we employed three different types of classification algorithms which work well for multi-label data. The *Multi Layer Perceptron* (MLP) classifier is a neural-network-based classifier using a simple feed-forward network. We chose the most basic architecture with a single hidden layer with 100 units. As a second classification technique, we chose a *K-Nearest-Neighbour* (KNN) classifier. Lastly, *Random Forest* (RF) classification is used as a decision-tree-based algorithm.

Clustering. Being an unsupervised task, clustering is used for identifying the class membership of entities by assigning them to separate clusters, each cluster ideally representing a class. For our experiments, since the ground truth for class labels of entities is known, we are able to measure the quality of clustering by comparing the actual labels with the predicted class labels. Previous works have attempted to identify conceptual clusters in a vector space by applying simple techniques such as *K-Means* to entity embeddings obtained from KG embedding models [52]. We expand our analysis to multiple clustering techniques to weigh the merits and flaws of the techniques and draw conclusions about the characteristics of the underlying embeddings on which clustering is performed. In our experiments, we leverage *Spectral* clustering, *Optics* clustering as well as *Hierarchical Agglomerative* clustering techniques in addition to the simple *K-Means* technique. While hierarchical clustering is particularly suitable for representing the class hierarchy present in most KG ontologies, *Spectral* clustering has shown promising performance for graph based data. *Optics* is a density-based technique that is suited for identifying clusters in spatial data and fits well to our use case.

It is to be noted that our intention for performing clustering in this work is not to discover new concepts but rather to re-discover the existing concepts that the entities are already associated with. Therefore, we provide the required parameter of the number of expected clusters and calculate cluster quality based on ground truth class labels of the entities under consideration.

4.3.2 Datasets

For the experiments, we have chosen the popular benchmark datasets Yago3-10 and FB15K-237. This allows for our results to be put in the correct context with regard to the

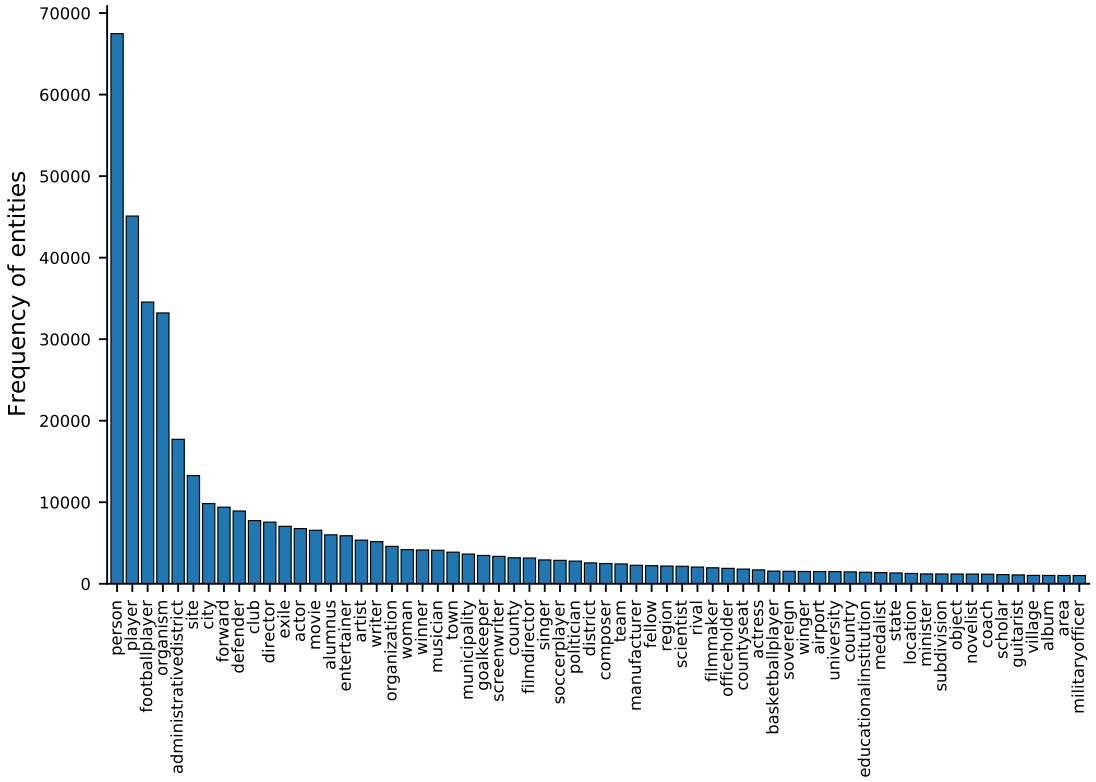


Figure 4.1: Yago3-10 class frequency analysis

numerous other related works that have shown good performance on these datasets [22]. Here, we discuss the main characteristics of these datasets and describe the selection of a suitable subset for the clustering and classification experiments.

Yago3-10. This dataset was created from the Yago3 knowledge graph [106] by filtering out the entities having less than 10 relations. It consists of a total of 1,079,040 triples with 123,181 entities and 37 relations. It is important to note that Yago is a semantic knowledge base associated with a hierarchical ontology that was derived from *Wordnet* taxonomy [109] combined with Wikipedia categories that are often fine-grained and noisy.

In order to explore the differences in semantic representation for entities with varying type granularity, we proceeded to extract entities belonging to classes at different levels of the Yago ontology that resembles a tree-like structure. We limited our analysis to the concepts in Yago that are directly mapped to the *Wordnet* taxonomy to obtain a clean sub-tree of classes that are related to each other. Starting with the main branches of Yago class hierarchy, we chose the classes *person*, *organization*, *body_of_water* and *product*, then progressively explored their sub-trees to design experiments at different levels of the class hierarchy. For this, we manually performed a systematic analysis of the sub-classes of the above four classes and chose the most frequent classes for our experiments. This was a non-trivial task for the Yago3-10 dataset due to the presence of a highly skewed class frequency distribution. As reported previously [65], a large proportion of the entities in this dataset belongs to very few classes, while a long list of classes have very few representative entities. Almost 62% of all the entities belong to the 1% most

Table 4.1: Yago3-10 experiments for different levels

Experiment	Classes
Level-1	person, organization, body_of_water, product
Level-2-organization	institution, musical_organization, party, enterprise, nongovernmental_organization
Level-2-body_of_water	stream, lake, ocean, bay, sea
Level-2-person	artist, politician, scientist, officeholder, writer
Level-3-person-writer	journalist, poet, novelist, scriptwriter, dramatist, essayist, biographer
Level-3-person-artist	painter, sculptor, photographer, illustrator, printmaker
Level-3-person-player	hockey_player, soccer_player, ballplayer, volleyball_player, golfer
Level-3-person-scientist	social_scientist, biologist, physicist, mathematician, chemist, linguist, psychologist, geologist, computer_scientist, research_worker

Table 4.2: FB15K-237 experiments for different levels

Experiment	Classes
Level-1	person, organization, body_of_water, product
Level-2-organization	institution, musical_organization, party, enterprise, nongovernmental_organization
Level-2-person	artist, politician, scientist, officeholder, writer
Level-3-person-writer	journalist, poet, novelist, scriptwriter, dramatist, essayist, biographer

frequent classes in this dataset. The frequency distribution of the classes (having at least 1000 entities) is graphically represented by Fig. 4.1 which shows that the class frequency distribution follows Zipf's law.

Due to the constraint of sparse entities in many cases, for each class, a list of sub-classes having entities above a minimum threshold were explored and used for designing the experiments (sub-classes leading to a high skew were omitted to ensure data balance). This was done for three levels starting with the main Yago classes as stated above. Each experiment contains a set of classes that belong to the same level in the ontology. This is important for a fair comparison of the semantic representation of the classes at different granularity levels of the class hierarchy. Table 4.1 lists all the experiments at different levels along with their classes. For each experiment all the entities belonging to the set of classes in the experiment was compiled from the Yago dataset, then the corresponding embeddings for these entities was extracted from pre-trained KG embeddings models to serve as data for the clustering and classification experiments.

FB15K-237. This second dataset is a subset of the Freebase knowledge graph, frequently used by knowledge graph embedding models. FB15K-237 [151] comprises 272,115 triples with 14,541 entities and 237 relations. It was derived from the FB15k [20] dataset by filtering out redundant and inverse relations. With regard to the domains, it mainly pertains to *persons*, *organizations* and *products* and we aimed to design our experiments with a similar structure. We performed the mapping of Freebase entities to Yago through existing *sameAs* links and chose classes and sub-classes by following the Wordnet taxonomy. The experiments were designed in the same way as described above for the Yago dataset for allowing direct comparisons. The Freebase dataset is significantly smaller than the Yago dataset, such that the number of entities reduces dramatically when considering the classes at level-3. Therefore, we had to limit ourselves to fewer experiments as listed in Table 4.2.

4.3.3 Knowledge graph embeddings

For all the experiments, we obtain the pre-trained embeddings models for the benchmark datasets from the LibKGE library [22] since extensive hyperparameter tuning has already been performed. We used five different embedding techniques (as introduced in 1.3) that are widely popular — TransE, RESCAL, Complex, DistMult and ConvE. Since for Yago3-10 only the Complex embeddings were available, we trained the remaining embeddings ourselves by adapting the parameters that were used for the Freebase dataset[1]. Another popular branch of embedding approaches is based on paths in a knowledge graph, usually showing good results in entity typing tasks as discussed in Section 4.2 [137]. RDF2Vec was trained using paths created by a random walker algorithm which created paths of length 4. Then the model was trained for 50 iterations using pyRDF2Vec library.[2]

4.4 Experiments

In this section we present the results of our experiments for clustering and classification on Yago3-10 and FB15k-237 datasets. Additionally, we draw comparisons with a traditional statistical approach.

4.4.1 Non-embedding baseline

To ensure that the results are not driven solely by the performance of clustering and classification algorithms, we found it important to include a baseline that is unrelated to the embeddings. For this, we leveraged the *SDType* approach as introduced by Paulheim et al. in 2013 [127]. This is a heuristics based technique that simply uses the links between the entities to infer their type. Based on the incoming and outgoing relations associated with a particular entity, the average probability of each type for an entity is calculated. Purely relying on the statistical distributions of the entity links, this method is robust to noisy facts in the dataset and agnostic to existing type information. We rely on this approach to stipulate whether any semantic features are present in the underlying data that can help with the deduction of type information for the entities. If the statistical approach can already leverage the semantic features in data to identify the types for entities, this indicates that unsatisfactory scores for classification or clustering on embeddings must be due to the failure of embedding models to capture these semantic features during training. We report the performance of *SDtype* for our experiments along with the classification results in terms of the best F1 measure obtained (P-R curves are available on github link).

4.4.2 Evaluation metrics

Similar to previous works [52], we measured the Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and the V-measure to estimate the quality of the clusters. With the true and predicted labels as input, ARI measures the similarity of the assignments with values between -1 and 1 (0 stands for random assignment, 1 is the perfect score). NMI measures the agreement of the assignments and V-measure is the harmonic mean of

[1]The training parameters and performance scores are available on github link.
[2]https://github.com/IBCNServices/pyRDF2Vec

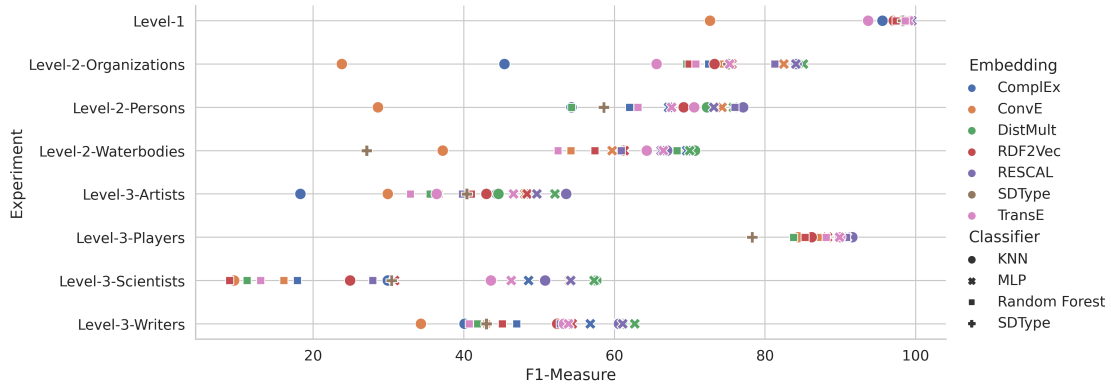


Figure 4.2: F1 measure for Yago3-10 classification experiments (best viewed in color)

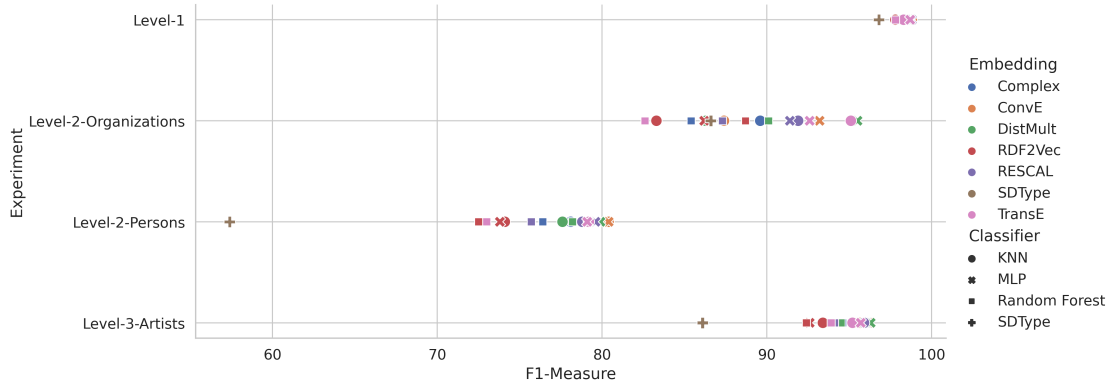


Figure 4.3: F1 measure for FB15K-237 classification experiments

homogeneity and completeness of the clusters. For both, the values lie between 0 and 1, with 1 being a prefect score. For the evaluation of classification experiments, an 80-20 ratio was used to split the dataset (consisting of entity embeddings and class labels) into train and test set. Since the task is a multi-label classification, the weighted average of F1 measures per class (in %) in the test set was used as an evaluation measure.

