Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

CUMULATIVE DISSERTATION

Knowledge Graph-Guided Information Extraction

Cedric Möller

Semantic Systems

Department of Informatics

Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg
Hamburg, Germany

A thesis submitted for the degree of

Doctor rerum naturalium (Dr. rer. nat.)

2025

Knowledge Graph-Guided Information Extraction

Dissertation submitted by: Cedric Möller

Date of Submission: 26.08.2025
Date of Disputation: 28.01.2026

Supervisor(s):

Prof. Dr. Ricardo Usbeck, Leuphana Universität Lüneburg

Committee:

1st Examiner: Prof. Dr. Ricardo Usbeck, Leuphana Universität Lüneburg
2nd Examiner: Prof. Dr. Janick Edinger, Universität Hamburg
3rd Examiner: Prof. Dr. Sören Auer, Gottfried Wilhelm Leibniz Universität Hannover

Universität Hamburg, Hamburg, Germany
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Semantic Systems

Affidavit

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

I hereby declare in lieu of an oath that I have written this dissertation myself and have not used any sources and aids other than those specified. If electronic services based on generative artificial intelligence (gAI) were used in the course of writing this dissertation, I confirm that my own work was the main focus and that all resources used are fully documented in accordance with good scientific practice. I am responsible for any erroneous or distorted content, incorrect references, violations of data protection and copyright law or plagiarism that may have been generated by the gAI.

26.08.2025

Date

Signature

(Cedric Möller)

Acknowledgements

I especially thank my mentor, Ricardo, for his ongoing support and invaluable feedback over the years.

My sincere thanks go to my colleagues Angelie, Junbo, Xixi, Debayan, Tilahun, and many others for their thoughtful conversations and the productive years we shared.

I want to thank Julia for her unwavering support throughout the development of this thesis. I am also thankful to my brothers, David and Tim, as well as my parents, Ralf and Antje, for their ongoing interest in and encouragement of my work.

Finally, I must thank my beloved cats, Nüschel and Müffel, for always being there to cuddle whenever I needed comfort.

Use of Third-Party Software

For correcting spelling, grammar, and general writing improvement, I relied on Grammarly, an AI-based tool, as well as a University of Hamburg available ChatGPT model. I relied on the ShareLaTeX instance provided by the Gesellschaft für Wissenschaftliche Datenverarbeitung mbH Göttingen for writing the thesis in LaTeX. The figures used in this thesis were created using Draw.io and Inkscape.

Abstract

A knowledge graph (KG) is a structured representation of facts where entities (such as people, places, or organizations) are connected by predefined relationships (like `birthplace` or `occupation`). The main goal of this thesis is to explore how incorporating knowledge graph information can improve information extraction methods for populating KGs with new facts.

Knowledge graph population involves identifying factual triples, each consisting of subject, relation and object (e.g., ⟨`Einstein`, `birthplace`, `Ulm`⟩), within text and enriching an existing KG with the extracted information. We focus on two essential steps of knowledge graph population: entity linking and relation extraction.

Entity linking matches entities mentioned in text to the ones in the KG, for example, linking *Apple* to the technology company rather than the fruit.

Relation extraction identifies relations expressed between two entities in text, such as recognizing that *Steve Jobs founded Apple* indicates a `founderOf` relationship.

First, we review current entity linking methods and how they incorporate KG information, demonstrating that such information remains underutilized despite its potential benefits.

Building on this, we investigate the impact of KG embeddings on entity linking for entities both inside and outside the KG, showing an impact on entities inside but not outside.

Next, we evaluate the role of KG information in relation extraction across texts of varying lengths, from single sentences to full paragraphs, and in both fully-supervised (with several labeled examples) and zero-shot (no labeled examples) settings.

We first incorporate KG information into relation extraction by integrating entity types (e.g., *person*, *shipyard*, *movie*) in textual and vectorized forms, both of which show significant positive effects.

Finally, we examine the impact of structural KG information by including paths, sequences of triples connecting two entities, which notably improve relation extraction, especially in zero-shot scenarios.

Overall, we assert that incorporating knowledge graph information can significantly influence the performance of information extraction methods, with the degree of impact depending on the type of text, availability of data and the kind of information used.

Zusammenfassung

Ein Knowledge Graph (KG) ist eine strukturierte Repräsentation von Fakten, bei der Entitäten (wie Personen, Orte oder Organisationen) durch vordefinierte Relationen (wie `birthplace` oder `occupation`) miteinander verbunden sind. Das Hauptziel dieser Arbeit ist es, zu untersuchen, inwieweit die Einbindung von Knowledge-Graph-Informationen die Leistung von Informationsextraktionsmethoden verbessert.

Das Anreichern eines Knowledge Graphs mit Informationen, auch KG-Population genannt, beinhaltet das Erkennen faktischer Tripel, die jeweils aus Subjekt, Relation und Objekt bestehen (z.B. ⟨`Einstein`, `birthplace`, `Ulm`⟩), innerhalb von Texten sowie das Hinzufügen dieser zu einem bestehenden KG. Wir konzentrieren uns dabei auf zwei zentrale Schritte von KG-Population: Entity Linking und Relation Extraction.

Entity Linking ordnet im Text erwähnte Entitäten den konkreten Entitäten im KG zu. Zum Beispiel wird *Apple* mit dem Technologieunternehmen statt mit der Frucht verknüpft.

Relation Extraction identifiziert Relationen, die zwischen zwei Entitäten im Text ausgedrückt werden, wie etwa das Erkennen, dass *Steve Jobs gründete Apple* auf eine `founderOf`-Beziehung hinweist.

Zunächst geben wir einen Überblick über aktuelle Entity-Linking-Methoden und analysieren, inwiefern sie KG-Informationen einbinden. Dabei zeigen wir, dass solche Informationen trotz ihres Potenzials bislang nur teilweise genutzt werden.

Darauf aufbauend untersuchen wir den Einfluss von KG-Embeddings auf das Entity Linking, sowohl für Entitäten, die im KG enthalten sind, als auch für solche außerhalb, und zeigen, dass ein Einfluss auf Entitäten innerhalb, jedoch nicht außerhalb besteht.

Anschließend evaluieren wir die Rolle von KG-Informationen bei der Relation Extraction über Texte unterschiedlicher Länge hinweg, von einzelnen Sätzen bis hin zu ganzen Absätzen, sowie in verschiedenen Szenarien: sowohl im fully-supervised (mit mehreren gelabelten Beispielen) als auch im zero-shot (ohne gelabelte Beispiele) Szenario.

Zuerst integrieren wir KG-Informationen in der Relation Extraction durch die Einbindung von Entitätstypen (z.B. *Person*, *Werft*, *Film*) in textueller sowie in vektorisierter Form, wobei beide Varianten signifikant positive Effekte zeigen.

Abschließend untersuchen wir den Einfluss struktureller KG-Informationen, indem wir Pfade, Sequenzen von Tripeln, die zwei Entitäten miteinander verbinden,

einbeziehen. Diese verbessern die Relation Extraction deutlich, insbesondere in Zero-Shot-Szenarien.

Insgesamt stellen wir fest, dass die Einbeziehung von Knowledge-Graph-Informationen die Leistung von Informationsextraktionsmethoden erheblich verbessern kann, wobei das Ausmaß des Einflusses von der Art des Textes, der Verfügbarkeit von Daten und der Art der verwendeten Informationen abhängt.

Contents

List of Figures

List of Tables

List of Abbreviations

ED	Entity Disambiguation
EL	Entity Linking
GNN	Graph Neural Network
KG	Knowledge Graph
KGC	Knowledge Graph Construction
KGE	Knowledge Graph Embedding
KGP	Knowledge Graph Population
LLM	Large Language Model
LM	Language Model
MD	Mention Detection
NER	Named Entity Recognition
OWL	Web Ontology Language
PLM	Pre-trained LM
RE	Relation Extraction
RDF	Resource Description Framework
RDFS	RDF Schema

1 Introduction

1.1 Motivation

A knowledge graph (KG) is "a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities" (Hogan et al., 2021, p. 2) (see Figure 1.1 for an example). It can be viewed as a specialized type of database designed to store and manage facts across a diverse range of domains in the form of a graph. These domains encompass common-sense knowledge (Ilievski et al., 2021), such as the typical shape of an apple, to more general information about countries or movies (Vrandečić and Krötzsch, 2014), and extend to highly specialized areas detailing facts about genes and their interactions (Chandak et al., 2023; The Gene Ontology Consortium et al., 2023). In KGs, facts are commonly represented as triples, each comprising a subject entity, a relation, and an object entity.

The task of populating these KGs with accurate information—i.e., identifying entities, extracting relationships between them, and incorporating this structured information into the graph—can be labor-intensive and challenging (Zhong et al., 2024). Some KGs are manually curated by a large network of contributors, which often ensures high data quality but incurs significant time and financial costs (Vrandečić and Krötzsch, 2014). Conversely, automated web scraping methods can rapidly collect data but might overlook nuances or produce incomplete datasets, as they depend on existing web content that may be error-prone or lacking in depth (Hogan et al., 2021).

The emergence of pre-trained language models (PLMs) (Devlin et al., 2019) has dramatically improved the potential for automatic extraction and integration of facts into KGs. These models, leveraging deep neural networks, enhance the precision and scope of information extraction, offering greater efficiency and scalability than previous methodologies. The advent of generative large language

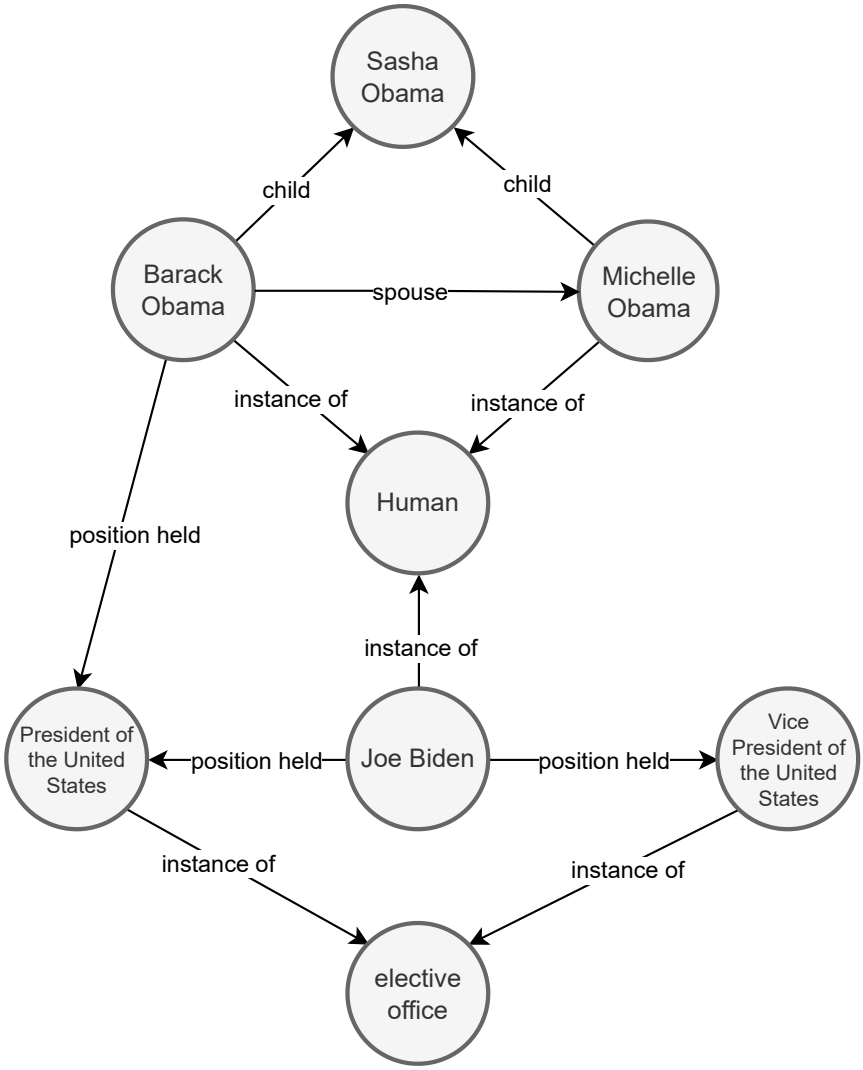


Figure 1.1: Example KG depicting various entities in connection to Barack Obama.

models (LLMs) (Zhao, Zhou, et al., 2023) has further simplified the generation of facts from text, making this process more accessible and robust.

In this thesis, we are particularly interested in **how information from an existing KG can be used to guide and improve automatic information extraction tasks, specifically entity linking and relation extraction**. Importantly, **we do not aim to construct a full knowledge graph from scratch** as is done in knowledge graph construction (Das et al., 2024; Heist, 2024; Prabhong et al., 2024; Zhang and Soh, 2024; Zhang, Cao, Wang, et al., 2024).

To better understand how KG information can support the task of entity linking (EL), which is vital for populating a KG, we first examine the major challenges this task presents. The aim in EL is to link entities mentioned in the text to their corresponding entity as existing in a KG. A difficulty is here the ambiguity between

different entities. For example, *War of the Worlds* could refer to the book, several movies or TV-shows. A significant challenge connected to this lies in managing entities that have not yet been or are newly added to a KG (Hoffart et al., 2014). In our dynamic world, new entities are constantly emerging. For instance, new individuals are born every day, new books and articles are published, and new scientific discoveries are made. Integrating these new entities along with their information into existing KGs is vital for keeping the relevance and accuracy of KGs.

In regard to the task of EL, we can differentiate between two major groups of entities: 1) entities that are already present in the KG (in-KG entities), and 2) entities that are not yet part of the KG (out-of-KG entities[1]). Only the first group—entities within the KG—can be further subdivided into 1a) entities available during the training phase of a method, and 1b) entities added to the KG after training. Training refers to the phase where an EL method's parameters are adjusted using a dataset to perform the corresponding task. See Figure 1.2 for a comparison of the different types of entities in the context of EL.

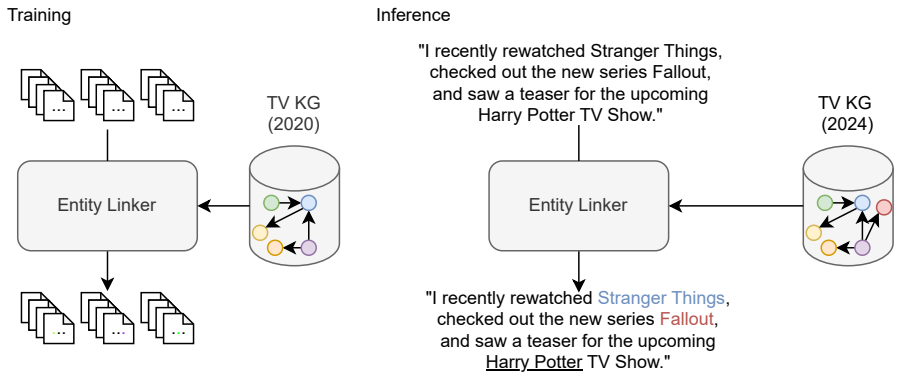


Figure 1.2: The figure illustrates an entity linker during both training and inference. During training, a knowledge graph (KG) focused on television shows, referred to as the TV KG (2020), is used. During inference, the input text mentions three entities: Stranger Things, Fallout, and Harry Potter. At this stage, the updated TV KG from 2024 is employed. Among the three, Stranger Things, in group 1a), already existed in the KG during training (blue node). Fallout, in group 1b), was added after training and is now present in the KG (red node). In contrast, the Harry Potter TV show, in group 2), does not exist in the 2024 KG, which only includes released shows. As a result, Stranger Things and Fallout can be linked to KG entries, whereas Harry Potter must be identified as not yet present in the KG.

EL methods supporting entities beyond group 1a) need to overcome different challenges. First, to support entities of group 1b), methods need to be developed that can generalize to entirely new entities added to an existing KG. In machine learning, a model's ability to generalize to unseen labels or outputs is referred to as zero-shot learning. Unlike fully-supervised learning, where models are trained on labeled examples that include all possible output labels, zero-shot settings involve

1. Note that such entities are also sometimes called NIL entities (Ilievski, 2019).

encountering new, previously unseen labels.[2] See Figure 1.3 for an illustration. Specifically, in EL, zero-shot generalizability refers to a model's capacity to link to completely new entities that it has not seen during training.

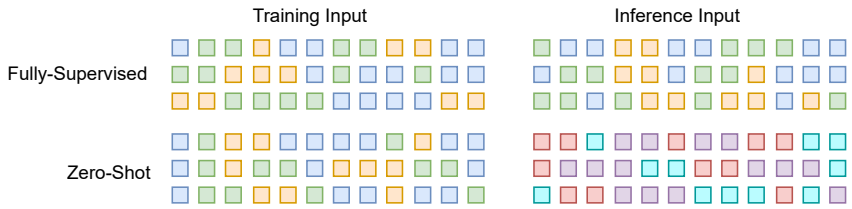


Figure 1.3: Zero-shot vs. Fully-Supervised learning: The labels (indicated by their colors) of the input examples during training and inference are the same for fully-supervised learning, while they are different for zero-shot learning.

Secondly, to support entities of group 2), another vital aspect is a model's ability to determine whether newly encountered entities in text already exist within a KG. This includes the ability to discern gaps in the graph where new entities should be placed, as well as identifying whether the information in the text matches existing entities. Finally, as a new entity might occur repeatedly, connecting multiple instances of the same new entity becomes crucial. This is usually solved by clustering all mentions of new entities globally (Agarwal et al., 2022; Heist and Paulheim, 2023), where clustering refers to the process of assigning multiple mentions of entities to each other based on their similarity. Since entities of group 2) are not encountered during training as well, an EL model supporting them must also have zero-shot capabilities. This thesis investigates whether incorporating KG information into the EL method is beneficial, especially with a focus on out-of-KG entities.

Another core information extraction task explored in this thesis is relation extraction (RE), where we aim to assess the impact of introducing information from a KG when extracting relations from text. The specific goal is to identify the relations expressed between pairs of entities. This is accompanied by specific challenges as well.

The use of relations can vary significantly from one KG to another, making it essential to accurately identify which relations are relevant for extraction. For example, a relation like `country` might only apply to associations between cities and countries within one KG. At the same time, in another KG, it might only relate to associations between persons and countries. This contextual discrepancy underscores the necessity of understanding each KG's specific schema to extract valid information from text.

Moreover, certain relations may be used less frequently than others, adding a layer of complexity to their identification. Rare relations are also referred to as long-tail relations, as the frequency distribution of all relations typically exhibits a *long*

2. Note that in zero-shot learning, supervision is still taking place during training. The model is trained with labeled data, just not for all of the labels it may encounter at inference time (Xian et al., 2019).

tail. This tail represents the many relations that occur infrequently, in contrast to the few common relations that appear frequently and dominate the head of the distribution. For example, every person has parents, but unfortunately, not every person has won an award. Therefore, the latter is more rarely encountered in the KG as well as in text. This is especially true when we encounter completely new relations (such as in a zero-shot setting) or when many relations exist, resulting in a higher level of ambiguity. If relations are rarer, this results in less training data, which affects model performance (Goodfellow et al., 2016; Kaplan et al., 2020).

In this thesis, we will focus on several of these problems, with a specific emphasis on how incorporating information from KGs can support and improve both EL and RE. In Chapter 2, we will introduce all the necessary knowledge to provide a foundation for the rest of the research. In Chapter 3 we have a look at the KG feature utilization of recent EL methods working on the popular open-domain KG Wikidata (Vrandečić and Krötzsch, 2014). By KG features, we are referring to any specific type of information from the knowledge graph, such as entity types, descriptions, or graph connections. Additionally, we also analyze all EL datasets connected to Wikidata. Finally, in Chapter 4, we will look at the impact of KG information on the identification of out-of-KG entities and their clustering. In Chapter 5, Chapter 6, and Chapter 7, we explore the impact of leveraging KG information in both large-scale and zero-shot RE settings. By examining how ambiguous and long-tail relations can be effectively extracted, we aim to enhance the methodologies for handling RE in diverse settings.

1.2 Research Questions

Based on the key challenges outlined in the previous section, we investigate five research questions across four critical aspects: (RQ1) the use of KGs in existing EL methods; (RQ2) the impact of KG information on out-of-KG EL; (RQ3 and RQ4) the role of fine-grained entity types in the task of RE; (RQ5) and the effect of including structural KG information into RE.

Our initial focus is on understanding how well recent Wikidata-based EL methods utilize the rich information within KGs. This question is essential in determining if current methodologies effectively tap into available data for tasks like EL, especially under challenging conditions where typical resources might be absent. We focus on the Wikidata KG because it is one of the largest publicly available knowledge graphs (Suchanek et al., 2024) and is widely used in research.[3]

Research Question 1

How do recent Wikidata-based entity linking methods leverage knowledge graph information?

3. Google Scholar lists "Wikidata: a free collaborative knowledge base" (Vrandečić and Krötzsch, 2014) as being cited 5,003 times as of July 8, 2025.

Building upon the first question, our second inquiry delves into the specific scenario of recognizing and handling out-of-KG entities. Here, we explore the potential advantages that a KG's inherent information might offer to this task.

Research Question 2

How does incorporating knowledge graph information affect the performance of entity linking methods on out-of-KG entities?

Shifting focus towards RE, the third question investigates the role of fine-grained entity-type information in zero-shot settings. This question is crucial for understanding the adaptability of current systems when confronted with entirely novel relation types, thus probing the versatility of existing models against unseen data.

Research Question 3

What effect does the inclusion of fine-grained entity types have on the task of zero-shot relation extraction?

Furthermore, we extend our examination of fine-grained entity information to discern how its inclusion influences performance in settings characterized by numerous relations.

Research Question 4

How does the utilization of fine-grained entity-type information affect the performance of relation extraction methods with numerous relations?

Finally, we extend our analysis to the overall impact of graph information. We investigate the effect of incorporating the path-wise information between entities in the fully-supervised and zero-shot settings of RE.

Research Question 5

What impact does the inclusion of structural information from the knowledge graph have in zero-shot and fully-supervised relation extraction settings?

1.3 Contributions

In this section, we specify the contributions of each of the contained papers to answering those research questions:

Research Question 1: In Chapter 3, we conduct a survey by gathering and analyzing all peer-reviewed EL papers, comprising method and dataset papers, published until 2021 that focused on the Wikidata KG. We categorize these papers regarding their usage of the Wikidata features such as type information, entity descriptions and the actual underlying graph, among others. Furthermore, we

analyze the potential for zero-shot EL usage. Our findings reveal that most methods make limited use of graph information.

Research Question 2: In Chapter 4, we examine the impact of KG information on the task of EL when dealing with out-of-KG entities. To accomplish that, we integrate KG embeddings into the EL process. To evaluate performance, we adapt a widely used EL dataset (Hoffart et al., 2014) by splitting the set of entities into two sets: some are present in the knowledge graph (KG), and others are not. In addition, we construct a new dataset derived from the "Current events" section of Wikipedia[4], denoted *Wikievents*. The texts are divided based on their publication date: before or after the beginning of 2019. We then use a Wikidata KG snapshot from 2019, ensuring that the texts published after that year include entities not present in the KG. Our results on those datasets demonstrate that while these embeddings enhanced the EL of in-KG entities, they did not affect the performance of out-of-KG EL.

Research Question 3: In Chapter 5, we investigate the impact of fine-grained entity types in the task of zero-shot RE by introducing the labels of the involved entity types into the RE method. Our findings show that including them in the model leads to significant improvements, especially when the difficulty of zero-shot RE increases.

Research Question 4: In Chapter 6, we analyse the impact of fine-grained entity types in the setting of closed information extraction, where EL and RE are performed together. We particularly examine scenarios involving a large number of potential relations. We show that by incorporating the fine-grained type information as vectors into the RE, the closed information extraction performance increases substantially. These improvements are connected to an enhanced ability to extract long-tail relations.

Research Question 5: In Chapter 7, we investigate the impact of structural KG information - specifically, in the form of paths between entities inside the graph - on the task of RE by incorporating it through the use of graph neural networks. Our findings show that including them in the model leads to significant improvements in the zero-shot setting while having a positive impact in the supervised setting as well.

1.4 Related Work

This section summarizes prior work related to the task of information extraction, emphasizing the inclusion of KG information. We focus in particular on two key subproblems: EL and RE.

4. https://en.wikipedia.org/wiki/Portal:Current_events

1.4.1 Entity Linking

To gain an overview of how KG information is used for EL, we conduct a survey. While many surveys on EL existed prior to our work, most focused on different architectures (Sevgili et al., 2019), on specific domains (French and McInnes, 2023) or were outdated, not considering modern deep learning methods (Shen et al., 2015). Furthermore, the KG features used by different deep learning EL models have not yet been investigated. To get a comprehensive overview, we analyze all methods focusing on the Wikidata KG regarding their KG feature utilization, specifically which elements in the KG the methods consider during EL (see Chapter 3).

Early EL systems relied on syntactic string similarity between entity names in the KG and the entity mentions in the text (Ferragina and Scaiella, 2012; Hoffart et al., 2011). Later models utilized graph information (Moussallem et al., 2017; Usbeck, Ngonga Ngomo, et al., 2014) but still struggled with semantic variability due to their reliance on syntactic similarities (e.g., matching *scientist* and *researcher*). This imitation was addressed by static word embeddings (e.g., GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013)), which enabled semantic representation-based matching and improved performance (Ganea and Hofmann, 2017; Le and Titov, 2018). However, such models require precomputed embeddings for entities, limiting their zero-shot capabilities.

The development of the transformer-based encoder-only models (Devlin et al., 2019; Vaswani et al., 2017) allowed for more robust zero-shot EL. Most prominent methods rely on a bi-encoder or cross-encoder, which compares the entity mention and entity description using a language model (Logeswaran et al., 2019; Wu, Petroni, et al., 2020). While such methods allow zero-shot EL, KG information was only utilized by a few methods (Ayoola et al., 2022; Mulang, Singh, Prabhu, et al., 2020; Raiman, 2022; Zhang, Cao, and Groth, 2024). For a more thorough introduction to encoder-only, bi-encoder and cross-encoder methods, please refer to Section 2.4.2.

Recent works have explored generative EL methods. These approaches generally fall into two categories. First, approaches generating entity identifiers, typically human-readable labels such as `Barack Obama`, directly (Cao et al., 2021) and second, models that concatenate the entity mention and candidate entity descriptions and then generate the target entity name (Zhou et al., 2024) (see Section 2.4.2 for more information). While the former lacks zero-shot capabilities, a disadvantage of the latter is the increased processing time due to the larger context window necessary for concatenating all information (Bhargav et al., 2022). Neither uses KG information.

Only a few EL methods focus on the identification of whether the entity a mention is referring to actually exists (Ayoola et al., 2022; Hoffart et al., 2014; Zhou et al., 2024; Zhu et al., 2023). Similarly, little emphasis is given to clustering out-of-KG entities (Agarwal et al., 2022; Heist and Paulheim, 2023; Kassner et al., 2022). Yet these capabilities are critical for keeping KGs up-to-date, as the same out-of-KG entity may appear multiple times across documents. Existing clustering methods do not explore the potential of KG information in supporting this process. While

zero-shot EL for in-KG entities has received increasing attention, this thesis focuses instead on a more underexplored problem: how KG information can support the identification and clustering of out-of-KG entities (see Chapter 4).