4.4.3 Classification results

Fig. 4.2 shows the weighted F1 measures for Yago3-10 dataset across all the embedding models (color coded) as well the different classifiers (pattern coded). It can be seen from this figure that all the classifiers perform very well for level-1 experiment (refer to Table 4.1), where the considered classes are coarse-grained and distinct from one another. However, the performance starts degrading once experiments at level-2 are considered and becomes worse for level-3, where the F1 measure drops below 20 for sub-classes of the *scientist* class. This is due to the fact that classes are finer-grained for these experiments, where they all have a common parent class and share certain common features. For instance, different types of *persons*, and further, different types of *artists*, *scientists* etc. would all share common properties of the *person* class (discussed in detail in Section 4.5). Even though the considered classes are conceptually distinct from one another, the classification algorithms find it hard to perform label matching correctly based on embeddings. This behaviour is uniform across all clustering algorithms and

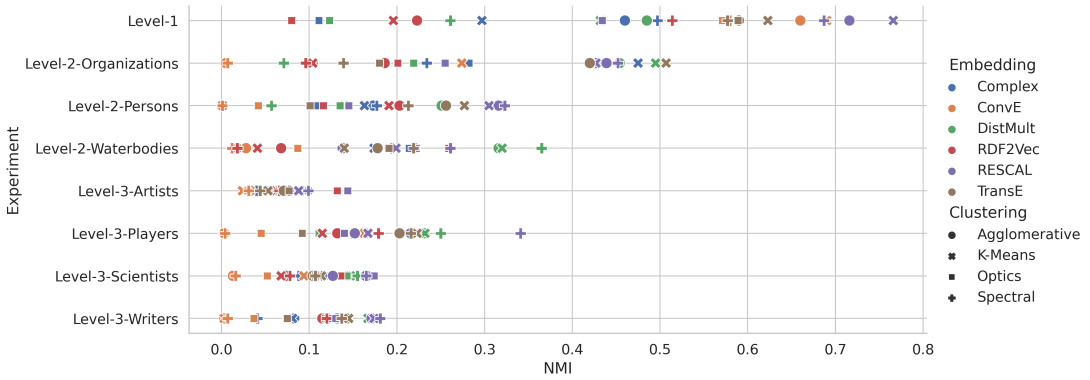


Figure 4.4: NMI measure for YAGO3-10 clustering experiments (best viewed in color)

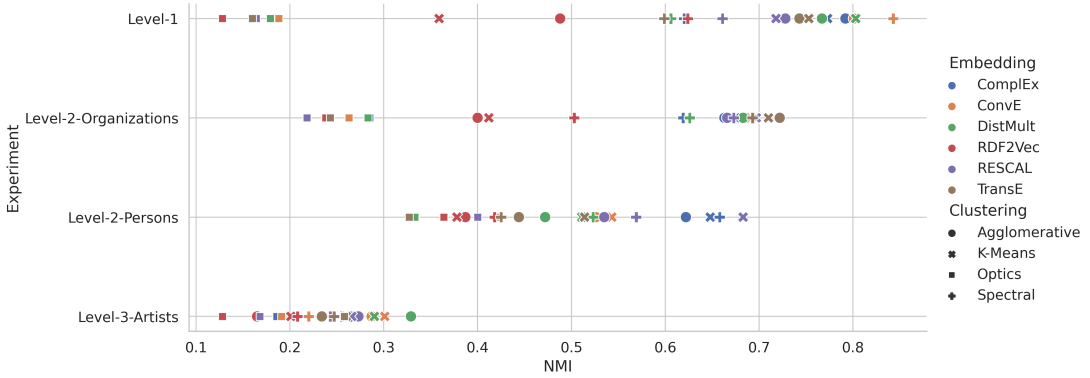


Figure 4.5: NMI measure for FB15K-237 clustering experiments

all embedding models, with no setting performing particularly better or worse. Though fine-grained entity typing is indeed a hard problem, our experiments are designed only for the top three levels of classes. It is indicated by these results that embeddings simply do not possess the necessary semantic features such that classification could identify correct entity types beyond the highly coarse-grained classes.

Similar trends are also seen for the FB15k-237 dataset (Fig. 4.3) where classification performs very well for the level-1 experiment, but gets worse progressively for level-2 and level-3. A few exceptions in this trend are noticed when the dataset is highly skewed towards entities of a particular class, such as *players* in case of Yago and *artists* in case of Freebase. In this case, the performance is improved to some degree as compared to other experiments at the same level. The performance of Freebase is generally better than Yago due to the presence of more relations in the dataset. Overall, the drop in classifier performance with increasing levels indicates a lack of sufficient semantic representation in embeddings for fine-grained entities for both the datasets.

To compare and contrast the performance of the *SDType* baseline approach, the F1 measures for *SDType* are also shown in Fig. 4.2 and Fig. 4.3 (coded with a different color and symbol). Significantly, it can be seen that *SDType* is able to achieve quite competitive results as compared to the embeddings, notably for the level-3 classes. This provides strong evidence for the shortcomings of embeddings for representing fine-grained classes for which even simple statistical approach can already give comparable results.

4.4.4 Clustering results

The results for the clustering experiments are reported in terms of the NMI scores and shown in Fig. 4.4 for the Yago3-10 dataset and Fig. 4.5 for the FB15k-237 dataset. Overall, clustering performs worse than classification, which raises doubts over the expected spatial closeness of similar entities in the vector space. Further, the clustering results also demonstrate a similar pattern to the classification results. The NMI scores are relatively better for level-1 classes but get progressively worse for lower levels[3]. All embedding techniques fair similarly, thus conveying that it is difficult to identify or re-discover even the existing entity types or classes from any of the embeddings with the help of clustering, except for very high-level classes. Considering the different algorithms, *Optics* shows worse clustering scores in many cases. Since *Optics* is a density-based clustering technique, the low quality of clusters again point towards the lack of proper conceptual representation in the embeddings in vector space.

4.5 Analysis

From the experimental results on both supervised and unsupervised tasks, it is clear that KG embeddings are unable to capture the latent features that would be sufficient for a good semantic representation for all entities of a KG. While entities belonging to a small set of high-level *easy* classes are relatively well-represented, the same does not hold true for most of the entities corresponding to other important classes in the dataset. We investigated further to understand the plausible reasons for this shortcoming and discuss our findings here.

Looking beyond the flaws in the training and evaluation process of the KG embedding models (that has been the focus of previous works as discussed in Section 4.2), we studied the characteristics of the underlying KG datasets on which the various embeddings are trained. Knowledge graphs such as Yago and Freebase are comprised of real world entities that frequently belong to more than one semantic type or class e.g. an artist can also be a politician in real life. Since such entities would reflect the characteristics of multiple classes, they are associated with a number of different relations that are neither unique nor indicative of any single class in particular.

To explore this further, we performed an analysis of the relations associated with the different classes that were used in our experiments for the Yago3-10 dataset. For each class, the incoming and outgoing relations associated with all the entities of the class were separately identified. Thereafter, the classes were compared to each other in terms of their relations within the same experiment as well as across experiments at different levels (as listed in Table 4.1). Fig. 4.6 shows a comparison for classes at different levels based on their outgoing relations for a few representative experiments. Here, a slot is shaded depending on the premise that the relation was found for a minimum number of entities of the class. The figure demonstrates that the classes at level-1 have different sets of relations associated with them, i.e. there are few overlapping relations. This is less so for level-2 classes where several relations are found to be common. Finally, at level-3 there are hardly any unique relations that could distinguish one class from another and the relations overlap is quite substantial.

[3]ARI and V-measure show similar trend, full results are available on github link.

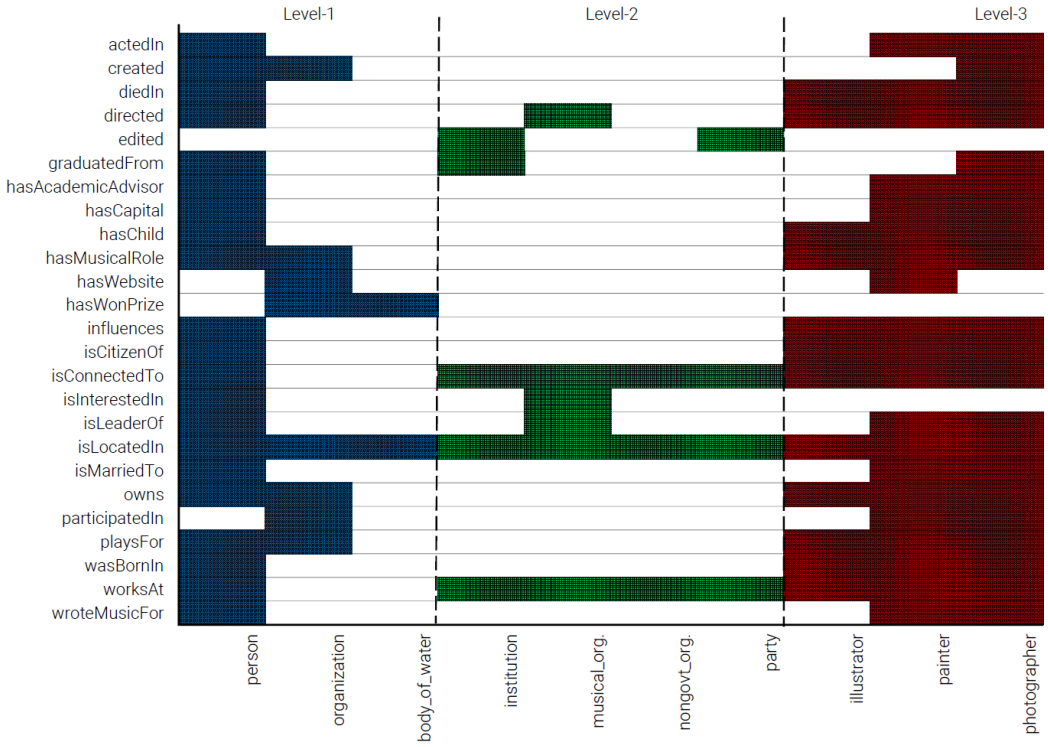


Figure 4.6: Representation of outgoing relations at different levels in Yago

These results stem directly from the characteristics of real-world data where, for instance, all persons have similar properties (e.g. *wasBornIn*, *isCitizenOf*) regardless of their profession. In Yago3-10, any specific relations that could have uniquely identified, e.g. an *artist* from a *politician* seem to be either missing or very sparse. This directly affects the embeddings since they are trained to learn the associations between the different entities of a KG (a heuristics based approach like *SDType* can exploit sparse links much better). The presence of overlapping relations among entities belonging to different semantic types hinders their ability to encapsulate *type-specific* features. In this case, an embedding model can only hope to learn from other entities that are found in the triples of the entities of a particular class, and find patterns and features from those entities. However, recent work has shown that relations in knowledge graphs can be ambiguous in the way they connect different entities [77]. This means that various types of entities might be connected to a particular entity by the same relation. Such generic and noisy links make it even harder for embedding models to derive type-specific features about the entities, thus limiting their capability to learn similar entities or identify any common traits for all entities belonging to the same class. It is worthwhile to note that some classes such as *musical_instrument* and *tv_program* in Freebase have been shown to cluster well in the vector space [114]. A closer inspection reveals that these classes have very few and unique incoming relations such that the embeddings would be able to learn their features well. However, classes with unique representative properties are not very common in real-world datasets.

The key insight from our detailed analysis in this work is that while KG embeddings are assumed to be representing the semantics for entities and relations, in reality their

semantic soundness is severely restricted and highly dependent on the datasets on which they are trained. Experimental results have clearly shown that several prominent embedding models often record worse semantic capability for a majority of the entities in real-world datasets as compared to a simple heuristics based approach that can derive the semantics directly from KG triples without any additional information. These findings indicate that a thorough inspection of the advantages and weaknesses of KG embeddings is necessary when employing them for semantic tasks. While the semantic web community is focused on novel architectures for training the KG embeddings models, a careful eye on the generalizability of these models in terms of their semantic representation also deserves more attention. We hope this work will guide further research in this direction. Recent efforts towards the explainability in KG embedding models [13, 52] could be the first steps towards understanding these models that could benefit all semantic tasks that leverage them.

4.6 Summary

In this chapter, we presented a comprehensive analysis of the popular knowledge graph embedding models in terms of their semantic utility. The results from our classification and clustering experiments on top of these embeddings brings attention to the weaknesses in semantic representation of embeddings. We showed that embeddings fare poorly in terms of identifying the concepts or classes for a majority of the entities in the underlying knowledge graph and simple statistical approaches can compete very well with them. We also presented a detailed analysis of the reasons for limited semantic understanding of the embeddings relating to sparse and noisy links in real-world datasets. We hope the results from this work would serve as a precautionary tale and help the research community become cognizant of the realistic semantic benefits of knowledge graph embeddings, such that they can make prudent decisions when applying these embeddings to new problem statements and semantic tasks. It would be interesting to extend this analysis to include further and more recent embedding techniques.

In the next chapter, we continue our exploration KG embeddings and look closer into the semantic soundness of their predictions for KG completion. Further, we present our approach for improving the semantics of existing embedding models by explicitly providing ontological knowledge during the training process.

Chapter 5

Improving Knowledge Graph Embeddings with Ontological Reasoning

"I'm smart enough to know that I'm dumb."
— Richard P. Feynman

In the previous chapters, we have discussed the inherent issue of incompleteness in knowledge graphs and the role of knowledge graph embedding methods for the *knowledge completion* task, i.e. predicting new triples. Chapter 4 has also established the limitations of existing embedding models for semantic tasks. In this chapter, we further explore the embedding models with regard to the flaws in their training process that may lead to semantically incorrect predictions. Significantly, the lack of ontological knowledge during the training is identified as the main reason behind this issue. As such, we propose a novel technique called *ReasonKGE* that generates reasoning based negative samples to improve the model performance.

The contributions in this chapter are based on the work done in Jain et al. [81]. The rest of the chapter is divided into sections as follows — in Section 5.2 we discuss the related work that concerns with negative sampling techniques and inclusion of ontology in embedding models. Section 5.3 presents the necessary background and notations on KGs, ontologies and embedding models. In Section 5.4 our *ReasonKGE* approach is introduced and the different modules are described in detail. Then, in Section 5.5 we discuss the results of our empirical evaluation that demonstrate the improvement in the performance of embedding models with our approach. Finally, we summarize and conclude in Section 5.6.

5.1 Training of Embedding Models

As mentioned in Section 1.3, typically, the training of KG embedding models aims at discerning between correct (positive) and incorrect (negative) triples. A completion model then associates a score with every input triple. The goal of the embedding models is to rank every positive triple higher than all its negative alternatives. Therefore, the quality

of embedding models is heavily impacted by the generated negative triples. Since KGs store explicitly only positive triples, proper negative triple generation is acknowledged to be a challenging problem [42, 92, 182, 183].

In the majority of existing methods the generation of negative triples is done either completely at random [20], relying on the (local) closed world assumption [117], or by exploiting the KG structure for the generation of likely true negative samples (e.g. [1, 6, 183]). However, these methods do not guarantee that the generated negative samples are actually incorrect ones. In d'Amato et al. [42] this issue is partially addressed by taking as negative examples precomputed triples that are inconsistent with the KG and the accompanied ontology. Since the generation of all such possible inconsistent triples as negative samples is clearly infeasible in practice, only a subset of them is precomputed, and hence certain important inconsistent triples might be missing in the set obtained in [42]. Furthermore, as embedding models rely purely on the data in the input KGs, they often lose the real semantics of entities and relations, and hence provide undesired predictions [169]. This calls for more goal-oriented approaches in which ontological reasoning is used to verify and improve the actual predictions made by embedding models.