1.4.2 Relation Extraction

RE, like EL, has a long-standing research history. One of the first efforts to define a standardized benchmark for the task was the ACE program, launched in 1999 (Doddington et al., 2004). Since then, significant advances have been made alongside the advancements in language models. Initially, identifying relations relied on extracting syntactical features. Similar to EL, the introduction of static word embeddings and transformer-based models also led to performance improvements (Nasar et al., 2022). The incorporation of KG information into the RE setup has rarely been studied in the past. Existing methods that incorporate KG information mostly focused on a fully-supervised setting with up to two hundred relations (Bastos et al., 2021; Jain et al., 2024; Jain et al., 2023; Mai et al., 2025; Sakor et al., 2019). Incorporating KG information into a single method that focuses on a large number of around 1000 relations with many long-tail relations is explored in our work in Chapter 6.

Exploring the impact of knowledge graph information on the task of zero-shot RE (Gao et al., 2019) is still largely missing in the literature. In our work, we examine the influence of KG information in this context in two ways: initially, through the inclusion of type information in Chapter 5, and subsequently, through the actual paths between entities in the KG in Chapter 7. While the inclusion of type information into the task of RE was already explored in the past (Zhao et al., 2024), many methods only focused on very broad entity types such as *person*, *location* or *organization*.

1.4.3 Joint Methods

Several methods have been proposed to jointly perform EL and RE (Dubey et al., 2018; Lin et al., 2021; Lin et al., 2020; Sakor et al., 2019; Sakor et al., 2020). Many of the existing approaches primarily focus on question answering where the encountered texts are short. While some methods incorporate information from KGs, their textual components are often outdated, and KG inclusion typically relies on exact matching techniques. Furthermore, zero-shot settings are largely neglected.

In contrast, our work considers texts of varying lengths and leverages neural language models alongside KG embedding techniques and targets zero-shot scenarios as well.

1.5 Publications

This section includes a list of all papers accepted as part of this thesis, along with all other papers to which the author contributed. Additionally, the contributions

of each author of the accepted papers comprising this thesis are described in more detail.

1.5.1 Accepted Papers Comprising This Thesis

- Cedric Möller, Jens Lehmann, and Ricardo Usbeck. 2022. Survey on English Entity Linking on Wikidata: Datasets and Approaches. *Semantic Web* 13 (6): 925–966 (**Chapter 3**)

- Cedric Möller and Ricardo Usbeck. 2024. Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering Using Only Knowledge Graphs. In *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands,* edited by Angelo A. Salatino, Mehwish Alam, Femke Ongenae, Sahar Vahdati, Anna Lisa Gentile, Tassilo Pellegrini, and Shufan Jiang, 60:88–105. Studies on the Semantic Web. IOS Press (**Chapter 4**)

- Cedric Möller and Ricardo Usbeck. 2024. Incorporating Type Information into Zero-Shot Relation Extraction. In *Joint proceedings of the 3rd International workshop on knowledge graph generation from text (TEXT2KG) and Data Quality meets Machine Learning and Knowledge Graphs (DQMLKG) co-located with the Extended Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 26-30, 2024,* edited by Sanju Tiwari, Nandana Mihindukulasooriya, Francesco Osborne, Dimitris Kontokostas, Jennifer D'Souza, Mayank Kejriwal, Maria Angela Pellegrino, Anisa Rula, José Emilio Labra Gayo, Michael Cochez, and Mehwish Alam, 3747:10. CEUR Workshop Proceedings. CEUR-WS.org (**Chapter 5**)

- Cedric Möller and Ricardo Usbeck. 2024. DISCIE-Discriminative Closed Information Extraction. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II,* edited by Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan, 15232:23–40. Lecture Notes in Computer Science. Springer (**Chapter 6**)

- Cedric Möller and Ricardo Usbeck. 2025. Analyzing the Influence of Knowledge Graph Information on Relation Extraction. In *The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I,* edited by Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena, 15718:460–480. Lecture Notes in Computer Science. **Best Student Paper Award**. Springer (**Chapter 7**)

1.5.2 Comments on the Degree of Authorship

The contributions to the five papers are described here following the CRediT (Contributor Roles Taxonomy) system[5].

For all five papers, I was responsible for conceptualization, methodology, software development, formal analysis, investigation, data curation, original draft writing, and visualization. This encompassed the full research workflow, including designing the architectures, implementing the code, planning and conducting experiments, analyzing the results, authoring the manuscripts, and creating the figures to effectively present the findings.

In the paper "Survey on English Entity Linking on Wikidata: Datasets and Approaches," Ricardo Usbeck and Jens Lehmann contributed primarily through writing-review and editing, as well as engaging in conceptual discussions that helped refine and clarify key ideas.

For the papers "DISCIE: Discriminative Closed Information Extraction," "Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering Using Only Knowledge Graphs," "Incorporating Type Information into Zero-Shot Relation Extraction," and "Analyzing the Influence of KG Information on Relation Extraction," Ricardo Usbeck's contributions involved writing-review and editing, and critical discussions that supported the development and strengthening of the research ideas.

1.5.3 Other Papers

In addition to the main contributions of this thesis, the following papers and book chapters were published throughout the doctorate as well. Some of these works are closely connected to the topic of this thesis, such as the Doctoral Consortium paper "Knowledge Graph Population with Out-of-KG Entities" and the book chapter "Neuro- Symbolic Relation Extraction".

Other papers with a more distant connection were produced in collaboration with colleagues, stemming from complementary research interests. The paper "Biomedical Entity Linking with Triple-aware Pre-Training" tackles the EL task as well; however, it focuses specifically on the biomedical domain and does not fit the KG-guided theme of the thesis itself.

Finally, the papers "Event Extraction Alone Is Not Enough" and "Ontology- Guided, Hybrid Prompt Learning for Generalization in Knowledge Graph Question Answering" are only loosely connected to the core topic of this thesis. They either address different information extraction tasks, such as event extraction, or focus on retrieving information from a KG in the form of KG question answering.

- Cedric Möller. 2022. Knowledge Graph Population with Out-of-KG Entities. In *The Semantic Web: ESWC 2022 Satellite Events - Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings,* edited by Paul Groth, Anisa Rula, Jodi Schneider, Ilaria Tiddi, Elena Simperl, Panos Alexopoulos, Rinke Hoekstra,

5. Please see https://credit.niso.org/ for a definition of each category.

Mehwish Alam, Anastasia Dimou, and Minna Tamper, 13384:199–214. Lecture Notes in Computer Science. **Runner-up**, **Best Doctoral Consortium Paper Award**. Springer

- Junbo Huang, Longquan Jiang, Cedric Möller, and Ricardo Usbeck. 2024. Event Extraction Alone Is Not Enough. In *Proceedings of Text2Story - Seventh Workshop on Narrative Extraction From Texts held in conjunction with the 46th European Conference on Information Retrieval (ECIR 2024), Glasgow, Scotland, UK, March 24, 2024,* edited by Ricardo Campos, Alípio Mário Jorge, Adam Jatowt, Sumit Bhatia, and Marina Litvak, 3671:105–114. CEUR Workshop Proceedings. CEUR-WS.org
- Xi Yan, Cedric Möller, and Ricardo Usbeck. 2025. Biomedical Entity Linking with Triple-aware Pre-Training. In *Proceedings of the Third International Workshop on Semantic Technologies and Deep Learning Models for Scientific, Technical and Legal Data (SemTech4STLD 2025), co-located with the Extended Semantic Web Conference (ESWC 2025),* edited by Rima Dessí, Joy Jeenu, Danilo Dessì, Francesco Osborne, and Hidir Aras. Portoroz, Slovenia: CEUR-WS.org, June
- Longquan Jiang, Junbo Huang, Cedric Möller, and Ricardo Usbeck. 2025. Ontology-Guided, Hybrid Prompt Learning for Generalization in Knowledge Graph Question Answering. **Best Student Paper Award**, *2025 19th International Conference on Semantic Computing (ICSC),* 28–35
- Xi Yan, Aida Usmanova, Cedric Möller, and Patrick Westphal. 2025. Neuro-Symbolic Relation Extraction. In *Handbook on Neurosymbolic AI and Knowledge Graphs.* IOS Press, March

1.6 Thesis Outline

As this is a cumulative thesis, the thesis contains two types of chapters. The first type provides the overarching framework for the published works. Chapters 1, 2 and 8 belong to this type. Specifically, Chapter 1 introduces the thesis as a whole, Chapter 2 introduces key concepts relevant to all included publications, while Chapter 8 summarizes the contributions and offers an outlook on future research.

In these chapters, typographic conventions are as follows: word and phrase examples are presented in *italics* (e.g., *running*), while relation names and entities in knowledge graph triples are set in monospaced font (e.g., `birthplace`, `BarackObama`). These conventions do not apply to chapters containing the original papers.

The second type comprises the chapters containing the original papers, which are included in their peer-reviewed and published form. The only modifications made are formatting adjustments to match the style of the thesis. Chapters 3 to 7 belong to this type.

2

Theoretical Background

2.1 Introduction

This chapter will introduce all concepts relevant to and the basis for the research conducted in the presented papers. Since knowledge graphs (KGs) are the overarching theme across all papers (relevant to Chapters 3 to 7) in this thesis, they are introduced in the first section. The section will describe their origin, structure and different KG domains.

Next, neural networks and specific types of neural networks are introduced as they are essential for all later chapters. After giving a general overview, we will focus in more detail on language models (relevant to Chapters 3 to 7) and knowledge graph embeddings (relevant to Chapters 3, 4 and 7). We introduce encoder-only, encoder-decoder and decoder-only transformer-based language models and distinguish between inductive and transductive knowledge graph embeddings.

The final section introduces the task of knowledge graph population (KGP), its connection to information extraction and their corresponding sub-tasks, especially entity linking (relevant to Chapters 3, 4 and 6) and relation extraction (relevant to Chapters 5 to 7).

2.2 Knowledge Graphs

While multiple definitions for knowledge graph exist, we base our understanding on the definition that it is "a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities" (Hogan et al., 2021, p. 2). While the term *knowledge graph* is not entirely new and has existed for more than 50 years (Schneider, 1973), today's popularity was heavily influenced by the introduction of the Google Knowledge Graph in 2012. While not all details on

how the Google Knowledge Graph works are public, it was introduced to improve the Google search results (Hogan et al., 2021; Singhal, 2012).

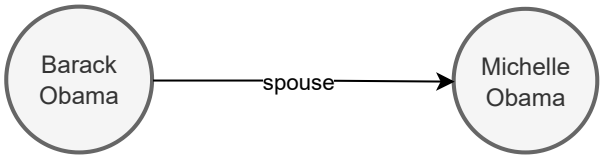


Figure 2.1: A triple expressing that Barack Obama is married to Michelle Obama. Barack Obama corresponds here to the subject node, Michelle Obama to the object node and spouse is the relation.

There are two major types of KGs: RDF (Resource Description Framework) graphs (World Wide Web Consortium, 2014) and labelled property graphs (Angles, 2018). Both consist of nodes and edges, but in labelled property graphs, each node or edge can be equipped with additional properties directly associated with the element. On the other hand, in RDF graphs, everything is modelled as triples. RDF "is a framework for representing information in the Web" (World Wide Web Consortium, 2014).

2.2.1 RDF and the Semantic Web

In this thesis, we focus exclusively on RDF graphs, given their widespread adoption in research (Nicolaie and Oprea, 2025) and the fact that major open-domain knowledge graphs, such as Wikidata (World Wide Web Consortium, 2014) and DBpedia (Lehmann et al., 2015), are based on RDF. Therefore, a more detailed introduction is warranted.

RDF plays a central role in the Semantic Web (Lassila et al., 2001), which extends the traditional document-based Web by making data machine-readable through the use of semantics, defining how data should be understood and related. RDF serves as the standard for describing resources on the Web, which facilitates interoperability by adhering to universal standards promoted by the World Wide Web Consortium (W3C). The W3C aims to develop open standards for the World Wide Web and has also created widely-used standards such as CSS and HTML (World Wide Web Consortium, 2025).

Therefore, RDF graphs are particularly suitable for applications where linking resources in different data sources is crucial. In contrast, labelled property graphs allow for more flexible data modeling with attributes on nodes and edges, making them favored in industries requiring rich metadata representation, such as social networking services (Tian, 2022).

Formally, RDF represents information as triples of the form $\langle$`subject`, `relation`, `object`$\rangle$, where each element is either a resource, a literal, or a blank node (with relations always being resources). See Figure 2.1 for an example of a triple. These triples form a directed, labeled multi-graph, with the subject and object as nodes and the relation as a labeled edge connecting them. It is a multi-graph, as multiple

edges can connect the same nodes and directionality is important because relations are commonly asymmetric, like `fatherOf`.

The node from which an edge originates is called the subject node, while the node with the incoming edge is the object node. Note that the subject node is sometimes also called the head and the object node is called the tail. Additionally, relations are often referred to as predicates. However, we adhere to the convention of calling them subject node, object node and relation to align with the tasks of entity linking and relation extraction, where these terms are commonly used.

A *resource* is anything that has a unique identity. Resources can be individuals, such as specific persons, plastic materials, or movies. Furthermore, abstract types, such as the notion of a person, are resources, and so are relations. Each resource is identified by a Uniform Resource Identifier (URI), which is similar in structure to URLs. For example, in the Wikidata KG, the resource corresponding to the former US president Barack Obama is assigned the URI `http://www.wikidata.org/entity/Q76`. In the context of entity linking, when we refer to *entities*, we refer to resources that are not relations and is often further restricted to not includes types as well.