To address the presented shortcomings, in this work we propose an iterative method that dynamically identifies inconsistent predictions produced by a given embedding model via symbolic reasoning and feeds them as negative samples for retraining this model. We first start with any available negative sampling procedure (e.g., [92, 183]) and train the embedding model as usual. Then, among predictions made by the model, we select those that cause inconsistency when being added to the KG, as negative samples for the next iteration of our method. To avoid predicting similar wrong triples, along with the inconsistent triples explicitly inferred by the embedding model, we also generate triples that are semantically similar via a *generalization procedure*. To address the scalability problem that arises when integrating ontological reasoning into the training process of embedding models, we consider ontologies in an extension of the Description Logic (DL) *DL-Lite* [9] so that consistency checking and the generalization procedure can be performed efficiently. Our method can support any embedding model, and with the increasing number of iterations it yields better embeddings that make less inconsistent predictions and achieve higher prediction accuracy w.r.t. standard metrics.

In this chapter, we introduce the *ReasonKGE* framework for exploiting ontological reasoning to improve existing embedding models by advancing their negative sampling. To efficiently filter inconsistent embedding-based predictions, we exploit the locality property of light-weight ontologies. Moreover, in the spirit of previous work by Tran et al. [152] we generalize the computed inconsistent facts to a set of other similar ones to be fed back to the embedding model as negative samples. The evaluation of the proposed method on a set of state-of-the-art KGs equipped with ontologies, demonstrates that ontological reasoning exploited in the suggested way indeed improves the existing embedding models with respect to the quality of fact prediction.

5.2 Related Work

We discuss previous works related to our approach in separate categories, including those concerned with various negative sampling techniques for embedding models as well as

the ones that integrate ontological reasoning with embeddings in different ways. We also discuss previous works relating to inconsistencies in ontologies.

Negative sampling strategies. The closest to our method is the work by d'Amato et al. [42], in which ontologies are used to generate a selection of negative samples in the pre-processing step for training a certain embedding model. While we use this pre-processing based sampling as a baseline for comparison in Section 5.5, our method is different in that we do not generate all negative examples at once, but rather compute them iteratively on demand relying on the inconsistent predictions produced by the given embedding. The major advantage of the *ReasonKGE* method compared to [42] is the dynamic and adaptable nature of negative sample generation, wherein, the method is able to specifically target the weaknesses of the previously trained model by leveraging inconsistent predictions to derive negative samples, and use them for re-training of the model in next iterations. This is in contrast to the process of precomputing negative samples using ontology axioms as suggested in d'Amato et al. [42].

Another related method is concerned with type-constrained negative sampling [95]. Given a triple from the KG, the negative candidates (subjects or objects) are mined by constraining the entities to belong to the same type as that of the subject or object of the original triple. However, unlike our inconsistency-driven method, the typed-constrained sampling can generate false negatives. This sampling method can be in principle also used as the starting point for our method instead of the random sampling.

More distant random negative samplings generate false candidate triples based on the (local) closed world assumption [117]. Alternatives include Distributional Negative Sampling (DNS) [34] and its variation [6], where during training, given a positive triple, negative examples are generated by replacing it's entity with other similar entities. Unlike in our method, no ontological information is considered in these sampling strategies. The same holds for the triple perturbation or triple corruption approach [144].

Nearest Neighbor and *Near Miss sampling* [92] resp. exploit a pre-trained embedding model for generating negative samples by selecting triples that are close to the positive target triple in vector space. Intuitively, this strategy is supposed to help the model to learn to discriminate between positives and negatives that are very similar to each other. These approaches are similar to ours, in that the embedding training procedure itself is exploited for the generation of negative samples. However, in [92] no ontological knowledge is taken into account which is in contrast to our work.

Another research direction concerns making use of Generative Adversarial Networks (GANs) [24, 164, 182] for negative sampling. Ahrabian et al. [1] present structure-aware negative sampling (SANS), which utilizes the graph structure by selecting negative samples from a node's neighborhood. The NSCaching sampling method [183] suggests to sample negatives from a cache that can dynamically hold large-gradient samples. While in these works negative triples are updated dynamically like in our method, these approaches are totally different from ours, as they rely purely on the machine learning techniques, and do not consider any extra ontological knowledge. Thus, the proposals are rather complementary in nature.

Integration of ontological knowledge into KG embeddings. Another relevant line of work concerns the integration of ontological knowledge directly into embedding

models (e.g., [42, 55, 65, 95, 111, 169, 189]), which is typically done via changes in the loss function, rather than negative sampling.

For example, a related method *Embed2Reason (E2R)* has been proposed by Garg *et al.* [55]. *E2R* relies on the quantum logic, and injects ontology axioms via the loss function, by summing up the terms relevant for these axioms. However, it is unclear how this method captures the interaction among the axioms, which is often the reason for inconsistency. Since the available code of [55] only supports a limited set of axioms, i.e., `SubClassOf, SubPropertyOf, Domain, Range`, which are insufficient for generating inconsistencies, we could not perform a direct comparison of our method to *E2R*. Note that in general, our method is conceptually different from *E2R*. Indeed, in contrast to [55], we focus on ontology-driven targeted improvements of the negative sampling procedure with the goal of teaching a given embedding model to make only consistent predictions, and interactions among the axioms are key to our method. Moreover, our proposed approach can be built on top of any embedding model including [55], making the two methods rather complementary in nature.

The recent work [169] suggests to exploit ontological reasoning for verifying consistency of predictions made by a machine learning method (e.g., embedding or rule learning). However, instead of feeding inconsistent predictions back to the given embedding model, the authors propose to get rid of them and feed other consistent predictions along with the original KG as input to a further KG completion method. In Hao et al. [65] the ontology is explicitly included in the training data to jointly embed entities and concepts. By treating the ontology and KG in the same way, only very restricted ontological knowledge is accounted for.

Our work can be also positioned broadly within neural-symbolic methods, and we refer the reader to relevant publications [14, 179] for other less related neural-symbolic approaches.

Inconsistency in ontologies. The problems of explaining and handling inconsistency in ontologies have been tackled in different settings [15, 16, 71, 98, 128, 152]. However, typically these works focus on detecting inconsistency [16, 71], scalable reasoning [128, 152], or performing reasoning in the presence of such inconsistency [15, 98] assuming that the KG is constructed and complete. In other words, these approaches deal purely with data cleaning rather than KG completion. In contrast, our method integrates the reasoning process into the embedding models to improve the accuracy of predicted triples.

5.3 Background

In this section, we introduce the basic concepts and notations that will be used in the rest of the chapter. We assume countable pairwise disjoint sets $\mathsf{N_C}, \mathsf{N_P}$ and $\mathsf{N_I}$ of class names (*a.k.a.* types), property names (*a.k.a.* relations), and individuals (*a.k.a.* entities). We also assume the standard relation *rdf*:*type* (abbreviated as *type*) to be included in $\mathsf{N_P}$. A *knowledge graph* (KG) is denoted as $\mathcal{G}$ having a finite set of *triples* of the form $\langle s, p, o\rangle$, where $s \in \mathsf{N_I}, p \in \mathsf{N_P}, o \in \mathsf{N_I}$, if $p \neq \mathit{type}$, and $o \in \mathsf{N_C}$ otherwise. It is to be recalled that KGs typically follow Open World Assumption (OWA), meaning that they store only a fraction of positive facts. For instance, given the KG from Fig. 5.1 $\langle \mathit{john}, \mathit{type}, \mathit{person}\rangle$ and $\langle \mathit{john}, \mathit{livesIn}, \mathit{germany}\rangle$ are true KG facts; however, whether $\langle \mathit{john}, \mathit{worksAt}, \mathit{bosch}\rangle$

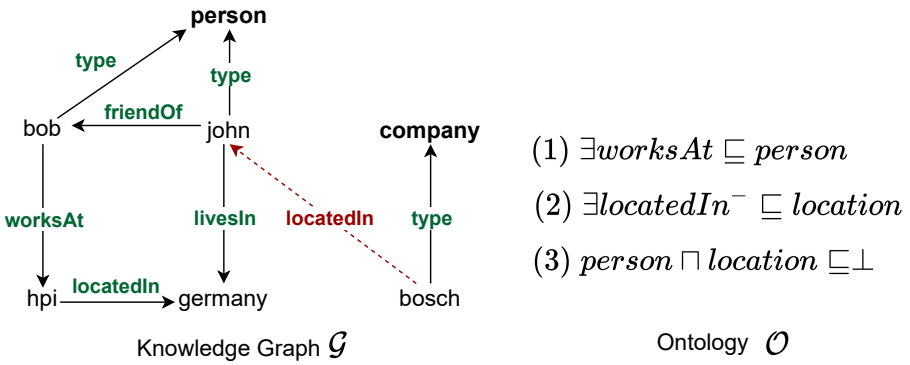


Figure 5.1: Example knowledge graph with its ontology, where solid links correspond to the true facts, while the dashed one to a spurious predicted fact.

holds or not is unknown. Given a triple α, we denote by $\mathsf{Ent}(\alpha)$ a set of all entities occurring in α and extend this notation to a set of triples as $\mathsf{Ent}(\mathcal{G}) = \bigcup_{\alpha \in \mathcal{G}} \mathsf{Ent}(\alpha)$.

An ontology $\mathcal{O}$ (*a.k.a.* TBox) is a set of axioms expressed in a certain Description Logic (DL) [10]. In this work we focus on *DL-Lite*$^{\mathcal{S}\sqcup}$, i.e., extension of *DL-Lite* [9] with transitive roles and concept disjunctions. Classes C denoting sets of entities, and roles R denoting binary relations between entities, obey the following syntax:

$$C ::= A \mid \exists R \mid A \sqcup B \mid A \sqcap B \mid \neg C$$
$$R ::= P \mid P^{-}$$

Here, $A, B \in \mathsf{N_C}$ are atomic classes and $P \in \mathsf{N_P}$ is an atomic property (i.e., binary relation). An ontology $\mathcal{O}$ is a finite set of axioms of the form $C_1 \sqsubseteq C_2$, $R_1 \sqsubseteq R_2$, $R \circ R \sqsubseteq R$, reflecting the transitivity of the relation R. The summary of the DL syntax in *DL-Lite*$^{\mathcal{S}\sqcup}$ and its translation to OWL 2[1] is presented in Table 5.1. In the rest of the paper, we assume that all ontologies in this work are expressed in *DL-Lite*$^{\mathcal{S}\sqcup}$.

Our running example of a KG with an ontology given in Figure 5.1 reflects the domain knowledge about people and their working places. The ontology states that (1) the domain of *worksAt* relation is *person*, (2) the range of *locatedIn* is *location*, and (3) *person* is disjoint with *location*.

Inconsistency and explanations. The semantics of knowledge graphs and ontologies is defined using the direct model-theoretic semantics via interpretations [115]. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$, the *domain* of $\mathcal{I}$, and an *interpretation function* $\cdot^{\mathcal{I}}$, that assigns to each $A \in \mathsf{N_C}$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to each $R \in \mathsf{N_R}$ a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each $a \in \mathsf{N_I}$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. This assignment is extended to (complex) classes and roles as shown in Table 5.1.

An interpretation $\mathcal{I}$ *satisfies* an axiom α (written $\mathcal{I} \models \alpha$) if the corresponding condition in Table 5.1 holds. Given a KG $\mathcal{G}$ and an ontology $\mathcal{O}$, $\mathcal{I}$ is a *model* of $\mathcal{G} \cup \mathcal{O}$ (written $\mathcal{I} \models \mathcal{G} \cup \mathcal{O}$) if $\mathcal{I} \models \alpha$ for all axioms $\alpha \in \mathcal{G} \cup \mathcal{O}$. We say that $\mathcal{G} \cup \mathcal{O}$ *entails* an axiom α (written $\mathcal{G} \cup \mathcal{O} \models \alpha$), if every model of $\mathcal{G} \cup \mathcal{O}$ satisfies α. A KG $\mathcal{G}$ is *inconsistent* w.r.t. an ontology $\mathcal{O}$ if no model for $\mathcal{G} \cup \mathcal{O}$ exists. In this case, $\mathcal{G} \cup \mathcal{O}$ is inconsistent. Intuitively, $\mathcal{G} \cup \mathcal{O}$ is inconsistent when some facts of $\mathcal{G}$ contradict some axioms of $\mathcal{O}$.

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

DL Syntax	OWL Syntax	Semantics
R	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
R^{-}	`ObjectInverseOf`(R)	$\{\langle e, d\rangle \mid \langle d, e\rangle \in R^{\mathcal{I}}\}$
A	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
$\top$	`owl:Thing`	$\Delta^{\mathcal{I}}$
$\bot$	`owl:NoThing`	$\emptyset$
$\neg C$	`ObjectComplementOf`(C)	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	`ObjectIntersectionOf`(C, D)	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	`ObjectUnionOf`(C, D)	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\exists P$	`ObjectSomeValuesFrom`$(P,$ `owl:Thing`$)$	$\{d \mid \exists e \in \Delta^{\mathcal{I}}. \langle d, e\rangle \in P^{\mathcal{I}}\}$
$C \sqsubseteq D$	`SubClassOf`(C, D)	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$P \sqsubseteq S$	`SubObjectPropertyOf`(P, S)	$P^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
$P \circ P \sqsubseteq P$	`TransitiveObjectProperty`(P)	$P^{\mathcal{I}} \circ P^{\mathcal{I}} \subseteq P^{\mathcal{I}}$
$\langle a, type, c\rangle$	`ClassAssertion`(C, a)	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$\langle a, p, b\rangle$	`ObjectPropertyAssertion`(P, a, b)	$\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in P^{\mathcal{I}}$

Table 5.1: Syntax and semantics of the ontology language considered in this paper where A, R are a class name and property name, respectively; C and D are class expressions, P, S are property expressions, and a, b are entities.

Under the considered ontology language, KG inconsistency has a locality property, i.e., the problem of checking inconsistency for a KG (w.r.t. an ontology $\mathcal{O}$) can be reduced to checking inconsistency for separated KG *modules* (w.r.t. $\mathcal{O}$) [152].