Anything without a specific identity, such as strings, numbers, dates or similar, is denoted as a *literal*. For example, labels and descriptions of entities are literals, which we will heavily rely on in the rest of the thesis. Finally, *blank nodes* denote resources without actually naming them. They are used to represent more complex statements when the actual individual or type is irrelevant. While not inherently necessary, they are commonly employed (Hogan et al., 2014). See Figure 2.2 for an example KG containing entities and string literals.

In contrast to labelled property graphs, no attributes are assigned to individual nodes or edges; such information must be explicitly modeled through additional triples.

While RDF KGs can be seen as a way to store facts, many also contain ontological information. Ontological information governs how the facts in the KG can be interpreted. For example, it can be modeled that the father of a father is a grandfather. Given only $\langle$`x`, `fatherOf`, `y`$\rangle$ and $\langle$`y`, `fatherOf`, `z`$\rangle$, the ontology makes it possible to infer $\langle$`x`, `grandfatherOf`, `z`$\rangle$.

Furthermore, domain and range information can be specified as well by, for example, defining that the relation `fatherOf` has always a `human` as the subject and object entity. Also, it can be specified whether certain types are subtypes of others such as `human` being a subtype of `living being`. Therefore, a taxonomy which specifies how different types relate to each other can be modelled as well.

In RDF knowledge graphs, *RDF Schema (RDFS)* and the *Web Ontology Language (OWL)* (Hitzler et al., 2010) serve as extensions that enable the modeling of ontological information and the support of inference. We omit a deeper introduction of RDFS, OWL or other standards that belong to the *Semantic Web Stack* (Berners-Lee, 2009), which describes different layers of technologies for representing, linking, and reasoning over data on the Web.

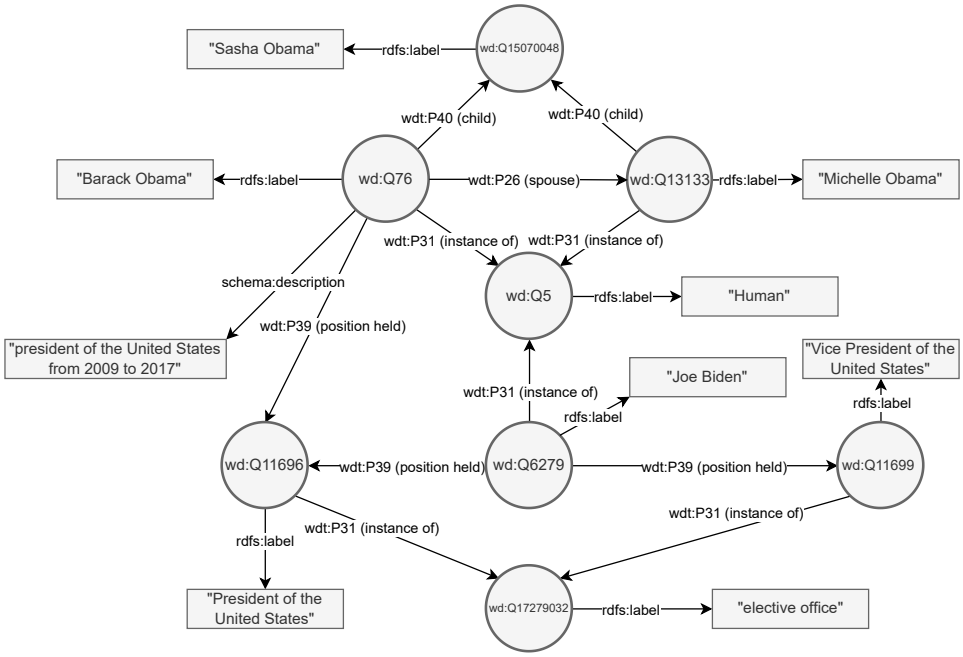


Figure 2.2: Example of a subgraph of the Wikidata KG depicting various entities in connection to Barack Obama. The round nodes denote specific entities while the rectangular nodes denote literals, which specify in this example information such as labels or descriptions of entities. The entities are marked with QIDs, which are the identifiers of entities in the Wikidata KG. Similar, the relations are marked with PIDs. `rdfs:label` and `schema:description` denote here the relations specifying the label and description of entities. Prefixes such as `rdfs`, `schema`, `wd`, and `wdt` are used to shorten full URIs. For example, `wd:Q76` stands for `http://www.wikidata.org/entity/Q76`.

2.2.2 Basic Components and Notation for Knowledge Graphs

This section formally defines a KG as understood and used in the following chapters.

A KG is defined as

$$\mathcal{K} = (\mathcal{N}, \mathcal{R}, \mathcal{A})$$

with $\mathcal{N}$ being the set of all nodes defined as $\mathcal{N} = \mathcal{E} \cup \mathcal{L}$ which consists of entities $\mathcal{E}$ and literals $\mathcal{L}$. $\mathcal{R}$ the set of relations and $\mathcal{A}$ the set of all triples. The set of triples $\mathcal{A}$ consists here of subject-relation-object expressions with $\langle x, r, y \rangle \in \mathcal{A}$ such that $x, y \in \mathcal{N}$ and $r \in \mathcal{R}$.

We distinguish entities based on whether they are *types* $\mathcal{T}$ or *instances* of a specific type $\mathcal{I}$. For example, a specific person called `Paul` is an instance. Instances are comparable to named entities as known in the area of natural language processing (Nadeau and Sekine, 2007). `Paul` does here belong the type `human` (Hogan et al., 2021). Types are also referred to as *classes* or *concepts*. Several types and corresponding instances may exist in a KG. Instances can belong to many different types such as `company`, `country`, `city`, `movie` and `gene`, among others. Note that this distinction is not explicitly made in RDF but only introduced in RDFS.

Relations denote a specific kind of relationship between two nodes. For example, an edge between two persons could be marked with `spouse` signifying that both persons are spouses. Relations can denote symmetric relationships such as a `spouse` or asymmetric ones such as `child`.

While RDF treats relations as resources (i.e., they have URIs and can be described just like entities), in the context of this thesis, we do not consider relations as nodes as well. The reason is that we do not extract statements about relations, but rather use relations to express relationsships between entities. Nevertheless, we do make use of literals in the source KG that are associated with relations (e.g., labels or descriptions), just as we do for entities.

2.2.3 Different Knowledge Graphs

The entities and relations that exist in a KG depend on the underlying domain. For example, in a biomedical KG, it is reasonable to model genes, diseases and their interactions. At the same time, modeling information about buildings is not of interest. In another domain, this might be different again.

Open-domain KGs such as DBpedia and Wikidata contain general information from different domains but often lack detail (Dong et al., 2014). Also, common-sense knowledge is often not modeled in such open-domain KGs. Statements such as that a dog has four legs are missing. Such information is modeled in common sense KGs (Ilievski et al., 2021). As already mentioned, there exist KGs with very specific domains such as biomedical (Bodenreider, 2004; Chandak et al., 2023; Wishart et al., 2018), geographical (GeoNames Team, 2025) or agricultural KGs (Coll et al., 2022). As all those graphs reside in the RDF format and this format allows the materialization of one of the main ideas of the Semantic Web (Lassila et al., 2001), data as linked information, the information from different graphs can be easily interconnected which allows reuse of existing resources.

Considering the significant role of Wikidata in this thesis, a deeper introduction to this KG is warranted. Wikidata is a KG created by the Wikimedia Foundation in 2012 (Vrandečić and Krötzsch, 2014). It is a public KG that virtually anyone can edit. It contains millions of entities[6] such as persons, organizations, locations, fruits and movies. Furthermore, it consists of billions of triples connecting the entities with each other. An important feature of Wikidata is qualifiers, which can be used to provide additional information about a statement. For example, by specifying since when two persons have been married.

Wikidata can be understood as the knowledge graph counterpart to its sister project, Wikipedia, representing a vast repository of encyclopedic knowledge in a structured, machine-readable form. Furthermore, Wikidata contains thousands of types connected via subtype relationships While not a strict taxonomy due to problems such as loops, multiple inheritance and inconsistencies in classification, these connections provide a rich hierarchical structure that can be leveraged further.

6. Note that in Wikidata, what we refer to as entities are called *items*, while relations are called *properties*. Furthermore, both items and properties are part of a broader category known as *entities* within the Wikidata context.

Wikidata has been steadily increasing in size in recent years, with the number of entities growing from approximately 92 million in 2021 to around 113 million by 2025 (Wikimedia Foundation). Similarly, the number of triples increased from 1.17 billion to 1.63 billion (Wikimedia Foundation). Furthermore, it contains numerous crosslinks to other KGs, thereby contributing to the broader vision of the Semantic Web.

While Wikidata is a very valuable resource, it also comes with some downsides, such as its loose conformity to an actual ontology, a representational bias and a certain level of noise due to its open-edit nature (Kraft and Usbeck, 2022; Santos et al., 2024; Shenoy et al., 2022). Lastly, while Wikidata is rich in information, it still lacks a large amount of information and is considered sparse (Abián et al., 2022).

2.3 Neural Networks

The field of neural networks is a subfield of machine learning, which is concerned with developing algorithms that enable computers to learn patterns from data. These learned patterns can then be applied to tasks such as prediction or clustering (Goodfellow et al., 2016, chapters 1 and 22).

2.3.1 Fundamentals

Artificial Neurons

Neural networks are one way to learn patterns from data. While modern neural networks were initially inspired by the structure and function of brain neurons, they process information and learn through mathematical functions that differ fundamentally from the complex biological mechanisms of actual neural activity (Russell and Norvig, 2020, p. 801).

The central idea is to approximate complex functions by combining multiple simple, learnable functions. A fundamental element of neural networks is an artificial neuron, where each artificial neuron consists of a combination of linear and nonlinear operations. The linear component of an artificial neuron is represented mathematically as follows:

$$a = \sum_{i=0}^{n} w_i x_i + b.$$

Here, w_i denotes the weight for input x_i, n is the number of inputs and b denotes the bias, which applies an offset to the linear combination. On top of that, a non-linear activation function σ is applied

$$y = \sigma(a).$$

The weight and bias of an artificial neuron are also denoted as the learnable parameters. The non-linear function is necessary to learn more complex functions than linear ones. Simply stacking linear functions would overall again be a linear function, making it impossible to learn more complex patterns (Goodfellow et al.,

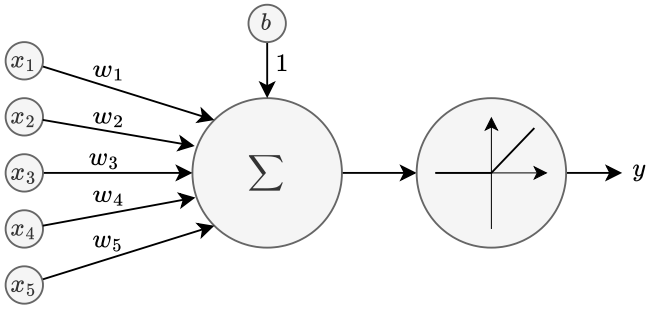


Figure 2.3: Example for an artificial neuron consisting of five inputs x_i, a bias b, the weighted summation using weights w_i and a non-linear ReLU activation function. The activation function is applied to the output of the summation.

.

2016, p. 803). Common non-linear activation functions are, for example, the standard logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

or the rectified linear unit (ReLU)

$$\sigma(x) = \max(0, x).$$

See Figure 2.3 for a schematic illustration of an artificial neuron.

Neural Network Architectures

Artificial neurons can be arranged in layers where each layer consists of multiple artificial neurons. Layers can be connected to each other where the previous layer's output is the next layer's input. If there are no feedback connections from later layers back at earlier layers, such networks are called feedforward neural networks. In such networks, the first layer is called the input layer and the last is the output layer. All layers in between are denoted as hidden layers (see Figure 2.4) (Goodfellow et al., 2016, chapter 22). The input layer receives the input features stemming from the data, which are then transformed into learned representations by the first hidden layer.

Today, most neural networks follow a deep learning architecture, consisting of several layers of artificial neurons. This depth allows for the extraction of increasingly abstract and sophisticated features at each layer (Goodfellow et al., 2016, chapters 19 and 22).

Training Neural Networks

A neural network's parameters are optimized over a loss function, which quantifies the difference between the actual model output and labels. In the context of machine learning, labels refer to the target outputs or ground truth values that a model is trained to predict, typically representing the correct answer for a given input. The type of loss function depends on the availability of data, namely whether actual labelled data (supervised) or no labels are available (unsupervised).

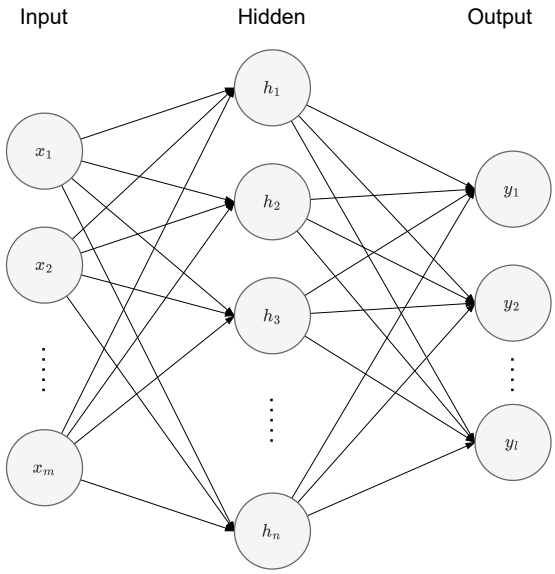


Figure 2.4: Three layers of fully connected artificial neurons, comprising an input layer with m units, a hidden layer with h_n units and an output layer with l units. Each unit of the hidden and output layer corresponds to an artificial neuron as depicted in Figure 2.3. In contrast, the input layer units correspond to the features of the input.