Definition 1 (Modules)**.** *Given a KG $\mathcal{G}$ and an entity $e \in \mathsf{Ent}(\mathcal{G})$, the* module *of e w.r.t. $\mathcal{G}$ is defined as $\mathcal{M}(e, \mathcal{G}) = \{\alpha \mid \alpha \in \mathcal{G}$ and e occurs in $\alpha\}$. We denote the set of all modules for individuals occurring in $\mathcal{G}$ as $\mathcal{M}_{\mathcal{G}} = \{\mathcal{M}(e, \mathcal{G}) \mid e \in \mathsf{Ent}(\mathcal{G})\}$.*

Lemma 1 (Consistency Local Property)**.** *Let $\mathcal{G}$ be a KG and $\mathcal{O}$ an ontology. Then $\mathcal{G} \cup \mathcal{O}$ is consistent iff $\mathcal{M}(a, \mathcal{G}) \cup \mathcal{O}$ is consistent for every $a \in \mathsf{Ent}(\mathcal{G})$.*

An *explanation* for inconsistency of $\mathcal{G} \cup \mathcal{O}$ [71], denoted by $\mathcal{E} = \mathcal{E}_{\mathcal{G}} \cup \mathcal{E}_{\mathcal{O}}$ with $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{G}$ and $\mathcal{E}_{\mathcal{O}} \subseteq \mathcal{O}$, is a (subset-inclusion) smallest inconsistent subset of $\mathcal{G} \cup \mathcal{O}$.

Example 1. *The KG from Fig. 5.1 with all facts including the dashed red one is inconsistent with the ontology $\mathcal{O}$, and a possible explanation for that is $\mathcal{E} = \mathcal{E}_{\mathcal{G}} \cup \mathcal{E}_{\mathcal{O}}$ with $\mathcal{E}_{\mathcal{G}} = \{\langle bosch, locatedIn, john\rangle, \langle john, type, person\rangle\}$ and $\mathcal{E}_{\mathcal{O}} = \{\exists locatedIn^{-} \sqsubseteq location$, $person \sqcap location \sqsubseteq \bot\}$.*

KG embeddings. As presented in Section 1.3, KG embeddings represent all entities and relations in a continuous vector space (usually as vectors or matrices called *embeddings*) and can be used to estimate the likelihood of a triple to be true via a scoring function: $f : \mathsf{N_I} \times \mathsf{N_P} \times \mathsf{N_I} \to \mathbb{R}$. Concrete scoring functions are defined based on various vector space assumptions. The likelihood that the respective assumptions of the embedding methods hold, should be higher for triples in the KG than for negative samples outside the KG. The learning process is done through minimizing the error induced from the assumptions given by their respective loss functions. Below we briefly recall widely-used assumptions for KG embeddings, particularly for TransE and ComplEx models that are used in this work:

(i) The translation-based assumption, *e.g.*, TransE [20] embeds entities and relations as vectors and assumes $\mathbf{v_s} + \mathbf{v_p} \approx \mathbf{v_o}$ for true triples, where $\mathbf{v_s}, \mathbf{v_p}, \mathbf{v_o}$ are vector

embeddings for subject s, predicate p and object o, respectively. The models that rely on the translation assumption are generally optimised by minimizing the following ranking-based loss function

$$\sum_{\langle s_i,p_i,o_i\rangle \in S^+} \sum_{\langle s'_i,p_i,o'_i\rangle \in S^-} [\gamma - f(s_i,p_i,o_i) + f(s'_i,p_i,o'_i)]_+ \tag{5.1}$$

where $f(s,p,o) = -\left\|\mathbf{v_s} + \mathbf{v_p} - \mathbf{v_o}\right\|_1$, S^+ and S^- correspond to the sets of positive and negative training triples respectively, that are typically disjoint.

(ii) The linear map assumption, e.g. ComplEx [153] embeds entities as vectors and relations as matrices. It assumes that for true triples, the linear mapping $\mathbf{M_p}$ of the subject embedding $\mathbf{v_s}$ is close to the object embedding $\mathbf{v_o}$: $\mathbf{v_s}\mathbf{M_p} \approx \mathbf{v_o}$. The loss function used for training the linear-map embedding models is given as follows:

$$\sum_{\langle s_i,p_i,o_i\rangle \in S^+} \sum_{\langle s'_i,p_i,o'_i\rangle \in S^-} l(1, f(s_i,p_i,o_i)) + l(-1, f(s'_i,p_i,o'_i))) \tag{5.2}$$

where $f(s,p,o) = \mathbf{v}_s\mathbf{M}_p\mathbf{v}_o$ and $l(\alpha,\beta) = log(1 - exp(-\alpha\beta))$.

5.4 Ontological Reasoning for Negative Sampling

While a variety of embedding models exist in the literature [165], one of the major challenges for them to perform accurate fact predictions is finding an effective way for generation of relevant negative samples [42, 140, 164]. Commonly used approaches for negative sampling randomly corrupt existing triples by perturbing their subject, predicate or object [20, 38, 144] or rely on the (local) closed world assumption (LCWA). Based on CWA all triples not present in the KG are assumed to be false, while LCWA is a variation of CWA, in which for every $\langle s,p,o\rangle$, only facts of the form $\langle s,p,o'\rangle \notin \mathcal{G}$ are assumed to be false. For instance, given the facts in Figure 5.1, the corrupted negative triples obtained based on the LCWA could be $\langle john, livesIn, hpi\rangle$ or $\langle bob, worksAt, bosch\rangle$.

However, since KGs follow OWA, the standard sampling methods might often turn out to be sub-optimal, resulting in false positive negative samples [42]. For example, the corrupted triple $\langle bob, worksAt, bosch\rangle$ from above might actually be true in reality.

A natural method to avoid false positives and generate only relevant negative samples is by relying on ontologies with which KGs are typically equipped. A naive approach for that is to generate all facts that can be formed using relations and entities in $\mathcal{G}$ (i.e., construct the Herbrand base) and check which among the resulting candidates are inconsistent with $\mathcal{G} \cup \mathcal{O}$. As modern KGs store millions of facts, the described procedure is infeasible in practice. To still sample some inconsistent triples, in [42] facts $p(s,o) \in \mathcal{G}$ are corrupted by substituting s (resp. o) with s' (resp. o') s.t. s and s' (resp. o and o') belong to disjoint classes and the resulting corrupted triple is inconsistent. For example, given $\mathcal{G}$ and $\mathcal{O}$ in Fig 5.1, from $\langle bob, worksAt, germany\rangle$ we can obtain $\alpha_1 = \langle germany, worksAt, germany\rangle$ or $\alpha_2 = \langle bob, worksAt, john\rangle$, as *person* is disjoint with *location*. However, this method might fail to avoid the inconsistent triples that the model actually predicts. E.g., $\langle bosch, locatedIn, john\rangle$ is not generated by this method as a negative example, and the model can in principle still predict it.

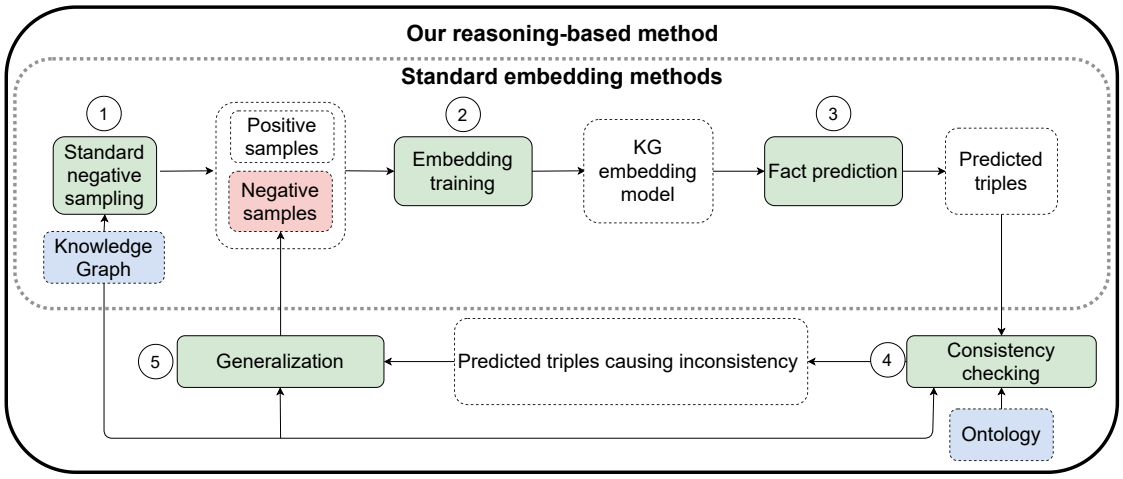


Figure 5.2: Standard embedding pipeline (grey dotted frame) and our reasoning-based method (black frame) in a nutshell

Therefore, instead of pre–computing a static set of negative examples, we propose to iteratively generate and improve this set (and subsequently also the embedding model) *dynamically* by computing a collection of negative samples in a guided fashion from embedding model based on its predictions that are inconsistent with the ontology. We refer to this negative sampling strategy as *dynamic sampling*. On the one hand, this intuitively allows us to overcome the computational challenge of generating all possible negative examples at once, but rather add the most relevant ones on demand to the embedding training process. On the other hand, this approach is capable of reducing frequently encountered errors (in terms of inconsistent predictions) for particularly difficult triples by directly incorporating feedback from incorrect predictions back to the model for further training. Indeed, when trained for increasing number of iterations, such method is capable of generating embeddings that predict fewer inconsistent facts, as empirically demonstrated in Section 5.5.

5.4.1 Overview of *ReasonKGE*

In this section, we describe in more detail the proposed framework referred to as *ReasonKGE*, whose main steps are depicted in Figure 5.2. Given a KG, ontology and an embedding method, we aim at generating an enhanced KG embedding, which is trained for predicting facts that are consistent with the KG and the ontology at hand.

The input to our method (represented by blue dashed boxes) is the KG and the ontology, while the output (the red dashed box) is the set of negative samples that is incorporated during the iterative training and tuning of a KG embedding model in each iteration. As negative samples are obtained based on predictions made by an existing embedding, a baseline model is required in the first iteration. For this, in step (1) we obtain the negative samples with **standard negative sampling** using any of the existing methods [20, 38, 42, 144]. We then perform **embedding training** in step (2) to construct the initial KG embedding model.

This model is used for obtaining predictions and computing the set of negative samples for the next training iteration. Specifically, in step (3) the model is used for **fact prediction** as follows. For every triple in the training set, given its subject s and

Algorithm 1 Training embedding models with negative samples using ontological reasoning

Input : Baseline embedding model $\mathbf{E}$, a knowledge graph $\mathcal{G}$, and an ontology $\mathcal{O}$

/* Step 1 and Step 2 */

1 Train the baseline embedding model $\mathbf{E}$ for a certain number of epochs. /* Retrain the baseline model with negative samples derived from reasoning */

2 **Loop**

/* Step 3 */

3 **foreach** *triple* $\alpha = \langle s, p, o \rangle \in \mathcal{G}$ **do**

4 Get a set $\mathsf{Predictions}(\alpha)$ of predicted triples of the form $\langle s, p, \hat{o} \rangle$ and $\langle \hat{s}, p, o \rangle$ by giving $\langle s, p \rangle$ and $\langle p, o \rangle$ as inputs to $\mathbf{E}$ and obtaining predicted entities $\hat{o}$ and $\hat{s}$, respectively. /* Step 4 */

5 $\mathsf{NegSamples}(\alpha) \leftarrow \emptyset$ **foreach** *predicted triple* $\beta \in \mathsf{Predictions}(\alpha)$ **do**

6 Compute the relevant set $\mathsf{Relv}(\beta, \mathcal{G})$ of β w.r.t. $\mathcal{G}$. **if** $\mathsf{Relv}(\beta, \mathcal{G}) \cup \mathcal{O}$ *is inconsistent* **then**

/* Step 5 */

7 Compute explanations for inconsistency. **foreach** *inconsistency explanation* $\mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{O}$ **do**

8 Compute $\mathsf{GeneralizedSamples}(\beta)$ as defined in Definition 4. $\mathsf{NegSamples}(\alpha) \leftarrow \mathsf{NegSamples}(\alpha) \cup \mathsf{GeneralizedSamples}(\beta)$

9 Retrain $\mathbf{E}$ in which, for each training step that considers $\alpha \in \mathcal{G}$, $\mathsf{NegSamples}(\alpha)$ is used as negative samples in the loss function, e.g. Equation 5.1 or Equation 5.2.

predicate p, we retrieve the top ranked object and obtain the fact $\langle s, p, o \rangle$ as the respective prediction. The same is done inversely for computing the top ranked subject given the object o and predicate p in the training set. Note that only triples that are not in the training set are considered as predictions. In step (4) we check whether the predicted triple complies with the ontology relying on the **consistency checking** procedure. In case the respective triple is found to be inconsistent, in step (5) we generalize it to other semantically similar triples using the **generalization** procedure to obtain an extended set of negative samples. Finally, the computed negative samples, both for subject and object predictions are fed back as input to the next iteration of the embedding training process. The detailed steps are presented in Algorithm 1 and explained in what follows.

5.4.2 Consistency checking

The goal of the consistency checking procedure is to verify which predictions made by the embedding model in step (3) are inconsistent with the ontology $\mathcal{O}$ and the original KG $\mathcal{G}$. In principle, any reasoner capable of performing consistency checking effectively for ontologies in the considered *DL-Lite*$^{\mathcal{S}\sqcup}$ language can be used in this step. As the task that we consider concerns verifying whether a particular triple causes inconsistency, for the target DL when performing the consistency check one does not need to account for the whole KG, but only a small subset of relevant facts. To this end, we define the *relevant sets* as follows.

Definition 2 (Relevant set)**.** *Let $\mathcal{G}$ be a KG and α be a triple. The* relevant set $\mathsf{Relv}(\alpha, \mathcal{G})$ *of α w.r.t. $\mathcal{G}$ is defined as* $\mathsf{Relv}(\alpha, \mathcal{G}) = \{\alpha\} \cup \{\beta \in \mathcal{G} \,|\, \mathsf{Ent}(\beta) \cap \mathsf{Ent}(\alpha) \neq \emptyset\}$.

Example 2. *For $\alpha = \langle bosch, locatedIn, john \rangle$ and $\mathcal{G}$ in Fig. 5.1, we have the following relevant set* $\mathsf{Relv}(\alpha, \mathcal{G}) = \{\alpha\} \cup \{\langle john, livesIn, germany \rangle, \langle john, friendOf, bob \rangle, \langle john, type, person \rangle, \langle bosch, type, company \rangle\}$.

The following proposition allows us to reduce the consistency checking of $\alpha \cup \mathcal{G} \cup \mathcal{O}$ to the consistency checking of $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$.