Beyond the dependence on the availability of the label, the loss function also depends on the problem at hand (Goodfellow et al., 2016, chapter 22).

Given a loss function $\mathcal{L}$, the parameters of the neural network are then optimized using gradient descent. First, backpropagation calculates the gradients of the loss function with respect to each parameter of a neural network. These gradients then guide the gradient descent algorithm in updating the parameters to minimize the loss. Formally, this is defined as

$$\theta = \theta - \eta \cdot \nabla_\theta \mathcal{L}(\theta)$$

where θ represents the model parameters, η is the learning rate controlling the impact of the gradient update and $\nabla_\theta \mathcal{L}(\theta)$ denotes the gradient of the loss function with respect to the parameters (Goodfellow et al., 2016, chapter 5). Note that in practice, more advanced variants of gradient descent are applied.

Loss functions employed in Chapters 4 to 7 are cross-entropy, binary cross-entropy (Goodfellow et al., 2016, chapter 4) and the HingeABL loss (Wang et al., 2023). In the following, X stands for a (sub-)set of training data while Z is the discrete set of all classes. Each element $(x, y) \in X$ consists of an input example x and a label $y \in Z$. Cross-entropy loss is here defined as

$$\mathcal{L}_{\text{cross}}(X) = -\frac{1}{|X|} \sum_{(x,y)\in X} \log \frac{e^{\mu(x,y)}}{\sum_{z\in Z} e^{\mu(x,z)}}$$

with $\mu(x, z) \in \mathbb{R}$ being a scalar score computed by the model that example x belongs to class z. $\frac{e^{\mu(x,y)}}{\sum_{z\in Z} e^{\mu(x,z)}}$ corresponds here to applying the softmax function to

obtain a probability distribution over all classes. Minimizing this loss encourages the model to increase the score for the correct class relative to the others. The loss function is appropriate when the label space Z contains more than two mutually exclusive classes.

If only two classes exist, binary cross-entropy loss can be used:

$$\mathcal{L}_{\text{binary}}(X) = -\frac{1}{|X|} \sum_{(x,y)\in X} y \log \sigma(\mu(x)) + (1-y) \log(1 - \sigma(\mu(x)))$$

with $y \in \{0, 1\}$, $\mu(x) \in \mathbb{R}$ and σ denoting the standard logistic function defined as $\sigma(x) = \frac{1}{1+e^{-x}}$. By minimizing this loss, the model learns to produce scores $\mu(x)$ that are large and positive when the true label is 1, and large and negative when the true label is 0. This loss function is a simplified form of the general cross-entropy loss when applied to the binary classification setting.

The binary cross-entropy loss is not limited to binary classification tasks. It is also commonly used in multi-label classification, where an instance can belong to multiple classes simultaneously or to none. In this context, the loss is calculated independently for each class (i.e., whether the class holds or not) and then summed or averaged over all classes. This allows the model to learn, for each class, whether it should be assigned to the instance or not. This is especially useful for the task of relation extractions, as multiple relations might hold between the same pair of entities.

The HingeABL loss (Wang et al., 2023) is also applied in the multi-label classification setting. Its primary motivation is to address class imbalance, where some labels occur much less frequently than others and therefore receive less attention during training.

The loss is defined as

$$\mathcal{L}(X) = -\frac{1}{|X|} \sum_{(x,P_x,N_x)\in X} \sum_{z\in Z} \frac{w(x,z,P_x,N_x)}{\sum_{z'\in Z} w(x,z',P_x,N_x)} \log\left(\sigma(-d\left(x,z,P_x,N_x\right))\right) \tag{2.1}$$

where Z is the set of all class labels and

$$w(x,z,P_x,N_x) = \max\left(0,\ m - d\left(x,z,P_x,N_x\right)\right) \tag{2.2}$$

$$d(x,z,P_x,N_x) = \begin{cases} \mu(x,z) - s_T & \text{if } z \in P_x \\ s_T - \mu(x,z) & \text{if } z \in N_x \end{cases} \tag{2.3}$$

with $P_x \subseteq Z$ and $N_x \subseteq Z$ denoting the sets of positive and negative labels for a given example, respectively. Here, $\mu(x,z) \in \mathbb{R}$ is the model score of example x for class z, s_T is a learnable threshold, m is a predefined margin constant and σ is again the logistic function.

The idea is to encourage the model to assign scores higher than the threshold s_T for positive classes and scores lower than the threshold for negative classes. The weight $w(x,z)$ places more emphasis on examples for which the margin m is not yet reached for a class, i.e., those that are not yet well-classified. This loss

is beneficial for relation extraction tasks that involve many relations, including long-tail relations.

While many considerations are involved in choosing and implementing loss functions and optimization techniques, these detailed discussions are beyond our current scope, but are well-documented in specialized literature (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016; Russell and Norvig, 2020).

2.3.2 Language Models

Language models aim to capture the statistical properties inherent to language with the goal of predicting word sequences (Jurafsky and Martin, 2025). Formally, they model the conditional probability distribution of word sequences:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_1, w_2, \dots, w_{i-1})$$

where w_i is a word. The probability of the next word therefore depends on all preceding words.

Attention Mechanism and Transformer Architecture

While language models were based on machine learning methods such as n-gram or hidden Markov models in the past, neural networks dominate today (Bengio et al., 2000; Jurafsky and Martin, 2025).

Especially with the development of the transformer architecture (Vaswani et al., 2017), training large models on even larger amounts of textual data has become feasible due to the use of the attention mechanism:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

where $Q \in \mathbb{R}^{m \times d_q}$, $K \in \mathbb{R}^{n \times d_k}$ and $V \in \mathbb{R}^{n \times d_v}$ are here denoted as the query, key and value matrices with d_q, d_k and d_v being the corresponding dimensionalities and m and n being the number of queries as well as keys/values. Softmax is defined as $\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ which leads to $\text{Softmax}(z_i) \in (0, 1]$ and $\sum_{i=1}^{n} \text{Softmax}(z_i) = 1.0$. The softmax function is applied to each row of the score matrix $\frac{QK^\top}{\sqrt{d_k}}$, effectively forming a probability distribution over all keys for each query. Finally, the output for each query is computed as a weighted average of the value vectors, where the weights correspond to the attention scores from the softmax. This means each query attends to all values, attending more to those values where the query and and corresponding key is similar.

Typically, the same input is processed by multiple parallel attention mechanisms, called *heads*, each of which learns to focus on different aspects of the input; this setup is known as *multi-head attention*. In that case, the output of each head l is computed as

$$\text{Attention}(QW_q^{(l)}, KW_k^{(l)}, VW_v^{(l)}).$$

A head-specific projection is thus applied to the key, value and query matrices. Attention is applied in the form of self-attention and cross-attention. In self-attention, all inputs come from the same sequence X, i.e., $Q = K = V = X$. In cross-attention, $Q = Y$ represents the target sequence, while $K = V = X$ corresponds to the source sequence. The attention mechanism makes efficient parallel training possible, which was vital for the success of recent language models.

The output of each attention head is, for example, concatenated and then usually passed through a feed-forward neural network. This, together with regularization methods, forms the transformer layer. There are two types of transformer layers: encoder and decoder transformer layers. The encoder layer relies solely on self-attention, while the decoder layer uses both self-attention and cross-attention. In the decoder, self-attention is applied over previously generated tokens while cross-attention allows the decoder to attend to the output representations of the encoder. See Figure 2.5 for a depiction of an encoder layer on the left combined with a decoder layer on the right. In today's models, multiple such transformer layers are stacked on top of each other to compute increasingly complex representations of the input (Vaswani et al., 2017).

We omit further details, such as the use of positional encoding to model sequence order or the definition of the regularization techniques necessary to enable generalizability of the model. We refer the reader to the original papers for a comprehensive introduction (Ba et al., 2016; He et al., 2016; Vaswani et al., 2017).

Types of Transformer-Based Language Models

In the context of language models, the attention mechanism allows the model to weigh the importance of each token[7] relative to other tokens, enabling the handling of long-range dependencies inherent in natural language text.

There are three different types of transformer-based language models: encoder-only, encoder-decoder and decoder-only methods. These models are usually pre-trained on a large amount of text data.

Encoder-only refers here to models encoding the input text to vector representations. They rely only on transformer encoder layers. Such models were mainly pre-trained on the task of masked language modeling (MLM) and next sentence prediction (NSP) (Devlin et al., 2019). MLM involves masking random tokens in a sentence and training the model to predict them based on the surrounding context and NSP trains a model to predict whether the succeeding sentence logically follows from the previous one. Both tasks proved to enhance the model's understanding of language structure. The benefit was that no manually labeled data was required to train the model; only the raw text was needed. This approach, known as self-supervised learning, is typically used during the pre-training phase of a model (Ericsson et al., 2022). Empirical results showed that the encodings learned during the pre-training stage retained rich semantic and syntactic information (Devlin et al., 2019; Jawahar et al., 2019).

7. Usually, due to the large number of words in languages, the word is split into multiple parts, denoted as tokens. A language model works directly on the tokens in a sequence.

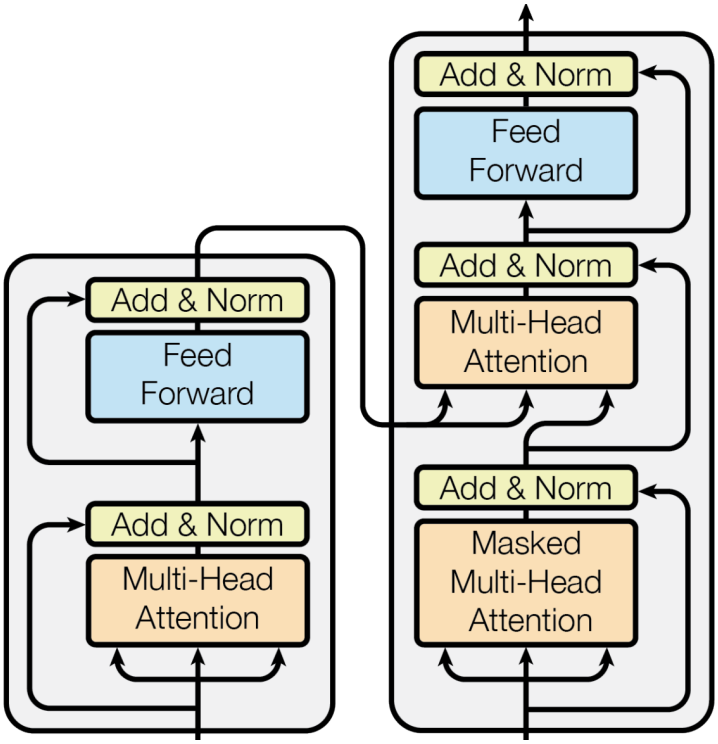


Figure 2.5: Schematic of transformer encoder and decoder layers. Add & Norm denotes the application of regularization in the form of layer normalization (Ba et al., 2016) combined with residual connections (He et al., 2016), which add the original input to the sub-layer output. On the left, the transformer encoder layer is shown, which performs self-attention on the input. On the right, the transformer decoder layer is depicted, which takes the already generated output and processes it through self-attention. This self-attention is masked (denoted as masked multi-head attention) to ensure that only previously generated outputs are attended to. Additionally, the decoder incorporates the encoder's output representations, which are processed using cross-attention. The three arrows entering the attention modules represent the computation of query, key and value vectors. Reproduced from (Vaswani et al., 2017).

The use of pre-trained representations for downstream tasks is known as transfer learning (Devlin et al., 2019; Pan and Yang, 2010). Without transfer learning, a model can either not be applied to a new task at all or will perform poorly when the new task differs from its pre-training objective. In the context of encoder-only models, *transferring* to a new task usually involves adding a small neural network, denoted as a task-specific *head*, on top of the pre-trained language model. The entire model is then trained further in a process known as fine-tuning on downstream tasks, such as named entity recognition or sentiment analysis. Pre-training may already lead the model to learn some notion of positive and negative sentiment in a sentence, with further transfer learning enhancing this capability. Commonly employed pre-trained encoder-only models are, for example, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) or DistilBERT (Sanh et al., 2019).

While encoder-only models are suitable to encode the input text for, e.g., classification tasks, other tasks like machine translation or text generation utilize encoder-decoder (Lewis et al., 2020) or decoder-only models (Radford et al., 2018),

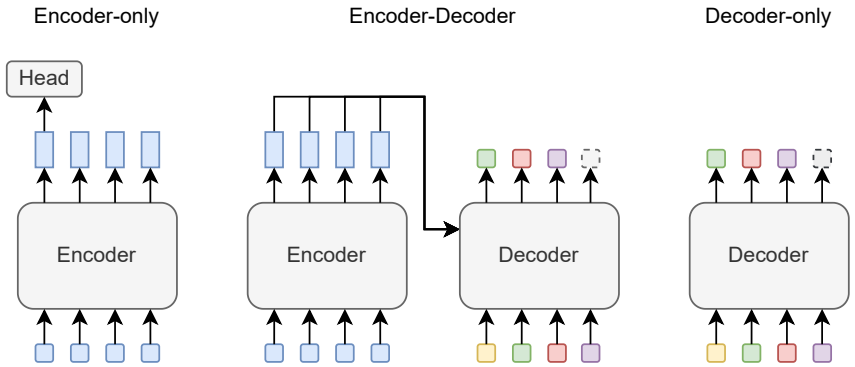


Figure 2.6: This figure compares encoder-only, encoder-decoder and decoder-only transformer architectures. Squares represent input/output tokens; rectangles represent vector representations. The same colors correspond to the same token or vector representation of the token. Note that in the decoder, output tokens are colored the same as their corresponding input tokens to indicate that each output is fed back as input at the next time step during autoregressive generation. The dashed squares symbolize the next token to predict in encoder-decoder and decoder-only models. The gray *Head* in the encoder-only model indicates a task-specific layer (e.g., classification).

respectively. Encoder-decoder and decoder-only models are also commonly referred to as generative models (Hagos et al., 2024). The encoder-decoder models first encode the input text using transformer layers with self-attention, then use the resulting representation in a separate decoder module, which combines self-attention over already decoded tokens and cross-attention over the encoded tokens to predict the next token. Encoder-decoder models are pre-trained according to sequence-to-sequence objectives where the goal is to predict an output sequence given a input sequence. Famous models in that regard are T5 (Raffel et al., 2020) or BART (Lewis et al., 2020).

In contrast to that, decoder-only models rely on the decoder alone. Rather than initially calculating a latent representation of the input, they treat the input as equivalent to the output. In this approach, each token in the sequence focuses solely on all preceding tokens and is used to generate the next token one at a time, which is also referred to as autoregressive generation (Brown et al., 2020). Decoder-only models are therefore trained on causal language modelling task where the objective is to predict the next token given all previous. Conversely, in encoder-only and encoder-decoder models, the encoder considers each token in relation to all other tokens in the sequence input. Popular decoder-only models are, for example, OpenAI's GPT models or the LLaMA (Radford et al., 2018; Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023a) models. Both encoder-decoder and decoder-only models predict output tokens by computing a similarity (using, for example, the dot product) between the decoder's representation and each token embedding in a vocabulary, followed by a softmax operation to produce probabilities. See Figure 2.6 for a comparison.

In the context of generative models, the primary architecture employed today is the decoder-only architecture, allowing large language models (Zhao, Zhan, et al.,

2023), containing billions of parameters, to be trained on petabytes of data (Radford et al., 2018).

Yet, this thesis primarily utilizes encoder-only architectures because they generate rich, contextualized word embeddings that seamlessly integrate with other forms of data representations, such as vector representations of knowledge graph elements. Additionally, as our primary focus is closed information extraction (see Section 2.4.5), where matching elements is more relevant than generating new ones, encoder-only models are a good fit.

2.3.3 Knowledge Graph Embeddings

Knowledge graph embeddings (KGE) aim to assign a vector representation to various knowledge graph elements. The fundamental assumption behind graph embeddings is that latent, high-dimensional representations of the underlying graph elements can be learned. These representations of nodes, edges, or (sub-) graphs can then be utilized for graph-specific tasks such as link prediction, node classification, or graph classification (Cao et al., 2024). As the computation of KGEs depends on the downstream task to be solved, we will briefly introduce these tasks.

Link prediction tries to identify, given a subject node and a relation, which other node in the whole graph is a probable object node. For example, given the node `SashaObama` and the relation `grandfather`, a link prediction model predicts the missing link to `BarackObamaSr`. The prediction is here inferred from other triples such as ⟨`SashaObama`, `father`, `BarackObama`⟩ and ⟨`BarackObama`, `father`, `BarackObamaSr`⟩. While this could also be explicitly modeled with logical rules, link prediction aims to learn implicit patterns from data.

Node classification aims to assign task-specific categories to each node, such as identifying users' interests in a social media network (Kipf and Welling, 2017).

Graph classification assigns categories to entire graphs and is commonly used in the biomedical field to, for example, classify whether a chemical compound graph is an enzyme (Gilmer et al., 2017).

In the following, we will primarily focus on KGE methods trained on the link prediction task (Ye et al., 2022) due to their focus on relations holding between nodes, matching our interest in the relation extraction problem.

KGE models can be categorized into transductive and inductive methods (Teru et al., 2020). In the transductive setting, the KG remains stable; the set of nodes and relations does not change and the overall structure of the KG does not change significantly. Conversely, the inductive setting permits the introduction of new elements, including new nodes, relations, or even a completely new graph.

Traditional Knowledge Graph Embedding Models

Traditional KGE models learn a fixed vector representation for each element in the KG. One example is the TransE architecture (Bordes et al., 2013), a popular model

for link prediction, where embeddings are learned by representing relationships between nodes as translations in vector space, formulated as

$$h_s + h_p = h_o$$

where s, p and o are the subject node, relation, object node, respectively (see Figure 2.7). h_s, h_p and h_o are here vector representations of the three elements. By

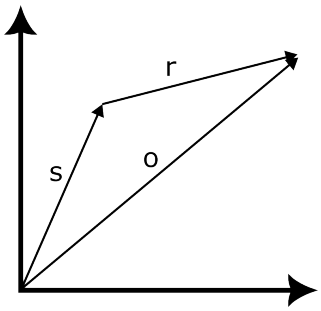


Figure 2.7: TransE: s is the vector representing the subject entity, r the vector representing the relation and o the vector representing the object entity. The goal is to arrive at o when adding r to s.

training this model on a graph, one strives to learn the underlying structure of the graph. This method comes with certain problems; for example, certain relationships, such as symmetric ones, can not be modeled as a simple translation, as it cannot encompass such patterns. To solve that, several extensions were proposed, such as DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), ComplexE (Trouillon et al., 2016), which can model more complex relationships.

Nevertheless, these graph embeddings are not as expressive in every scenario and do not generalize to new nodes. All such knowledge graph embeddings are transductive as they are learned over a static KG and are not adaptable to new elements without retraining.

Graph Neural Networks

In response to such limitations, graph neural networks (GNNs) were introduced. GNNs leverage neural networks to encode information from graph data structures, capturing complex relationships between nodes and edges (Ye et al., 2022). While graph neural networks exist for simple graphs, in the context of KGs, we will focus solely on graph neural networks designed for multi-graphs. Simple graphs are defined as graphs with at most one edge connecting any two nodes, whereas multi-graphs allow for the possibility of multiple edges between nodes (Diestel, 2000, pp. 25-26). For example, in a knowledge graph, a person node and a company node may be connected by multiple relationships, such as `occupation`, `founded` and `investedIn`.

Note that we use the terms *start node* and *target node* in the following, which here refer to the direction in which information flows during computation: the start node is the source of the information and the target node is the recipient. While this sometimes aligns with the terms subject and object node in a knowledge

graph triple, it more broadly reflects the direction of information flow in the computational model.

If GNNs are applied to KGs, this is also known as KG-Representation Learning (Ye et al., 2022). GNNs usually follow the *message-passing paradigm*, focusing on iterative updates and aggregation of node information to learn dynamic representations. We use such models in Chapter 7. This propagation relies on three steps (Gilmer et al., 2017): calculating the message, aggregating the messages and updating the node representation. Specifically, each node receives messages from all its neighboring nodes. The message is computed conditioned on the representation of the target node u, a neighboring node v and, optionally, the relation r connecting both:

$$m_{u,v,r}^{(l)} = M_t(g_u^{(l-1)}, g_v^{(l-1)}, e_{u,v,r}^{(l-1)}).$$

$e_{u,v,r}^{(l-1)}$ is the latent representation of the edge conditioned on the relation. $g_u^{(l)}$ is here the hidden representation of a node u at a specific layer l and M_l is the message computation function. Multiple different message computation functions exist, such as attention-based message computation used in RGAT (Busbridge et al., 2019) or edge-weighted messages used in RGCN (Schlichtkrull et al., 2018) methods. The messages can be conditioned on the start node, target node or relation of the edge. Then, the messages from all neighboring nodes are aggregated:

$$m_u^{(l)} = \bigotimes_{(v,r)\in\mathcal{N}(u)} m_{u,v,r}^{(l)}$$

where $\bigotimes$ is an aggregation function, being for example a sum, mean or maximum operation, and $\mathcal{N}(u)$ gives all the incoming edges to u equipped by the start node and the relation. Finally, the aggregation is used to update each node's representation:

$$g_u^{(l)} = U(g_u^{(l-1)}, m_u^{(l)})$$

where U is an update function. The update function can, for example, be a summation or a linear projection. Note that the message passing commonly traverses in the inverse direction of an actual edge in the graph as well; this is a frequent property of message passing algorithms and enabled by introducing inverse relations and edges into the graph. See Figure 2.8 for an illustration of the message passing process. This process is repeated multiple times, allowing information to propagate over n hops. Here, n hops refers to any nodes in the graph reachable by jumping n times from an initial node. Therefore, by repeating this process multiple times, information can propagate progressively further through the graph, reaching nodes at increasing distances. Given the final representation, it can be utilized for downstream tasks such as link prediction, node classification, or similar applications.

While message passing as described above is a potent approach, it still relies on learning the initial node representations. Consequently, it can not handle new nodes, limiting the applicability to a transductive setting where the graph remains static.

Conditional message passing (Zhu et al., 2021) extends the traditional paradigm by allowing GNNs to handle unseen nodes or relationships, operating in an inductive

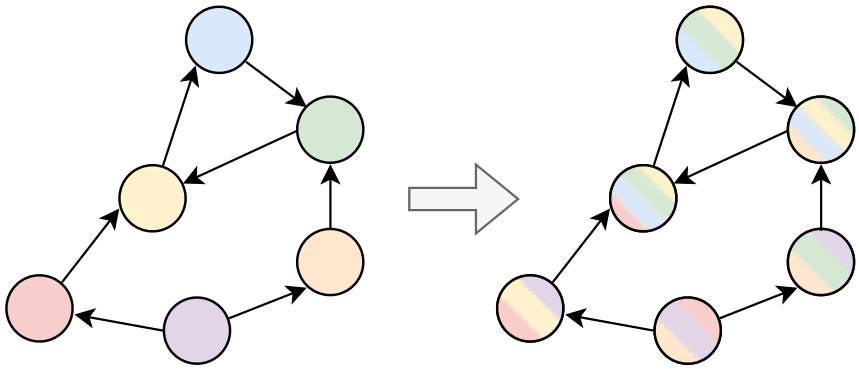


Figure 2.8: Illustration of a single iteration of regular message passing. On the left, each node starts with a distinct initial feature, indicated by different solid colors. After one iteration (right), each node aggregates information from its neighbors, resulting in a blended feature representation visualized by multicolored nodes. The color segments represent the influence of neighboring nodes, demonstrating how message passing fuses information across the graph.

setting.[8] Instead of initializing all nodes with a node representation, only a single node is initialized. This node determines how the message passing is conditioned. It is useful because, in the link prediction task, the goal is to predict an object node based on the subject and relation. As the message passing process is conditioned on this subject node, the network can focus its computation on the parts of the graph that are most relevant to the subject. This enables the model to generate context-aware representations, thereby improving the accuracy of object node prediction. The initialization of the subject node is a fixed learned vector that does not depend on the actual node, but rather on the structure of the KG.

After initialization, message passing is applied several times to compute a representation of other nodes in the same knowledge graph. As no node representations need to be learned, this allows such methods to generalize beyond known nodes. Effectively, the model predicts the potential object node based on all the paths of n-hop distance between the subject and the potential object node. See Figure 2.9 for an example of how conditional message passing works.

Finally, conditional message passing methods can even be extended to generalize beyond known relations (Galkin et al., 2024). For a proper definition of conditional message passing, please refer to Chapter 7 where they are employed.

8. Note that there exist regular message passing methods that work in the inductive setting as well (Teru et al., 2020). Conditional message passing methods are, however, the currently most powerful.

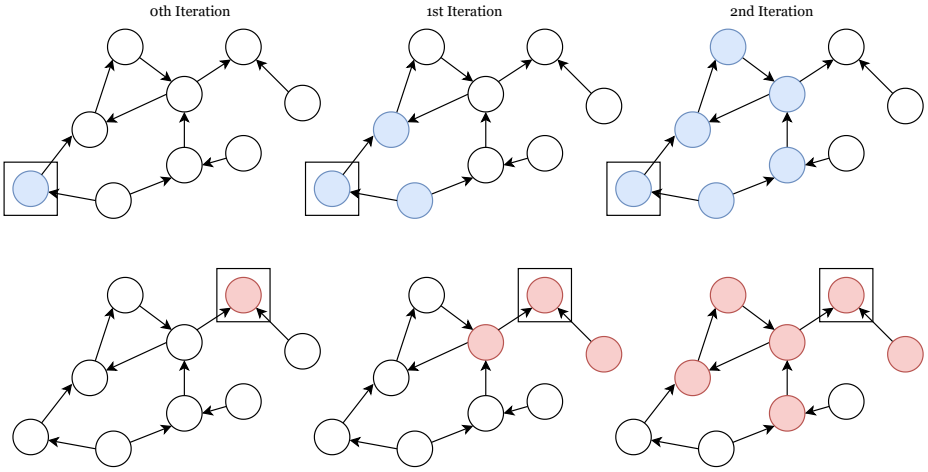


Figure 2.9: Illustration of conditional message passing across three iterations. The top and bottom show two separate propagation processes conditioned on different start nodes. Furthermore, the colors indicate that the process is conditioned on different relations as well. At each iteration (0th, 1st, 2nd), newly influenced nodes are highlighted in blue (top) or red (bottom), starting from the initial boxed node. The color-coded activations demonstrate how message propagation in the graph is modulated by the conditioning relation.

2.4 Knowledge Graph Population

Knowledge graph population (KGP) is a subtask of knowledge graph construction whichs objective is to create a full knowledge graph from scratch. It consists of two steps, *ontology construction* and the aforementioned *knowledge graph population*. While ontology construction is concerned with creating an ontology — defining all the entity types, relations — to describe the underlying domain, knowledge graph population actually aims to populate an existing knowledge graph using data (Heist, 2024).

In our context, we focus on natural language text as the data source. A more general task is information extraction or knowledge extraction, where the goal may include but is not limited to populating a knowledge graph with the extracted information (Ji and Grishman, 2011). Note that KGP is sometimes called *knowledge base population* as well.