Proposition 2. *Let $\mathcal{G}$ be a knowledge graph, $\mathcal{O}$ an ontology such that $\mathcal{G} \cup \mathcal{O}$ is consistent, and α a triple. Then, $\alpha \cup \mathcal{G} \cup \mathcal{O}$ is consistent iff $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$ is consistent.*

Proof. Since $\mathsf{Relv}(\alpha, \mathcal{G}) \subseteq \mathcal{G}$, we have $\alpha \cup \mathcal{G} \cup \mathcal{O}$ being consistent implies that $\mathsf{Relv}(\alpha, \mathcal{G})$ $\cup \mathcal{O}$ is also consistent. We start showing the remaining direction by assuming that $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$ is consistent and then show that $\alpha \cup \mathcal{G} \cup \mathcal{O}$ is also consistent. Let $\alpha = \langle s, p, o \rangle$, by Definition 2, we have $\mathsf{Relv}(\alpha, \mathcal{G}) = \mathcal{M}(s, \alpha \cup \mathcal{G}) \cup \mathcal{M}(o, \alpha \cup \mathcal{G})$. Since $\mathcal{G} \cup \mathcal{O}$ is consistent, by Lemma 1, we have $\mathcal{M}(e, \mathcal{G}) \cup \mathcal{O}$ is consistent for every entity in $\mathsf{Ent}(\mathcal{G}) \setminus \{s, o\}$. Since $e \notin \{s, o\}$, we have $\mathcal{M}(e, \mathcal{G}) = \mathcal{M}(e, \alpha \cup \mathcal{G})$, which implies $\mathcal{M}(e, \alpha \cup \mathcal{G}) \cup \mathcal{O}$ is consistent ($\star$). From the assumption that $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$ is consistent and $\mathsf{Relv}(\alpha, \mathcal{G}) = \mathcal{M}(s, \alpha \cup \mathcal{G}) \cup \mathcal{M}(o, \alpha \cup \mathcal{G})$, we obtain $\mathcal{M}(s, \alpha \cup \mathcal{G})$ and $\mathcal{M}(o, \alpha \cup \mathcal{G})$ are consistent w.r.t. $\mathcal{O}$ (†). From ($\star$) and (†) we have $\alpha \cup \mathcal{G} \cup \mathcal{O}$ is consistent using Lemma 1. □ □

Relying on Proposition 2, it is sufficient to check the consistency of a triple α with respect to $\mathcal{G} \cup \mathcal{O}$ using $\mathsf{Relv}(\alpha, \mathcal{G})$ rather than the whole KG. We make use of this property in step (4), and for every prediction produced by the embedding model, we first construct the relevant set for the respective prediction, and then perform the consistency check relying only on the corresponding relevant sets.

Example 3. *Assume that the fact $\alpha = \langle bosch, locatedIn, john \rangle$ has been predicted by the embedding model in step (3). Then in the consistency checking step (4) we first construct the relevant set for α as $\mathsf{Relv}(\alpha, \mathcal{G})$ given in Example 2 and check the consistency of $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$. Clearly, we have $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O} = \{\langle bosch, locatedIn, john \rangle\} \cup$ $\{\langle john, livesIn, germany \rangle, \langle john, type, person \rangle, \langle john, friendOf, bob \rangle, \langle bosch, type,$ $company \rangle\} \cup \mathcal{O}$ is inconsistent, since $\langle bosch, locatedIn, john \rangle$ and $\{\exists locatedIn^- \sqsubseteq$ $location\} \in \mathcal{O}$ imply that $\langle john, type, location \rangle$, which contradicts the fact that $\langle john,$ $type, person \rangle \in \mathcal{G}$ and $person \sqcap location \sqsubseteq \bot \in \mathcal{O}$. Thus, we have that $\alpha \cup \mathcal{G} \cup \mathcal{O}$ is inconsistent by monotonicity. Proposition 2 further guarantees that it is sufficient to check the consistency of $\alpha \cup \mathcal{G} \cup \mathcal{O}$ this way.*

5.4.3 Negative sample generalization

Given each triple of the input KG in the training step, one needs to sample not a single corrupted triple but a set of such triples to train the embedding model at hand. In other words, the inconsistent prediction needs to be *generalized* to obtain a set of similar inconsistent facts within the KG, which ideally have the same structure. Therefore, once an inconsistent prediction for a triple is identified, we proceed with detecting the inconsistency pattern from that prediction and relying on the respective pattern we generate other similar incorrect triples (in step 5 of our method). This allows us to compute sufficient number of negative samples for retraining the embedding model, and to give hints to the embedding model about the wrong patterns that it learned, subsequently avoiding the prediction of similar incorrect triples in next iterations.

A naive approach to obtain the generalized triples of an inconsistent predicted triple, e.g. $\langle s, p, \hat{o} \rangle$, is to replace $\hat{o}$ by another entity o in the input KG such that o has similar KG neighborhood as $\hat{o}$. However, it might happen that only a subset of triples containing $\hat{o}$ is inconsistent w.r.t. the ontology. Therefore, it is sufficient to find such o that it has similar triples as in that subset. This will increase the number of generalized triples as

demonstrated in Example 4. To compute a subset of triples of $\hat{o}$ that is inconsistent w.r.t. the ontology, we compute explanations for the inconsistency of $\mathsf{Relv}(\langle s, p, \hat{o}\rangle, \mathcal{G}) \cup \mathcal{O}$.

Example 4. *Consider the KG $\mathcal{G}$ and ontology $\mathcal{O}$ as in Figure 5.1. Assume that $\alpha = \langle bosch, locatedIn, john\rangle$ is the predicted triple, i.e., the embedding model predicted john as the object entity for the given subject bosch and relation locatedIn. The explanation for inconsistency of $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$ is $\mathcal{E} = \mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{O}$, for which it holds that $\mathcal{E}_\mathcal{G} = \{\langle bosch, locatedIn, john\rangle, \langle john, type, person\rangle\}$ and $\mathcal{E}_\mathcal{O} = \{\exists located^- \sqsubseteq location$, $person \sqcap location \sqsubseteq \bot\}$. Note that there is no other entity in $\mathcal{G}$ that has similar triples as those for john. However, if we restrict to the triples in the explanation for inconsistency of $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$, then bob has the same neighborhood triple $\langle bob, type, person\rangle$ as john (the predicted triple is ignored). Therefore, we can take $\langle bosch, locatedIn, bob\rangle$ as another negative sample, which together with $\mathcal{G}$ is clearly inconsistent w.r.t. $\mathcal{O}$.*

To formally obtain generalized triples as in Example 4, we rely on the notion of *local type* of an entity [61, 62, 152] as follows.

Definition 3 (Local Types)**.** *Let $\mathbf{T}$ be a set of triples and e an entity occurring in $\mathbf{T}$. Then, the local type of e w.r.t. $\mathbf{T}$, written as $\tau(e, \mathbf{T})$ or $\tau(e)$ when $\mathbf{T}$ is clear from the context, is defined as a tuple $\tau(e) = \langle \tau_i(e), \tau_c(e), \tau_o(e)\rangle$, where $\tau_i(e) = \{p \mid \langle s, p, e\rangle \in \mathcal{G}\}$, $\tau_c(e) = \{t \mid \langle e, type, t\rangle \in \mathcal{G}\}$, and $\tau_o(e) = \{p' \mid \langle e, p', o\rangle \in \mathcal{G}\}$. The local type $t = \langle t_i, t_c, t_o\rangle$ is* smaller than or equal to *the local type $t' = \langle t'_i, t'_c, t'_o\rangle$, written as $t \preceq t'$, iff $t_i \subseteq t'_i, t_c \subseteq t'_c$, and $t_o \subseteq t'_o$.*

Intuitively, a local type of an entity represents a set of types (τ_c) as well as the incoming relations (τ_i) and outgoing relations (τ_o) for that entity in a set of triples.

Example 5 (Example 4 continued)**.** *For bob in Fig. 5.1, we have the local type of bob w.r.t. $\mathcal{G}$ being $\tau(bob) = \langle\{friendOf\}, \{person\}, \{worksAt\}\rangle$. The local type of john w.r.t. $\mathcal{E}_\mathcal{G} \setminus \alpha$ is $\tau(john) = \langle\emptyset, \{person\}, \emptyset\rangle$ and it holds that $\tau(john) \preceq \tau(bob)$.*

We now define the set of generalized samples of a given inconsistent predicted triple.

Definition 4 (Generalized Samples)**.** *Let $\mathcal{G}$ be a KG, $\mathcal{O}$ an ontology, and $\alpha = \langle s, p, \hat{o}\rangle$ be a triple in which $\hat{o}$ is predicted by an embedding model given the subject entity s and relation p. Furthermore, let $\mathcal{E} = \mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{O}$ be an inconsistency explanation of $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$. Then, the set of* generalized samples *of α (w.r.t. $\hat{o}$, $\mathcal{E}$, and $\mathcal{G}$) is defined as $\mathsf{GeneralizedSamples}(\alpha, \hat{o}) = \{\langle s, p, o\rangle \mid \tau(\hat{o}, \mathcal{E}_\mathcal{G} \setminus \alpha) \preceq \tau(o, \mathcal{G})\}$. The generalized samples $\mathsf{GeneralizedSamples}(\beta, \hat{s})$ of $\beta = \langle \hat{s}, p, o\rangle$, in which $\hat{s}$ is predicted by an embedding model, is defined analogously. When it is clear from the context, we often write $\mathsf{GeneralizedSamples}(\alpha)$ without mentioning the corresponding entity.*

Example 6 (Example 5 continued)**.** *According to Definition 4 and the local types of john and bob computed in Example 5, we have $\mathsf{GeneralizedSamples}(\alpha) = \{\alpha\} \cup \{\langle bosch, LocatedIn, bob\rangle\}$.*

The following Lemma guarantees that if a triple is inconsistent (together with the input KG) w.r.t. an ontology $\mathcal{O}$ then all generalized triples of that triple are also inconsistent.

Lemma 3. *Let $\mathcal{G}$ be a KG, $\mathcal{O}$ an ontology, α a triple such that $\mathsf{Relv}(\alpha, \mathcal{G}) \cup \mathcal{O}$ is inconsistent with an explanation $\mathcal{E} = \mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{O}$, and $\mathsf{GeneralizedSamples}(\alpha)$ is the set of generalized triples of α w.r.t. $\mathcal{E}$, $\mathcal{G}$, and some entity occurring in α. Then, we have $\mathsf{Relv}(\beta, \mathcal{G}) \cup \mathcal{O}$ is inconsistent for every $\beta \in \mathsf{GeneralizedSamples}(\alpha)$.*

Sketch. W.l.o.g. let $\alpha = \langle s, p, \hat{o} \rangle$, $\mathsf{GeneralizedSamples}(\alpha)$ is w.r.t. $\hat{o}$, and $\beta = \langle s, p, o \rangle$. Using the result in [152], one can show that if $\langle s, p, \hat{o} \rangle \in \mathcal{E}_\mathcal{G}$ then $\mathcal{E}_\mathcal{G}$ does not contain $\langle s', p, o \rangle$, where $s \neq s'$ due to the minimality of explanations. Together with the condition $\tau(\hat{o}) \preceq \tau(o)$, we can construct a homomorphism from $\mathsf{Relv}(\alpha, \mathcal{G})$ to $\mathsf{Relv}(\beta, \mathcal{G})$, which implies that $\mathsf{Relv}(\beta, \mathcal{G}) \cup \mathcal{O}$ is inconsistent. □ □

We now describe the details of step (5). For each predicted triple that is inconsistent w.r.t. the input KG and the ontology, we compute explanations for inconsistency, and for each such explanation, we obtain the generalized triples using Def. 4. These generalized triples are then used as negative samples to retrain the embedding model.

5.5 Experiments

We have implemented the proposed method in a prototype system *ReasonKGE* and evaluated its performance on the commonly used datasets enriched with ontologies. In this section, we present the results of the evaluation in terms of the impact of our method on the quality of fact predictions compared to the baselines.

5.5.1 Experimental setup

Datasets. Among commonly used datasets for evaluating embedding models, we chose those datasets that are equipped with ontologies. More specifically, the following datasets with their respective ontologies have been selected (Yago3-10 was already introduced in the last chapter, it is mentioned here again for the sake of completeness) —

- **LUBM3U:** A synthesized dataset derived from the Lehigh University Benchmark [63]. The ontology describing the university domain contains 325 axioms. The respective KG stores data for 3 universities.
- **Yago3-10:** A subset of the widely used Yago dataset. We use the ontology with 4551 axioms introduced in [145] based on Yago schema and class hierarchy.
- **DBpedia15K:** A subset of DBpedia KG proposed in [103]. We exploit the general DBpedia ontology enriched with axioms reflecting the disjointness of classes. The ontology comprises of 3006 axioms.

The statistics of the respective datasets are presented in Table 5.2.

Embedding Models. To demonstrate the benefits of the proposed iterative ontology-driven negative sampling, we apply our method over the following widely used embeddings: ComplEx and TransE. These models have been selected as prominent examples of translation-based and linear-map embeddings. While more recent embedding models exist in the literature, as shown in [140] classical embeddings are in fact very competitive when combined with effective parameter search. Thus, as baselines we have selected the most widely used and popular embedding models with the best parameters found using the LibKGE library [140].

We also consider another baseline [42] that incorporates background knowledge into the embedding models. We refer to such technique as *static sampling* because in contrast to our proposed *dynamic sampling* method, the approach from [42] generates the negative

Table 5.2: Knowledge graph statistics

	LUBM3U	Yago3-10	DBpedia15K
# Entities	127,645	123,182	12,842
# Predicates	28	37	279
# Training Facts	621,516	1,079,040	69,320
# Validation Facts	77,689	5,000	9,902
# Test Facts	77,689	5,000	19,805
# TBox Axioms	325	4,551	3,006

samples for all triples of the KG in the pre-processing step. Since the authors in [42] only mentioned that they utilized ontology axioms such as *Domain, Range, Functional,* and *Disjointness* without describing the exact procedure of how these were actually exploited to generate negative samples, we implemented such static sampling strategy based on our best knowledge and present detailed steps in Algorithm 2. Intuitively, for each entity e we first compute both asserted classes (explicitly known in the input KG) and derived classes (using ontology axioms together with the input KG) to which the entity belongs. Based on the *DisjointClasses* axioms, we then identify a set of entities that belong to any class known to be disjoint with one of the classes of e and refer to these entities as a set of *corrupted entities* for e. Such corrupted entities for e are then subsequently used to generate negative samples in the training step that considers triples in which e occurs.

One of the main steps of Algorithm 2 is to compute the TypeSet of an entity, which is the set of classes to which the entity belongs. For each entity in the KG, the *local type* of the entity (as defined in 3) is leveraged. A TypeSet of each entity is created as follows - the set of types (τ_c) is extended by adding the respective superclasses (parent types) of the classes present in (τ_c). For the set of incoming relations (τ_i), the super-relations of all incoming relations in the set (τ_i) are calculated, and the classes belonging to the range of these relations are extracted. Similarly, for the set of outgoing relations (τ_o), the super-relations of all outgoing relations in the set (τ_o) are computed and the classes belonging to the domain of these relations are extracted. The TypeSet is constructed by taking the union of the extracted classes. Thereafter, the set of classes that are disjoint with any of the classes in TypeSet are retrieved. For each entity e, every entity e' is added to the set of corrupted entities of e if e' has some type that is disjoint with at least one type in TypeSet of e.

During the training steps of embedding models, for each KG triple, the set of negative samples can be obtained by replacing the subject or object entity with the corresponding set of pre-computed corrupted entities.

Measures. We evaluate the performance of the embedding models in terms of the traditional metrics i.e ***MRR*** and ***Hits@k*** in the filtered setting [20]. In addition, we also compute the proportion of inconsistent facts (***Inc@k***) ranked in the *top-k* predictions produced by the presented methods. The measure ***Inc@k*** intuitively reflects how well the model is capable of avoiding inconsistent predictions (the lower the better).

System Configuration. In the experiments, we used HermiT [60] as the reasoner and the explanation method in [71] to compute inconsistency explanations. We run *ReasonKGE* for multiple iterations. In every iteration, the model is trained for $n = 100$ epochs during which, for each subject and object of a triple, $m >= 1$ negative examples are generated. We exploit the optimal value of m tuned for the respective baseline model. In the first iteration, m negative samples are generated using the default random sampling

Algorithm 2 Precomputing negative samples using ontology axioms

Input : A knowledge graph $\mathcal{G}$, and an ontology $\mathcal{O}$
Output : A set of negative samples $\mathsf{NegSamples}(\langle s,p,o\rangle)$ for each triple $\langle s,p,o\rangle$ in $\mathcal{G}$

/* Compute classes/types for each entity in $\mathcal{G}$ */
foreach *entity e occurring in* $\mathcal{G}$ **do**
  $\mathsf{TypeSet}(e) \leftarrow \emptyset$ Compute local type $\tau(e) = \langle \tau_i(e), \tau_c(e), \tau_o(e)\rangle$ of e w.r.t. $\mathcal{G}$ /* Compute super classes/types of e */
  $\tau'_c(e) = \{B \mid \mathcal{O} \models A \sqsubseteq B, \text{ and } A \in \tau_c(e)\}$. /* Compute incoming and outgoing super-properties/relations of e by calling a reasoner */
  $\tau'_i(e) = \{S \mid \mathcal{O} \models R \sqsubseteq S, \text{ and } R \in \tau_i(e)\}$ $\tau'_o(e) = \{S \mid \mathcal{O} \models R \sqsubseteq S, \text{ and } R \in \tau_o(e)\}$ /* Calculate the TypeSet (e) for the entity e */
  $\mathsf{TypeSet}(e) = \tau'_c(e) \cup \{A \mid \mathcal{O} \models \mathit{DomainOf}(R) \sqsubseteq A, R \in \tau'_o(e)\} \cup \{B \mid \mathcal{O} \models \mathit{RangeOf}(P) \sqsubseteq B, P \in \tau'_i(e)\}$

/* Compute the set of corrupted entities for e */
foreach *entity e occurring in* $\mathcal{G}$ **do**
  $\mathsf{DisjointType}(e) \leftarrow \emptyset$ **foreach** $A \in \mathsf{TypeSet}(e)$ **do**
    $\mathsf{DisjointType}(e) = \mathsf{DisjointType}(e) \cup \{B \mid \mathcal{O} \models A \sqcap B \sqsubseteq \bot\}$
  $\mathsf{CorruptedEntities}(e) \leftarrow \{e' \mid \mathsf{DisjointType}(e) \cap \mathsf{TypeSet}(e') \neq \emptyset\}$

/* Compute negative samples for each triple in $\mathcal{G}$ */
foreach $\langle s,p,o\rangle \in \mathcal{G}$ **do**
  $\mathsf{NegSamples}(\langle s,p,o\rangle) \leftarrow \{\langle s',p,o\rangle \mid s' \in \mathsf{CorruptedEntities}(s)\} \cup \{\langle s,p,o'\rangle \mid o' \in \mathsf{CorruptedEntities}(o)\}$

Table 5.3: Link prediction results

Model	KG	Default Training			Static Sampling			*ReasonKGE*		
		MRR	*Hits@1*	*Hits@10*	*MRR*	*Hits@1*	*Hits@10*	*MRR*	*Hits@1*	*Hits@10*
TransE	LUBM3U	0.119	0.069	0.214	0.125	**0.082**	0.213	**0.135**	0.079	**0.256**
	Yago3-10	0.226	0.044	0.537	0.351	0.183	0.621	**0.367**	**0.197**	**0.629**
	DBpedia15k	0.109	0.061	0.206	0.101	0.073	0.254	**0.118**	**0.101**	**0.299**
ComplEx	LUBM3U	0.159	0.119	0.242	0.181	0.136	0.276	**0.233**	**0.195**	**0.313**
	Yago3-10	0.482	0.400	0.643	0.515	0.431	0.665	**0.530**	**0.453**	**0.668**
	DBpedia15k	0.099	0.061	0.174	0.098	0.107	0.193	**0.115**	**0.125**	**0.221**

strategy.[2] In the subsequent iterations, we use the trained model to obtain the top $k = 1$ subject and object predictions and compute the inconsistent negative samples to be used for the next iteration of the embedding training as described in Section 5.4. The number m of negative samples for the next iteration is dynamically computed based on the statistical mean of the size of the generalized samples sets as an indicator.

5.5.2 Results

The results of the conducted experiments illustrate the benefit of *ReasonKGE* in producing higher quality predictions with less inconsistencies compared to the baselines.

Link prediction quality. Table 5.3 reports the results for the link prediction task obtained by our method and the baselines. Both TransE and ComplEx were trained using the default random sampling strategy [20], the *static sampling* [42], and using *ReasonKGE* for 3 iterations. For fair comparison, the number of the training epochs was kept the same as for *ReasonKGE* in all cases (i.e., 300 epochs).

[2]For each triple the subject (resp. object) is randomly perturbed to obtain m samples [20].

Table 5.4: Link prediction results for different iterations

Model	KG	*ReasonKGE*- Iteration 2				*ReasonKGE*- Iteration 3			
		MRR	*Hits@1*	*Hits@3*	*Hits@10*	*MRR*	*Hits@1*	*Hits@3*	*Hits@10*
TransE	LUBM3U	0.133	0.078	0.159	0.242	0.135	0.079	0.162	0.256
	Yago3-10	0.356	0.184	0.493	0.627	0.367	0.197	0.511	0.629
	DBpedia15k	0.116	0.091	0.130	0.287	0.118	0.101	0.132	0.299
ComplEx	LUBM3U	0.229	0.190	0.237	0.310	0.233	0.195	0.240	0.313
	Yago3-10	0.521	0.442	0.569	0.664	0.530	0.453	0.577	0.668
	DBpedia15k	0.111	0.119	0.154	0.216	0.115	0.125	0.162	0.221

One can observe that reasoning-based sampling consistently achieves better results than random sampling for training all considered embeddings on all KGs. For the Yago3-10 dataset the improvements are the most significant, achieving more than 10% enhancement for all measures over TransE. This indicates the advantage of ontology-based reasoning for enhancing the existing KG embeddings.

The comparison of our dynamic sampling method against static sampling [42] presented in Table 5.3 reveals that *ReasonKGE* outperforms the *static sampling* approach in almost all cases, which illustrates the benefits of exploiting inconsistent predictions as negative samples dynamically using our method, as opposed to their pre-computation.

By keeping the same training configuration and total number of training epochs, we ensure that the reflected performance gains are not merely due to additional training steps, but rather a result of the proposed reasoning-based approach.

Intermediate training results. We also present the complete results obtained by the *ReasonKGE* method at all iterations. As explained in Section 5.5.1, in the first iteration the negative samples generated using the default random sampling technique are exploited for training the embedding model. Thereafter, further iterations leverage the trained model from the previous iteration to predict subjects for given relations and objects, as well as similarly, to predict objects for given relations and subjects. Predicted triples that are found to be inconsistent w.r.t. the existing KG and ontology are then subsequently used for the generation of further negative samples for the next round of model training. This process is repeated for multiple iterations until no significant improvement in the performance of the embedding model is observed. In Table 5.4 we present the results of *all* iterations of our method. With every iteration, the model is trained for additional 100 epochs, i.e., in the second iteration the model training has been performed for 200 epochs, while in the third iteration altogether for 300 epochs respectively.

It can be seen that the improvement from iteration 2 to iteration 3 is below approximately 1% for all datasets. Therefore, in our experiments reported in Table 5.3 of Section 5.5 the training has been stopped at the third iteration, and the best results obtained have been compared to the results for the default training approach run for the same number of epochs (i.e., 300). In general, the differences in the results of the model obtained with the increasing number of iterations can be used as a stopping criteria for finalizing the training process, i.e., the small difference witnesses the convergence of the training process.

Table 5.5: Ratio of inconsistent predictions (the lower, the better)

Model	KG	Prediction	Default Training		Static Sampling		*ReasonKGE*	
			Inc@1	*Inc@10*	*Inc@1*	*Inc@10*	*Inc@1*	*Inc@10*
TransE	LUBM3U	*subject*	0.169	0.270	0.428	0.250	**0.037**	**0.133**
		object	0.095	0.097	0.212	0.104	**0.005**	**0.007**
	YAGO3-10	*subject*	**0.075**	0.280	0.629	0.492	**0.075**	**0.273**
		object	0.026	0.136	0.114	**0.089**	**0.020**	0.117
	DBpedia15K	*subject*	0.311	0.652	0.401	0.663	**0.217**	**0.585**
		object	0.413	0.538	0.428	0.544	**0.170**	**0.460**
ComplEx	LUBM3U	*subject*	0.041	0.097	0.177	0.136	**0.036**	**0.069**
		object	0.008	0.012	**0.003**	**0.007**	0.005	**0.007**
	YAGO3-10	*subject*	0.113	0.198	0.169	**0.128**	**0.071**	0.143
		object	0.037	0.115	0.065	0.084	**0.015**	**0.074**
	DBpedia15K	*subject*	0.488	0.667	0.436	0.695	**0.344**	**0.583**
		object	0.397	0.585	0.365	0.528	**0.318**	**0.533**

Consistency of predictions. In Table 5.5, we measure the proportion of inconsistent facts that were obtained when retrieving *top-k* ($k = \{1, 10\}$) predictions for the triples in the test set. We report the inconsistency values both for the prediction of the *subject* and the *object* of the triple separately. From the results, we can observe that for all models in the majority of the cases *ReasonKGE* managed to reduce the ratio of inconsistent predictions over the test sets compared to the results of training the models using *default* random and *static* sampling. This illustrates that the proposed procedure is effective for improving embeddings with respect to the overall consistency of their predictions.

SANS vs. *ReasonKGE* negative sampling. In this section, we additionally compare our *ReasonKGE* sampling technique with a recently proposed state-of-the-art sampling method SANS (structure aware negative sampling) by Ahrabian et. al [1]. SANS generates negative samples for an entity by utilizing the graph structure of the knowledge graphs. Hard negative samples are constructed for a triple from the entities in the k-hop neighbourhood of the head or tail entity that have no direct relation in the knowledge graph.

This technique requires the pre-processing step of the construction of the k-hop neighbourhood for each entity in the KG, which is a computationally intensive task. Therefore, the authors approximate the local neighbourhood with the help of n_rw random walks. Both k and n_rw are parameters that need to be optimized by manual tuning on the validation split of the KG datasets that were considered by the paper. We similarly obtain the TransE and ComplEx embedding models by training with the SANS technique on LUBM3U and Yago3-10 datasets. There are two proposed variants of SANS, the first is based on uniform sampling (Uniform SANS) while the other extends the Self-adversarial approach (Self-Adv. SANS) as proposed by Sun et al. [147].

The link prediction results for both variants with different parameter configurations are shown in Table 5.6 and compared with the *ReasonKGE*. It can be seen that *ReasonKGE* outperforms SANS for most configurations, especially in the case of ComplEx embeddings.

Table 5.6: Link prediction results with *SANS sampling* and *ReasonKGE*

Model	Sampling	LUBM3U		Yago3-10	
		MRR	*Hits@10*	*MRR*	*Hits@10*
TransE	Uniform SANS (k=3)	0.204	**0.280**	**0.412**	0.611
	Self-Adv. SANS (k=3)	**0.205**	0.278	0.409	0.581
	Uniform SANS (k=4)	0.202	0.278	0.405	0.604
	Self-Adv. SANS (k=4)	0.180	0.275	0.408	0.583
	ReasonKGE	0.135	0.256	0.367	**0.629**
ComplEx	Uniform SANS (k=3)	0.089	0.111	0.401	0.544
	Self-Adv. SANS (k=3)	0.072	0.091	0.379	0.505
	Uniform SANS (k=4)	0.088	0.111	0.396	0.545
	Self-Adv. SANS (k=4)	0.065	0.086	0.386	0.520
	ReasonKGE	**0.233**	**0.313**	**0.530**	**0.668**

5.6 Summary

This chapter has presented a method for ontology-driven negative sampling that proceeds in an iterative fashion by providing at each iteration negative samples to the embedding model on demand from its inconsistent predictions along with their generalizations. The main insight from this work is that targeted negative example generation is beneficial for training the embedding models to predict consistent facts, as witnessed by our empirical evaluation on state-of-the-art KGs equipped with ontologies. In this way, ontological knowledge can be useful for improving not only the overall link prediction performance, but more importantly, the semantic representation in the embedding model as well. Notably, the proposed *ReasonKGE* method is independent of the embedding model used, and can be exploited for improving any of the existing embedding methods. Indeed, it would be insightful to extend this work to more embedding models.

Chapter 6

Conclusion

"We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance."
- John Archibald Wheeler

In this thesis, we have explored the research problems relating to knowledge graphs, their construction and curation. We have investigated knowledge graph embeddings as models for representation of the data in KGs, and explored their shortcomings as well as semantic enhancement. This final chapter concludes the thesis with a summary and brief discussion of the overall work. This chapter also includes a discussion of the future directions in the area, particularly in the context of the research problems that were the focus of this thesis.

6.1 Summary

Knowledge graphs are the most popular repositories for structured and organized information on the real-world. While they have been acting as the backbone for numerous applications such as search, recommendation and chat bots, there are several important research gaps concerned with their construction, curation and representation that have been the central focus of this thesis.

In Chapter 1, we discussed knowledge graphs and the issues concerning automated KG construction and curation. We also included an explanation of the basics concepts for open information extraction as well as for knowledge graph embeddings that are used for KG representation. Here, we summarize the contributions of the remaining chapters comprising this thesis in terms of how they address the research questions as stated in Section 1.4.

Chapter 2 introduced the research challenges for constructing knowledge graphs, particularly in domain-oriented scenarios. We address the first research question concerning the use of Open IE techniques for the construction of a domain-specific KG and the recognition of domain-specific named entities. We considered the cultural heritage domain and described our approach to construct an art-historic KG from unstructured and noisy texts with the help of existing Open IE techniques. This chapter not only described the features of the obtained KG, but also highlighted and discussed in detail the various shortcomings that were encountered at each step during the process in Section 2.3.

Moreover, NER, being one of the first and most important steps towards the creation of KGs, was further explored and the lack of training data was identified as the main limiting factor for the poor performance. As such, an approach for generating domain and task specific annotated training data was also proposed and explained in Section 2.4. This approach was shown to be successful at significantly improving the performance of NER for artworks. Overall, the contributions of this chapter emphasize the need for recognizing the limitations of generic solutions in the face of domain-specific challenges. Our solution towards improving NER for domain oriented entities is easily extensible for other entities in the same domain as well as adaptable to various other domains.