Effective KGP requires solving several subproblems, including named entity recognition, entity linking, entity typing and relation extraction (Heist, 2024). When named entity recognition, entity linking and relation extraction are combined to extract triples, with each element linked to a KG entity or relation, it is referred to as closed information extraction as well (Cao et al., 2021), a special type of information extraction. In the following sections, we will introduce named entity recognition, entity linking and relation extraction. While entity typing, the task of assigning a type to an entity given data, is an important part of KGP as well, we do not focus on it throughout the rest of the thesis and therefore omit it.

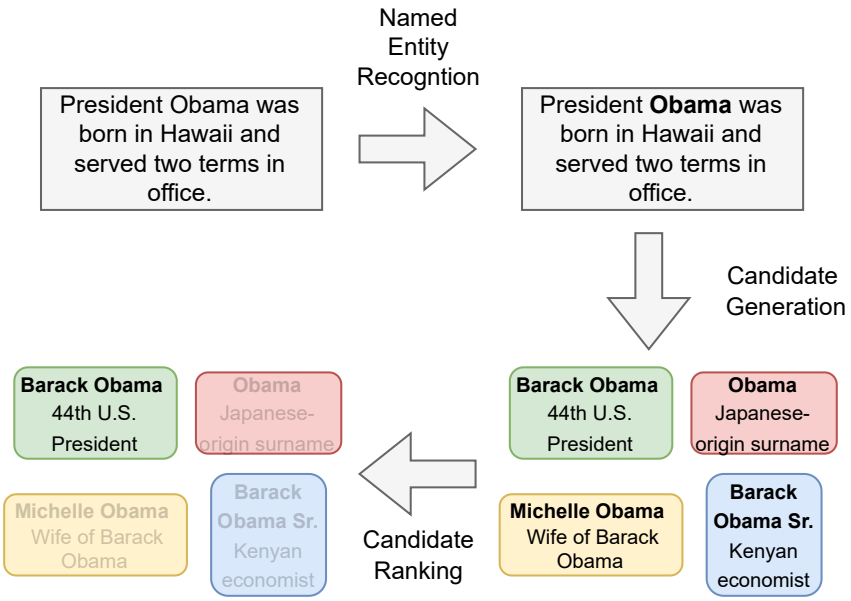


Figure 2.10: The entity linking process, illustrating the three main stages: named entity recognition, candidate generation and candidate ranking. For clarity, only one recognized entity is shown in the named entity recognition step. Additional entities, such as *Hawaii*, could also be extracted.

2.4.1 Named Entity Recognition

Named entity recognition (NER) is the task of identifying the spans of named entities in text, also denoted as *entity mentions*. Typically, named entity recognition also involves classifying the entity type. Commonly, only the four entity types `person`, `location`, `organisation` and `misc` are classified, but more entity types are possible as well.

Named entities are entities that correspond to specific individuals. For example, a successful NER system should extract *Volkswagen* and *Europe* from the sentence *The car manufacturer Volkswagen is one of the largest in Europe*, but not the noun *car manufacturer*. Additionally, the entity type `location` would be assigned to *Europe* while *Volkswagen* would get assigned `organisation`.

Named entity recognition is usually solved as a sequence tagging task where each token in the sentence is either tagged as the beginning, inside or outside of a named entity (Nasar et al., 2022) by both using non-generative and generative models (Keraghel et al., 2024).

2.4.2 Entity Linking

Entity linking (EL) is the task of linking entities in the text to the corresponding entities in a KG. When referring to EL, some works include the task of NER as well, which is then referred to as End-to-End EL (Sevgili et al., 2019). NER is then also often denoted mention detection (MD) and assigning the entity type

is excluded. In this thesis, in Chapter 4 we assume that the entity mentions are already identified, while in Chapter 6, we train a NER model to identify the entity mentions as well.

If the entity spans are identified, the next step is entity disambiguation (ED), which identifies the actual entity in the KG that is referred to. More formally, we assume that there is a set of entities $\mathcal{E}$ available in the KG, each equipped with some information about them, such as label information, a description, or connections to other entities in the KG. The goal of entity linking is now given an input text t, to identify a set of tuples in the input text denoted as $\{(s_1, e_1), (s_2, e_2), \dots, (s_i, e_i), \dots (s_{n-1}, e_{n-1}), , (s_n, e_n)\}$ where $e_i \in E$ and s_i denoting a span in the given text t.

Furthermore, the entity disambiguation step is usually split into two parts, entity candidate generation and candidate ranking. This is necessary as a KG can consist of millions or even billions of different entities and comparing against all of them in one step is usually unfeasible. Figure 2.10 visualizes the pipeline of entity linking with an example.

Candidate Generation

There are multiple ways to do candidate generation. One approach is string-based matching using the available labels in the KG (Logeswaran et al., 2019). Usually, this means splitting the labels into multiple n-grams on a character-level. For example, the label `Berlin` can be split into the following character-level 3-grams: `Ber`, `erl`, `rli` and `lin`. These 3-grams are then used to match parts of the input text for candidate generation. This is usually done using methods like TF-IDF, which are calculated as the product of term frequency (how often a term appears in a label) and inverse document frequency (a measure of how rare the term is across all labels), helping to weight important terms higher while downplaying common ones (Leskovec et al., 2014, chapter 1).

In contrast, a second approach leverages frequency-based lookup tables. These are created from a large number of already labelled documents to estimate, for each potential entity mention, the probability that it corresponds to a specific entity in a knowledge graph. For example, if *Obama* is mentioned, the former US president *Barack Obama* will be returned with the highest likelihood instead of less popular entities like *Obama County*, because the lookup table shows how often the mention is linked to each possible entity based on past labeled documents. And there are, of course, more documents mentioning *Obama* that are referring to *Barack Obama*. All Wikipedia articles are commonly used for creating such a lookup table (Hoffart et al., 2014).

Finally, semantic retrieval methods can be employed, for example, by using a bi-encoder. A bi-encoder applies an encoder-only model to an entity mention and each candidate entity from the KG. The model is fine-tuned to map each input to a vector, with the goal that the entity mention's vector is similar to the correct candidate's vector. Similarity is usually measured by the dot product between the vectors or their cosine similarity, which is the dot product of the normalized vectors (see Figure 2.11). Usually, vector indexes are employed to efficiently find

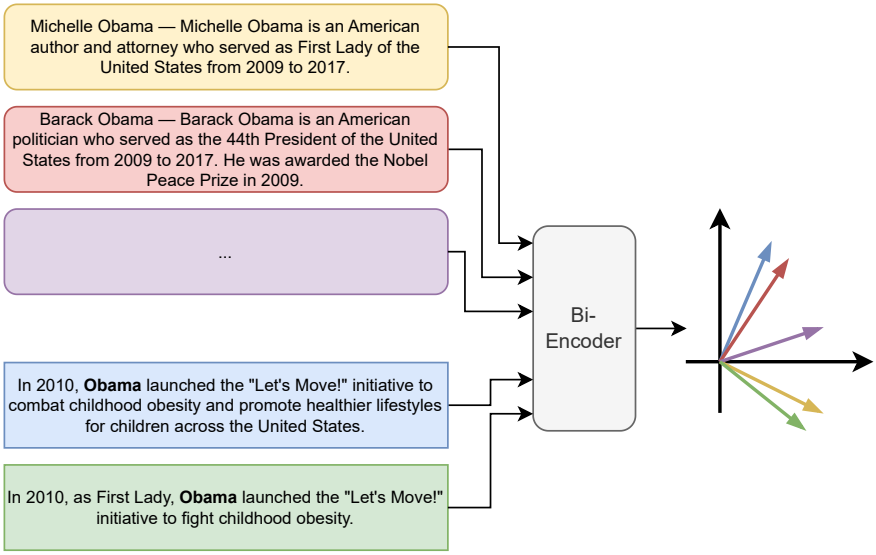


Figure 2.11: Applying entity linking examples to a bi-encoder. The red, yellow and purple blocks represent candidate descriptions, while the blue and green blocks represent the input examples with the marked entity mentions *Obama*, which need to be linked. The bi-encoder generates a vector for each input, aiming for the correct candidate's vector to be near the entity mention. It is shown that the vectors of the input examples are closer to the vectors of the correct candidates.

the closest entities to a given mention (Sevgili et al., 2019; Wu, Petroni, et al., 2020), which are then used as the candidate set.

Candidate Ranking

In the candidate ranking stage, all the candidates in the candidate set C are then ranked using more elaborate models to identify the highest-ranked entity, which is then determined as the linked entity. The candidate ranking is applied as only applying the previous candidate generation step usually leads to an inferior performance (Wu, Petroni, et al., 2020). Candidate ranking can be performed via a cross-encoder (Wu, Petroni, et al., 2020) which concatenates the entity mention text and the candidate text and feeds the combined input into an encoder-only model. This setup allows the model to jointly attend to both the mention and the candidate, enabling it to capture fine-grained interactions and context-dependent relevance signals better (see Figure 2.12). A cross-encoder outputs a single score for each concatenation. As the term hints at, a cross-encoder relies on an encoder-only model.

Similar to a cross-encoder, a generative model can also be employed during the entity ranking step. In this approach, the entity mention is concatenated with all candidate entities retrieved during candidate generation into a single input sequence. This sequence is then passed to a generative model, which is trained to produce the identifier of the most relevant candidate (Zhou et al., 2024).

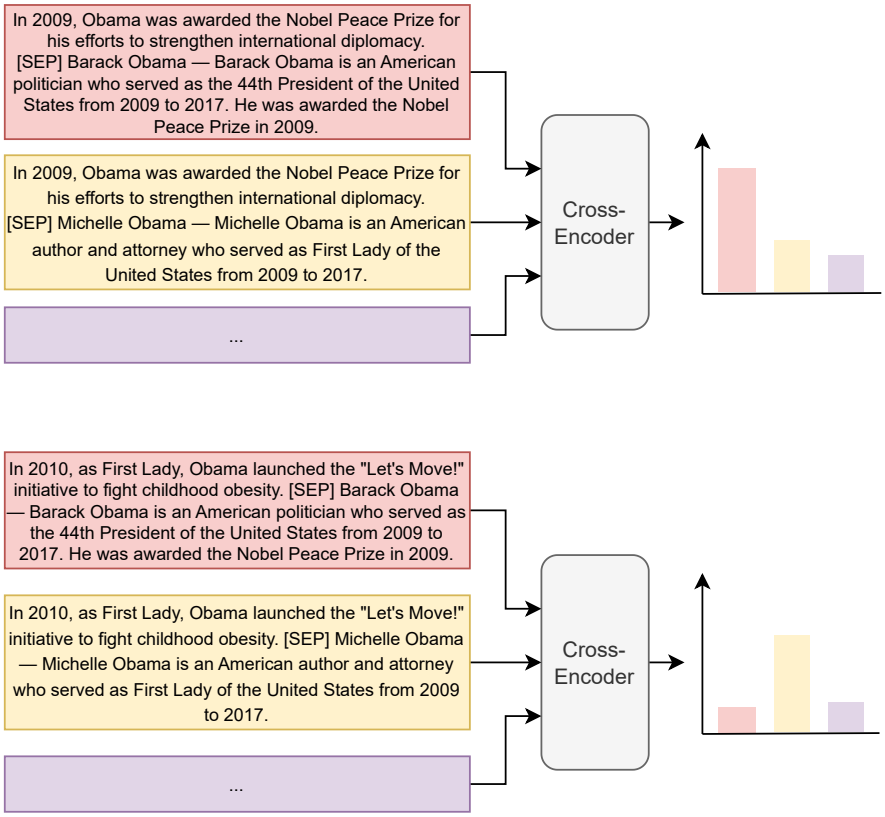


Figure 2.12: Using the same entity linking examples from Figure 2.11, we illustrate their application in a cross-encoder setup. The red, yellow and purple blocks represent concatenations of input examples with their respective candidate descriptions. These concatenated pairs form the inputs to the cross-encoder, which computes a relevance score for each. Notably, the cross-encoder must be applied six times, whereas the bi-encoder in Fig. 2.11 is applied only five times. As the number of reused candidates increases, the bi-encoder becomes significantly more efficient than the cross-encoder.

Alternatively, generative models can be trained to perform entity linking without an explicit candidate generation step. It involves generating entity identifiers, typically human-readable labels, such as `Barack Obama`, directly while restricting the output word distribution to only those identifiers of existing entities using constrained decoding (Cao et al., 2021). Constrained decoding restricts a model's output, ensuring that it only generates tokens or sequences that satisfy specific predefined rules or constraints, such as the set of entity identifiers. The model is trained to encode contextual information of all possible entities within its parameters (Cao et al., 2021). However, this makes it necessary to repeatedly retrain the model as today's KGs are still increasing rapidly in size (Foundation, 2025) and new entities are created every day.

Entity candidates are typically represented by combining the entity label with a description of the entity, as depicted in Figure 2.11. When performing entity linking with Wikidata, due to its close association with Wikipedia, a common method is also to use the first paragraphs of a Wikipedia article as the description

of the entity (Ayoola et al., 2022). While Wikidata also includes a description, the opening paragraphs of Wikipedia typically provide a more comprehensive overview of a person.

Out-of-KG Entities

Detecting out-of-KG entities, also denoted NIL entities (Heist, 2024), is typically done by applying a threshold to the scores produced by the entity disambgiuation stage. If no candidate surpasses this threshold, the entity is identified as an out-of-KG entity.

In the context of entity linking, when multiple out-of-KG entities need to be linked together, clustering methods are commonly applied to group similar entities (Agarwal et al., 2022; Heist and Paulheim, 2023; Kassner et al., 2022). Clustering, in this setting, refers to the task of assigning elements to clusters based on a defined similarity measure, allowing the aggregation of related entity mentions. Similarity is usually calculated using a bi-encoder or a cross-encoder.