Chapter 3 presented our solutions for the research question of refining the knowledge graphs, in particular, relating to the disambiguation of polysemous relations in KG. The inherent ambiguity in the natural language from which KG facts are extracted, coupled with the added noise and errors during the process of construction of KGs, inadvertently lead to semantic and factual mistakes in the resulting KGs. This chapter focused on the presence of polysemous relations in KGs that convey different semantics depending on the context. The motivation for finding fine-grained relation semantics is discussed in detail in the context of various use cases that depend on them. To address this gap, our proposed approach *FineGReS* is presented and explained in Section 3.5 which identifies polysemous relations and discovers the sub-relations with finer semantics by leveraging knowledge graph embeddings and performing clustering in the vector space. The benefits from this approach were demonstrated with the help of extensive experimental evaluation in terms of firstly, the quality of the clusters compared to several baselines and secondly, the positive impact on a downstream application of entity classification, as shown in Section 3.6.

Chapter 4 concentrated on the semantic representation in knowledge graph embeddings and addressed the research question on the non-uniform semantic capabilities of popular embeddings models. Embedding models have been utilized for several different semantic tasks in recent literature, including our own work for representing and distinguishing multiple semantics for KG relations. However, closer scrutiny of their semantic capability had not been performed in a systematic manner. This chapter motivated the importance of the task and presented our experiments for a quantitative evaluation of the semantics in popular embedding models. The results clearly and concretely demonstrated the limitations of the models for representing the semantics of entities in KGs beyond the most generic types such as person and organization. This study necessitates the need for a careful analysis of the properties and utility of embedding models when applying them to logical and reasoning based tasks such as rule mining.

For addressing the research question pertaining to the semantic issues in the embedding models, in Chapter 5 we presented a novel and ontology-driven approach *ReasonKGE* for generating negative samples during the training of the models. Section 5.4 presented the details of the approach that identifies semantically-inconsistent predictions made by a model with the help of an ontological reasoner and generalizes the inconsistency patterns to derive negative samples to be fed to the next iteration of the training. *ReasonKGE* method showed notable improvements for the link prediction quality of popular embedding methods, not only in terms of the higher number of correct predictions, but also for obtaining a higher ratio of semantically consistent predictions, as shown in Section 5.5. Moreover, the proposed method is agnostic to the underlying negative sampling technique or scoring function and can be leveraged for improving any existing embedding model

that can be employed for KG curation and completion.

6.2 Outlook

This thesis has advanced the state-of-the-art with respect to different research problems within the Semantic Web and Knowledge Graphs community. There are several directions for future work that have been identified by the results of this work.

With regard to the construction of domain-specific knowledge graphs, there are several open questions as well as opportunities for improvement. Due to the noisy and heterogeneous dataset that is typical of digitized art-historic collections, we encountered challenges at various steps of the KG construction process. During the very first step, it was difficult to correctly identify the mentions of artworks (i.e. titles of paintings) in the dataset due to the noise and inherent ambiguities. We proposed a method to mitigate this by generating annotated datasets for the identification of artwork titles (Section 2.4) and it would be interesting to extend our techniques for named entity recognition to other important entities in the corpus such as auctions, exhibitions and art styles. This could improve the quality as well as coverage of the resulting KG to facilitate entity-centric text exploration for cultural heritage resources. In addition, a co-reference resolution tool [31] could greatly help with the identification and linking of relevant entities. While we leveraged existing tools and libraries like SpaCy for NER, Stanford CoreNLP for triple extraction and CESI for canonicalization, these off-the-self solutions need to be further fine-tuned for domain-specific KG construction. In particular, we observed that existing techniques for canonicalization on generic datasets do not show comparable performance for domain-specific datasets. The performance is especially poor for the canonicalization of relation phrases which has been largely overlooked by the state-of-the-art methods [35, 159] in terms of a quantitative evaluation. We have been working towards addressing this research gap in terms of generating a gold standard for the evaluation [104] and plan to propose better techniques for relation canonicalization in the near future. This would facilitate the extraction of meaningful triples from the text via Open IE methods towards obtaining a KG with high quality. The scalability of the Open IE approach and the completeness of the resulting KG in the presence of new and expanding cultural heritage datasets is also an open research question to be addressed by future works.

Following KG construction, the task of KG curation encompasses the efforts for finding missing triples in existing KGs as well as alleviate issues pertaining to factual and semantic errors. We identified one such issue of semantic ambiguity in the relations of popular KGs, that had received surprisingly little attention by previous works. While the proposed *FineGReS* technique as presented in Chapter 3 identified fine-grained relation semantics and served as a first step to address this research gap, the utility and impact of this approach could be further studied with other downstream applications such as question answering and search. These applications would demonstrate the most benefit from precise and unambiguous relations in the KGs that they query. Apart from relation polysemy, there are various other quality issues in KGs resulting from inaccuracies during the construction process, such as missing entries for domain or range of relations in the ontology, incorrect facts, duplication of information etc. Though not part of this thesis, these curation tasks are challenging and warrant detailed research exploration to ensure

the quality and reliability of knowledge graphs for practical usability.

Towards knowledge graph completion, this thesis has investigated knowledge representation learning methods that aim to predict missing links in KGs by representing entities and relations as vectors. On one hand, these embeddings models have shown promising performance for link prediction, but at the same time, their semantic capabilities have not been properly understood and analysed. Our investigation of the entity semantics in popular knowledge graphs embeddings models to judge their capability for semantic tasks, as discussed in Chapter 4 is a step in this direction. There is a lot more to be done to ascertain the interpretability of the KG embeddings. Ongoing work on conceptual spaces for identifying semantically meaningful properties in the dimensions of vector spaces [21, 37] could pave the way for future research on this important topic. In the context of ensuring semantically consistent predictions from KG embedding models, our proposed method *ReasonKGE* leverages ontological reasoning to detect inconsistencies as explained in Chapter 5. Other avenues for detecting nonsensical predictions such as commonsense reasoning and language models could also prove effective and need to be further explored to determine their feasibility.

This thesis has contributed to advancing the state-of-the-art on several research problems related to knowledge graphs and their representation and curation. It is our sincere hope that the findings and insights from this work would pave the way for future research efforts on these topics.

References

[1] Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L Hamilton, and Avishek Joey Bose. Structure aware negative sampling in knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6093–6101, 2020.

[2] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics COLING 2018*, pages 1638–1649, 2018.

[4] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. Realistic Re-Evaluation of Knowledge Graph Completion Methods: An Experimental Study. In *Proceedings of the 2020 ACM International Conference on Management of Data*, SIGMOD '20, page 1995–2010, 2020.

[5] Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. Polyglot-NER: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM, 2015.

[6] Mirza Mohtashim Alam, Hajira Jabeen, Mehdi Ali, Karishma Mohiuddin, and Jens Lehmann. Affinity dependent negative sampling for knowledge graph embeddings. In *(DL4KG2020) - (ESWC 2020)*, 2020.

[7] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

[8] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. 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)*, pages 344–354, 2015.

[9] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *CoRR*, abs/1401.3487, 2014.

[10] Franz Baader, Ian Horrocks, Steffen Staab, and Rudi Studer. Description logics. In *Handbook on Ontologies (Second Edition)*, pages 21–43. Springer Dordrecht Heidelberg, 2009.

[11] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.

[12] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[13] Rajarshi Bhowmik and Gerard de Melo. Explainable Link Prediction for Emerging Entities in Knowledge Graphs. In *Proceedings of the International Semantic Web Conference*, pages 39–55, 2020.

[14] Federico Bianchi, Gaetano Rossiello, Luca Costabello, Matteo Palmonari, and Pasquale Minervini. Knowledge graph embeddings and explainable AI. In Ilaria Tiddi, Freddy Lécué, and Pascal Hitzler, editors, *KGs for XAI: Foundations, Applications and Challenges*, volume 47, pages 49–72. IOS Press, 2020.

[15] Meghyn Bienvenu. A Short Survey on Inconsistency Handling in Ontology-Mediated Query Answering. *Künstliche Intelligence*, 34(4):443–451, 2020.

[16] Stefan Bischof, Markus Krötzsch, Axel Polleres, and Sebastian Rudolph. Schema-agnostic query rewriting in SPARQL 1.1. In *Proceedings of the 2014 International Semantic Web Conference*, pages 584–600, 2014.

[17] Russa Biswas, Radina Sofronova, Mehwish Alam, and Harald Sack. Entity type prediction in knowledge graphs using embeddings. *arXiv preprint arXiv:2004.13702*, 2020.

[18] Danushka Tarupathi Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th International Conference on World wide web*, pages 151–160, 2010.

[19] Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182, 1987.

[20] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[21] Zied Bouraoui and Steven Schockaert. Learning conceptual space representations of interrelated concepts. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1760–1766, 2018.

[22] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - A knowledge graph embedding library for reproducible research. In *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, 2020.

[23] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, 2007.

[24] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1470–1480, 2018.

[25] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.

[26] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1306–1313, 2010.

[27] Valentina Anita Carriero, Aldo Gangemi, Maria Letizia Mancinelli, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti, and Chiara Veninata. ArCo: The Italian cultural heritage knowledge graph. In *Proceedings of the 18th International Semantic Web Conference*, pages 36–52. Springer, 2019.

[28] Giovanna Castellano, Giovanni Sansaro, and Gennaro Vessio. ArtGraph: Towards an Artistic Knowledge Graph. *CoRR*, 2021.

[29] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zheng-Yu Niu. Unsupervised feature selection for relation extraction. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.

[30] Weize Chen, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. Quantifying Similarity between Relations with Fact Distribution. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2882–2894, 2019.

[31] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods on Natural Language Processing*, 2016.

[32] Luciano Del Corro and Rainer Gemulla. ClausIE: Clause-based open information extraction. *Proceedings of the 22nd International Conference on World Wide Web*, 2013.

[33] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. KBQA: Learning question answering over QA corpora and knowledge bases. *arXiv preprint arXiv:1903.02419*, 2019.

[34] Sarthak Dash and Alfio Gliozzo. Distributional negative sampling for knowledge base completion. *CoRR*, abs/1908.06178, 2019.

[35] Sarthak Dash, Gaetano Rossiello, Nandana Mihindukulasooriya, Sugato Bagchi, and Alfio Gliozzo. Open knowledge graphs canonicalization using variational autoencoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10379–10394, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[36] Louise Delėger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferrė, Philippe Bessieres, and Claire Nėdellec. Overview of the bacteria biotope task at bionlp shared task 2016. In *Proceedings of the 4th BioNLP shared task workshop*, pages 12–22, 2016.

[37] Joaquin Derrac and Steven Schockaert. Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. *Artificial Intelligence*, 228:66–94, 2015.

[38] T Dettmers, P Minervini, P Stenetorp, and S Riedel. Convolutional 2D knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, volume 32, pages 1811–1818, 2018.

[39] Chris Dijkshoorn, Lizzy Jongma, Lora Aroyo, Jacco Van Ossenbruggen, Guus Schreiber, Wesley ter Weele, and Jan Wielemaker. The Rijksmuseum Collection as Linked Data. *Semantic Web*, 9(2):221–230, 2018.

[40] Kien Do, Truyen Tran, and Svetha Venkatesh. Knowledge graph embedding with multiple relation projections. In *Proceeding of the 24th International Conference on Pattern Recognition (ICPR)*, pages 332–337. IEEE, 2018.

[41] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pages 601–610, 2014.

[42] Claudia d'Amato, Nicola Flavio Quatraro, and Nicola Fanizzi. Injecting background knowledge into embedding models for predictive tasks on knowledge graphs. In *Proceedings of the European Semantic Web Conference*, pages 441–457. Springer, 2021.

[43] Maud Ehrmann, Giovanni Colavizza, Yannick Rochat, and Frédéric Kaplan. Diachronic Evaluation of NER Systems on Old Newspapers. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 97–107, 2016.

[44] Evangelos Milios Ehsan Sherkat. Vector embedding of wikipedia concepts and entities, 2017.

[45] Patrick Ernst, Amy Siu, and Gerhard Weikum. Knowlife: A versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC bioinformatics*, 16(1):1–13, 2015.

[46] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[47] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying Relations for Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[48] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

[49] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382, 1971.

[50] Nuno Freire, José Borbinha, and Pável Calado. An approach for named entity recognition in poorly structured data. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, *The Semantic Web: Research and Applications*, pages 718–732, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[51] Mohamed H. Gad-Elrab, Vinh Thinh Ho, Evgeny Levinkov, Trung-Kien Tran, and Daria Stepanova. Towards Utilizing Knowledge Graph Embedding Models for Conceptual Clustering. In *Proceedings of the International Semantic Web Conference 2020 Demos and Industry Tracks*, volume 2721, pages 281–286, 2020.

[52] Mohamed H. Gad-Elrab, Daria Stepanova, Trung-Kien Tran, Heike Adel, and Gerhard Weikum. ExCut: Explainable Embedding-Based Clustering over Knowledge Graphs. In *Proceedings of the International Semantic Web Conference*, pages 218–237, 2020.

[53] Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M Suchanek. Canonicalizing open knowledge bases. In *Proceedings of the 23rd acm international conference on conference on information and knowledge management*, pages 1679–1688, 2014.

[54] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. Predicting Completeness in Knowledge Bases. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 375–383, 2017.

[55] Dinesh Garg, Shajith Ikbal, Santosh K Srivastava, Harit Vishwakarma, Hima Karanam, and L Venkata Subramaniam. Quantum embedding of knowledge for reasoning. *Advances in Neural Information Processing Systems*, 32:5595–5605, 2019.

[56] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[57] Abbas Ghaddar and Philippe Langlais. Transforming Wikipedia into a Large-Scale Fine-Grained Entity Type Corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[58] Abbas Ghaddar and Phillippe Langlais. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, 2017.

[59] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[60] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[61] Birte Glimm, Yevgeny Kazakov, Thorsten Liebig, Trung-Kien Tran, and Vincent Vialard. Abstraction refinement for ontology materialization. In *Proceedings of the International Semantic Web Conference*, pages 180–195. Springer, 2014.

[62] Birte Glimm, Yevgeny Kazakov, and Trung-Kien Tran. Ontology Materialization by Abstraction Refinement in Horn SHOIF. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1114–1120, 2017.

[63] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics*, 3(2-3):158–182, 2005.

[64] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. OpenKE: An Open Toolkit for Knowledge Embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[65] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1709–1719, 2019.

[66] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, 2004.

[67] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

[68] Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology, 2005.

[69] Vinh Thinh Ho, Daria Stepanova, Mohamed H Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. Rule learning from knowledge graphs guided by embedding models. In *Proceedings of the International Semantic Web Conference*, pages 72–90, 2018.

[70] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[71] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Explaining inconsistencies in owl ontologies. In *Proceedings of the International Conference on Scalable Uncertainty Management*, pages 124–137. Springer, 2009.

[72] Shanshan Huang and Xiaojun Wan. AKMiner: Domain-specific knowledge graph mining from academic literatures. In *Proceedings of the International Conference on Web Information Systems Engineering*, pages 241–255. Springer, 2013.

[73] Jane Hunter and Suleiman Odat. Building a Semantic Knowledge-base for Painting Conservators. In *Proceedings of the 2011 IEEE Seventh International Conference on eScience*, pages 173–180, 2011.

[74] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. Are Meta-Paths Necessary? Revisiting Heterogeneous Graph Embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 437–446, 2018.

[75] Nitisha Jain. Domain-Specific Knowledge Graph Construction for Semantic Analysis. In *Proceedings of the Extended Semantic Web Conference (ESWC) 2020 Satellite Events*, pages 250–260, Cham, 2020. Springer International Publishing.

[76] Nitisha Jain, Jan-Christoph Kalo, Wolf-Tilo Balke, and Ralf Krestel. Do embeddings actually capture knowledge graph semantics? In *Proceedings of the European Semantic Web Conference*, pages 143–159. Springer, 2021.

[77] Nitisha Jain and Ralf Krestel. Learning Fine-Grained Semantics for Multi-Relational Data. In *Proceedings of the International Semantic Web Conference 2020, Posters and Demos*, 2020.

[78] Nitisha Jain and Ralf Krestel. Discovering Fine-Grained Semantics in Knowledge Graph Relations. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, page 822–831, New York, NY, USA, 2022. Association for Computing Machinery.

[79] Nitisha Jain, Alejandro Sierra, Jan Ehmueller, and Ralf Krestel. Generation of Training Data for Named Entity Recognition of Artworks. *Semantic Web*, 2022.

[80] Nitisha Jain, Alejandro Sierra-Múnera, Maria Lomaeva, Julius Streit, Simon Thormeyer, Philipp Schmidt, and Ralf Krestel. Generating Domain-Specific Knowledge Graphs: Challenges with Open Information Extraction. In *Proceedings of the European Semantic Web Conference, Text2KG workshop.*, 2022.

[81] Nitisha Jain, Trung-Kien Tran, Mohamed H Gad-Elrab, and Daria Stepanova. Improving Knowledge Graph Embeddings with Ontological Reasoning. In *Proceedings of the International Semantic Web Conference*, pages 410–426. Springer, 2021.

[82] Prachi Jain, Pankaj Kumar, Soumen Chakrabarti, et al. Type-sensitive knowledge base inference without explicit type supervision. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 75–80, 2018.

[83] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 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)*, pages 687–696, 2015.

[84] Zhengbao Jiang, Jun Araki, Donghan Yu, Ruohong Zhang, Wei Xu, Yiming Yang, and Graham Neubig. Learning Relation Entailment with Structured and Textual Information. In *Automated Knowledge Base Construction*, 2020.

[85] Jan-Christoph Kalo, Philipp Ehler, and Wolf-Tilo Balke. Knowledge graph consolidation by unifying synonymous relationships. In *Int. Semantic Web Conf.*, pages 276–292, 2019.

[86] Mayank Kejriwal. *Domain-specific knowledge graph construction*. Springer, 2019.

[87] Kimmo Kettunen and Teemu Ruokolainen. Names, Right or Wrong: Named Entities in an OCRed Historical Finnish Newspaper Collection. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, pages 181–186, 2017.

[88] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 70–75. Citeseer, 2004.

[89] Tomáš Kliegr and Ondřej Zamazal. LHD 2.0: A text mining approach to typing entities in knowledge graphs. *Journal of Web Semantics*, 39:47–61, 2016.

[90] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, et al. DrugBank 3.0: A comprehensive resource for 'omics' research on drugs. *Nucleic acids research*, 39(suppl_1):D1035–D1041, 2010.

[91] Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761, Online, November 2020. Association for Computational Linguistics.

[92] Bhushan Kotnis and Vivi Nastase. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *CoRR*, abs/1708.06816, 2017.

[93] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):S2, 2015.

[94] Klaus Krippendorff. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70, 1970.

[95] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *Proceedings of the International Semantic Web Conference*, pages 640–655, 2015.

[96] Ora Lassila and Ralph R Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999.

[97] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[98] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics*, 33:3–29, 2015.

[99] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[100] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. SVM based learning system for information extraction. In *International Workshop on Deterministic and Statistical Methods in Machine Learning*, pages 319–339. Springer, 2004.

[101] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[102] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[103] Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S Rosenblum. MMKG: multi-modal knowledge graphs. In *Proceedings of the European Semantic Web Conference*, pages 459–474, 2019.

[104] Maria Lomaeva and Nitisha Jain. Relation Canonicalization in Open Knowledge Graphs: A Quantitative Analysis. In *Proceedings of the European Semantic Web Conference*, pages 21–25. Springer, 2022.

[105] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2):72–79, 2001.

[106] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research*, 2014.

[107] Robert Malouf. Markov models for language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL)*, 2002.

[108] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[109] George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995.

[110] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1027–1037, 2012.

[111] P Minervini, T Demeester, T Rocktäschel, and S Riedel. Adversarial sets for regularising neural link predictors. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*. Curran Associates Inc, 2017.

[112] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

[113] Thahir Mohamed, Estevam Hruschka, and Tom Mitchell. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1447–1455, 2011.

[114] Changsung Moon, Paul Jones, and Nagiza F Samatova. Learning entity type embeddings for knowledge graph completion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2215–2218, 2017.

[115] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. OWL 2 Web Ontology Language Direct Semantics (Second Edition). Technical report, 2012.

[116] Arvind Neelakantan and Ming-Wei Chang. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, 2015.

[117] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

[118] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[119] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-way Model for Collective Learning on Multi-relational Data. In *Proceedings of the International Conference on Machine Learning*, ICML'11, pages 809–816, 2011.

[120] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 271–280, 2012.

[121] Joel Nothman. Learning named entity recognition from wikipedia. *Honours Bachelor thesis, The University of Sydney Australia*, 2008.

[122] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175, 2013.

[123] Dominic Oldman and CRM Labs. The CIDOC Conceptual Reference Model (CIDOC-CRM): PRIMER. *CIDOC-CRM official web site*, 2014.

[124] Sergio Oramas, Luis Espinosa-Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. Information extraction for knowledge base construction in the music domain. *Data and Knowledge Engineering*, 106:70–83, 2016.

[125] Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971, 2015.

[126] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[127] Heiko Paulheim and Christian Bizer. Type inference on noisy rdf data. In *Int. Semantic Web Conf.*, pages 510–525, 2013.

[128] Heiko Paulheim and Aldo Gangemi. Serving DBpedia with DOLCE – More than Just Adding a Cherry on Top. In *Proceedings of the 2015 International Semantic Web Conference*, pages 180–196. Springer, 2015.

[129] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[130] Thierry Poibeau and Leila Kosseim. Proper name extraction from non-journalistic texts. *Language and computers*, 37:144–157, 2001.

[131] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the 17th Conference on Computational Natural Language Learning*, pages 143–152, 2013.

[132] Roman Prokofyev, Gianluca Demartini, and Philippe Cudré-Mauroux. Effective named entity recognition for idiosyncratic web collections. In *Proceedings of the 23rd international conference on World Wide Web (WWW)*, pages 397–408. ACM, 2014.

[133] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[134] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 2017.

[135] Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.

[136] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, 2013.

[137] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *Proceedings of the International Semantic Web Conference*, pages 498–514, 2016.

[138] Kepa Joseba Rodriquez, Mike Bryant, Tobias Blanke, and Magdalena Luszczynska. Comparison of Named Entity Recognition Tools for Raw OCR Text. In *Proceedings of the Conference on Natural Language Processing 2012*, pages 410–414. Austrian Society for Artificial Intelligence (ÖGAI), 2012.

[139] Andrea Rossi and Antonio Matinata. Knowledge Graph Embeddings: Are Relation-Learning Models Learning Relations? In *EDBT/ICDT Workshops*, 2020.

[140] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN Teach an Old Dog New Tricks! On training knowledge graph embeddings. In *Proceedings of the International Conference on Learning Representations*, 2020.

[141] Roxane Segers, Marieke Van Erp, Lourens Van Der Meij, Lora Aroyo, Guus Schreiber, Bob Wielinga, Jacco van Ossenbruggen, Johan Oomen, and Geertje Jacobs. Hacking history: Automatic historical event extraction for enriching cultural heritage multimedia collections. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP)*, pages 26–29, 2011.

[142] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. Incremental Knowledge Base Construction using Deepdive. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, volume 8, page 1310, 2015.

[143] Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311, 2006.

[144] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[145] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, 2007.

[146] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proc. VLDB Endow.*, 13(12):2326–2340, July 2020.

[147] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*, 2018.

[148] Tabea Tietz, Jörg Waitelonis, Kanran Zhou, Paul Felgentreff, Nils Meyer, Andreas Weber, and Harald Sack. Linked Stage Graph. In *SEMANTICS Posters&Demos*, 2019.

[149] Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, page 1–4, USA, 2002. Association for Computational Linguistics.

[150] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, page 142–147, USA, 2003. Association for Computational Linguistics.

[151] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.

[152] Trung-Kien Tran, Mohamed H Gad-Elrab, Daria Stepanova, Evgeny Kharlamov, and Jannik Strötgen. Fast computation of explanations for inconsistency in large-scale knowledge graphs. In *Proceedings of The Web Conference 2020*, pages 2613–2619, 2020.

[153] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 2071–2080, 2016.

[154] Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. Cross-lingual named entity recognition via Wikification. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 219–228, 2016.

[155] Özlem Uzuner, Yuan Luo, and Peter Szolovits. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563, 2007.

[156] Seth Van Hooland, Max De Wilde, Ruben Verborgh, Thomas Steiner, and Rik Van de Walle. Exploring Entity Recognition and Disambiguation for Cultural Heritage Collections. *Digital Scholarship in the Humanities*, 30(2):262–279, 2013.

[157] Seth Van Hooland and Ruben Verborgh. *Linked Data for Libraries, Archives and Museums: How to clean, link and publish your metadata.* Facet publishing, 2014.

[158] Paroma Varma and Christopher Ré. Snuba: Automating weak supervision to label training data. *Proceedings of the VLDB Endowment*, 12(3):223–236, 2018.

[159] Shikhar Vashishth, Prince Jain, and Partha Talukdar. CESI: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference*, pages 1317–1327, 2018.

[160] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85, 2014.

[161] Oleksandra Vsesviatska, Tabea Tietz, Fabian Hoppe, Mirjam Sprau, Nils Meyer, Danilo Dessì, and Harald Sack. ArDO: An ontology to describe the dynamics of multimedia archival records. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1855–1863, 2021.

[162] Thuy Vu and D Stott Parker. K-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1262–1267, 2016.

[163] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.

[164] PeiFeng Wang, Shuangyin Li, and Rong Pan. Incorporating GAN for negative sampling in knowledge representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2005–2012, 2018.

[165] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[166] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[167] Koki Washio and Tsuneaki Kato. Neural latent relational analysis to capture lexical semantic relations in a vector space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 594–600, 2018.

[168] Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. In *Foundations and Trends in Databases*, 2021.

[169] Kemas Wiharja, Jeff Z. Pan, Martin J. Kollingbaum, and Yu Deng. Schema aware iterative knowledge graph completion. *Journal of Web Semantics*, 65:100616, 2020.

[170] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[171] Han Wu, Shuo Yan Liu, Wenkai Zheng, Yifu Yang, and Han Gao. PaintKG: The painting knowledge graph using biLSTM-CRF. In *Proceedings of the 2020 International Conference on Information Science and Education (ICISE-IE)*, pages 412–417, 2020.

[172] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics.

[173] Han Xiao, Minlie Huang, and Xiaoyan Zhu. TransG: A Generative Model for Knowledge Graph Embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325, 2016.

[174] Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, 2017.

[175] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, 2020.

[176] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, May 2015.

[177] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: Open Information Extraction on the Web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007.

[178] Jianbo Yuan, Zhiwei Jin, Han Guo, Hongxia Jin, Xianchao Zhang, Tristram Smith, and Jiebo Luo. Constructing biomedical domain-specific knowledge graph with minimum supervision. *Knowledge and Information Systems*, 62(1):317–336, 2020.

[179] Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. Neural-symbolic reasoning on knowledge graphs. *CoRR*, abs/2010.05446, 2020.

[180] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. In *Proceedings of the 2019 World Wide Web Conference*, WWW '19, page 2366–2377, 2019.

[181] Ye Zhang and Byron C Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, 2017.

[182] Yongqi Zhang, Quanming Yao, and Lei Chen. Efficient, simple and automated negative sampling for knowledge graph embedding. *CoRR*, abs/2010.14227, 2020.

[183] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. NSCaching: Simple and efficient negative sampling for knowledge graph embedding. In *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE)*, pages 614–625. IEEE, 2019.

[184] Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3198–3207, 2018.

[185] Xuejiao Zhao, Zhenchang Xing, Muhammad Ashad Kabir, Naoya Sawada, Jing Li, and Shang-Wei Lin. HDSKG: Harvesting domain specific knowledge graph from content of webpages. In *Proceedings of the 24th International Conference on Software Analysis, Evolution and Re-engineering (SANER)*, pages 56–67. IEEE, 2017.

[186] Yu Zhao, Ruobing Xie, Kang Liu, WANG Xiaojie, et al. Connecting Embeddings for Knowledge Graph Entity Typing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6419–6428, 2020.

[187] GuoDong Zhou and Jian Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics, 2002.

[188] Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. A Survey on Neural Open Information Extraction: Current Status and Future Directions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence(IJCAI-22) Survey Track*, 2022.

[189] Konstantin Ziegler, Olivier Caelen, Mathieu Garchery, Michael Granitzer, Liyun He-Guelton, Johannes Jurgovsky, Pierre-Edouard Portier, and Stefan Zwicklbauer. Injecting semantic background knowledge into neural networks using graph embeddings. In *Proceedings of the IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 200–205. IEEE, 2017.

Declaration

I hereby confirm that

- this dissertation with the topic **Representation and Curation of Knowledge Graphs with Embeddings** is the result of my own work, it was prepared without unauthorized help and using only the given literature,
- this dissertation has not been previously submitted, in part or whole, to any other university,
- I am aware of the doctorate regulations of the Digital Engineering Faculty of the University of Potsdam from November 27, 2019.

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift mit dem Thema **Representation and Curation of Knowledge Graphs with Embeddings** selbständig und ohne unerlaubte Hilfe angefertigt sowie nur die angegebene Literatur verwendet habe,
- die Dissertation keiner anderen Hochschule in gleicher oder ähnlicher Form vorgelegt wurde,
- mir die Promotionsordnung der Digital Engineering Fakultät der Universität Potsdam vom 27. November 2019 bekannt ist.

Potsdam, October 19, 2022

Nitisha Jain