In Chapter 4, we employ the popular DBSCAN clustering algorithm due to its ability to discover clusters of arbitrary shapes (Ester et al., 1996). DBSCAN clusters based on two parameters: a neighborhood radius ϵ and a minimum number of points ω. It classifies points as core points if they have at least ω neighbors within the ϵ-radius and border points if they are within ϵ-distance of a core point but do not meet the core point criterion themselves. Furthermore, any points that do not belong to any cluster are declared noise points. Points stand in the context of entity linking for the vector representations of entity mentions. By connecting core points and including their reachable border points, DBSCAN forms clusters. This approach makes DBSCAN particularly suitable for entity linking scenarios where the number of clusters is unknown, as the number of entities to be encountered is not predetermined.

2.4.3 Relation Extraction

Relation extraction focuses on the problem of identifying the relations expressed between two entities (see Figure 2.13). The usual assumption is that all the entity mentions are already marked via their spans $S = \{s_1, \dots, s_n\}$ in the text t. The goal is to determine whether and which relations hold between identified entity pairs. Therefore, a function is learned

$$r : S \times S \mapsto P(\mathcal{R})$$

where $P(\mathcal{R})$ is the power set of all relations. The function maps each pair of spans to a subset of all relations, including the empty set (Nasar et al., 2022).

In contrast to entity linking, relation extraction typically does not include a candidate retrieval step. This is because the set of possible relations is usually much smaller than the set of entities in entity linking. As a result, relation extraction can be performed by considering all possible relations directly.

There are different variants of relation extraction, such as sentence-level relation extraction, document-wide relation extraction, or cross-document relation

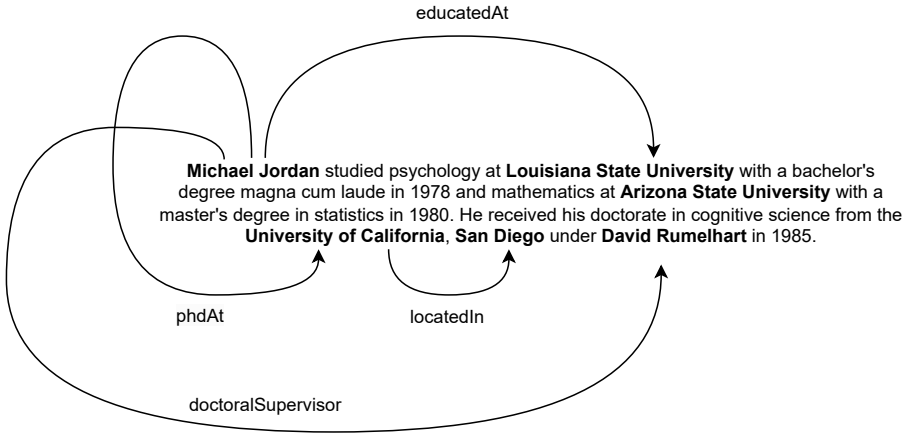


Figure 2.13: The figure illustrates how expressed relations, such as educatedAt, phdAt, doctoralSupervisor and locatedIn, that hold between the given entities, are extracted from unstructured data text.

extraction (Zhao et al., 2024). As indicated by the names, the number of and the distance between entity mentions differ between the subtasks. Document-wide relation extraction considers relations spanning an entire document, whereas sentence-level extraction focuses on single sentences. For example, consider the following texts:

Sentence-Level Example

Barack Obama was born in Honolulu, Hawaii.

Document-Level Example

Apple Inc. is headquartered in Cupertino, California. Tim Cook has been the CEO since 2011. Under his leadership, the company has launched several new products.

While a single sentence is enough to extract the relation `birthplace` holding between *Barack Obama* and *Honolulu*, multiple sentences need to be considered to identify that *Tim Cook* is the `ceoOf` *Apple Inc.*.

Cross-document relation extraction extends beyond single documents and focuses on relations that are only apparent when multiple documents are considered jointly (Yao et al., 2021; Zhao et al., 2024).

Furthermore, few-shot (Gao et al., 2019) and zero-shot relation extraction (Chen and Li, 2021) exist as well, where either only a few or even no examples are available for encountered relations. The objective is, therefore, to generalize to entirely new relations with only limited available data.

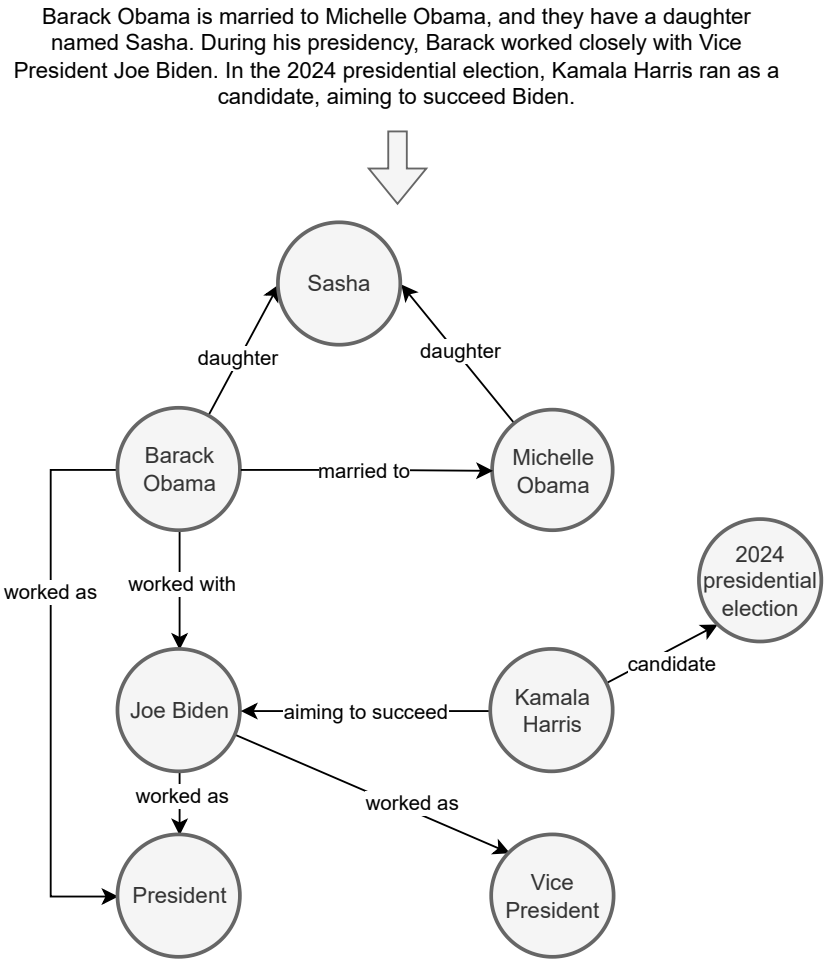


Figure 2.14: Open Information Extraction: Given the input document, the full knowledge graph is created. Notice that the relation and entity labels in the KG correspond more closely to the actual terms mentioned in the text. This is in contrast to closed information extraction, where the elements are linked to actual existing ones in a KG.

2.4.4 Open Information Extraction

Open information extraction involves extracting triples from text without relying on pre-defined identifiers[9] dedicated to entities or relations (Zhou et al., 2022). For example, a sentence such as *Barack is married to Michelle Obama* would be transformed into a triple of form $\langle$`Barack`, `married to`, `Michelle Obama`$\rangle$. Matching the actual triple elements to the entities and relations in the KG is not necessary. Only the words in the sentence matter.

Open information extraction is solved in two different ways. First, it is done by identifying the spans of the subject noun and object noun using NER and then detecting the span of the verb that expresses the relation between the two entities.

9. Also called Uniform Resource Identifiers (URIs) in RDF KGs.

The problem with this approach is that it only works if the relation is explicitly stated in the sentence. For example, the sentence *Bob has not lived in any other country than Germany* implicitly states that Bob's birthplace is Germany.

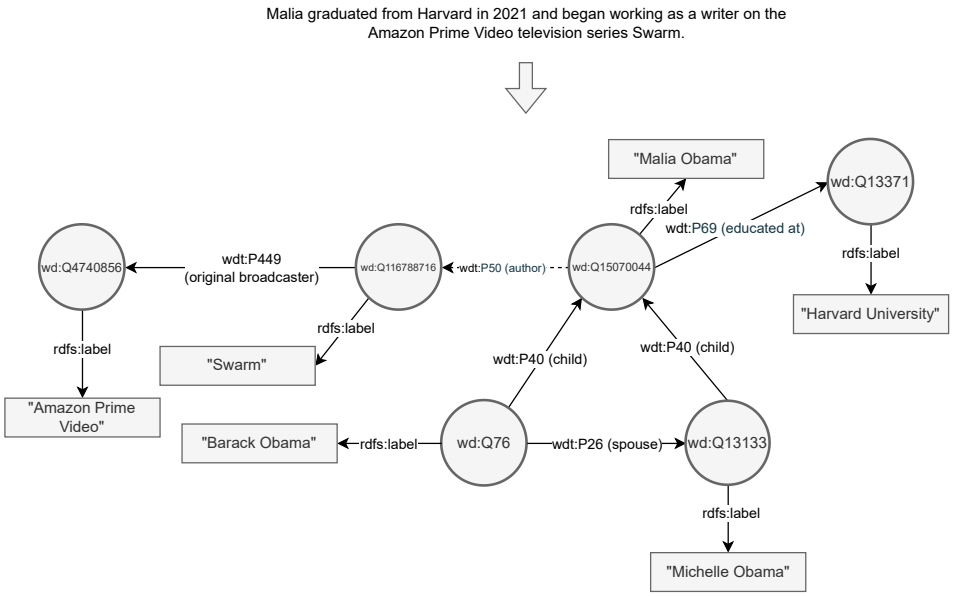


Figure 2.15: Closed Information Extraction: Solid edges correspond to already existing triples in the knowledge graph, while dotted edges denote the newly added triples from the input text. Furthermore, notice that in contrast to open information we extraction, we add the triples between existing entities using already defined reltions. The same prefixes as in Figure 2.2 are used.

Second, generative open information extraction approaches take in the input sentence and output triples directly. As they are generative, they do not rely on the existence of actual verbs (or nouns) describing the triple. This generative approach is particularly beneficial for identifying implicit relations that are not directly stated in text, such as inferred relationships. While open information extraction methods generalize very well to new domains, they usually do not focus on the problem of resolving ambiguous relations or entities over multiple input texts (Niklaus et al., 2018; Zhou et al., 2022). Nevertheless, open information extraction can be a preprocessing step to the task of closed information extraction. See Figure 2.14 for an example that builds a graph based on the input text.

2.4.5 Closed Information Extraction

In contrast to open information extraction, closed information extraction assumes that a set of entities and relations exists for which we want to extract triples (Josifoski et al., 2022). For example, entities and relations that are already part of an underlying KG. Entity linking and relation extraction are part of closed information extraction. Nevertheless, both can also be seen as part of open information extraction, especially if new relations or entities in the text that do not yet exist in the KG need to be extracted. See Figure 2.15 for an example.

Closed information extraction is necessary because being able to extract information that aligns with the relations or entities in the KG is essential to populate an existing KG with new information from unstructured data.

While we primarily focus on the task of closed information extraction in Chapters 5 to 7, the emphasis on out-of-KG entities in Chapter 4 shifts our attention toward open information extraction as well.

3

Survey on English Entity Linking on Wikidata

Bibliographic Information

Cedric Möller, Jens Lehmann, and Ricardo Usbeck. 2022. Survey on English Entity Linking on Wikidata: Datasets and Approaches. *Semantic Web* 13 (6): 925–966

Abstract

Wikidata is an always up-to-date, community-driven, and multilingual knowledge graph. Hence, Wikidata is an attractive basis for Entity Linking, which is evident by the recent increase in published papers. This survey focuses on four subjects: (1) How do current Entity Linking approaches exploit the specific characteristics of Wikidata? (2) Which unexploited Wikidata characteristics are worth to consider for the Entity Linking task? (3) Which Wikidata Entity Linking datasets exist, how widely used are they and how are they constructed? (4) Do the characteristics of Wikidata matter for the design of Entity Linking datasets and if so, how?

Our survey reveals that most Entity Linking approaches use Wikidata in the same way as any other knowledge graph missing the chance to leverage Wikidata-specific characteristics to increase quality. Almost all approaches employ specific properties like labels and sometimes descriptions but ignore characteristics like the hyper-relational structure. Thus, there is still room for improvement, for example, by including hyper-relational graph embeddings or type information. Many approaches also include information from Wikipedia which is easily combinable with Wikidata and provides valuable textual information which is Wikidata lacking.

The current Wikidata-specific Entity Linking datasets do not differ in their annotation scheme from schemes for other knowledge graphs like DBpedia. The potential for multilingual and time-dependent datasets, naturally suited for Wikidata, is not lifted.

3.1 Introduction

3.1.1 Motivation

Entity Linking (EL) is the task of connecting already marked mentions in an utterance to their corresponding entities in a knowledge base, see Figure 3.1.

There are multiple knowledge bases such as DBpedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2008), Yago4 (Tanon et al., 2020) or Wikidata (Vrandečić and Krötzsch, 2014). In contrast to DBpedia, Yago4, or Freebase, which mostly extract information from existing sources, Wikidata is a curated, community-based Knowledge Graph (KG). That is, the elements are added and edited by the community. The number of active editors is continuously increasing, see Figure 3.2. This allows Wikidata to stay up-to-date while automatically, one-time generated KGs such as Yago4 or Freebase become outdated over time (Ringler and Paulheim, 2017). Note, DBpedia stays also up-to-date but has a delay of a month.[10] DBpedia Live (DBpedia) exists, which is consistently updated with Wikipedia information. But it is more challenging to work with as no full dump is provided. Furthermore, the DBpedia ontology is not continuously updated, for example, with new emerging classes. The addition of new classes only comes with an update of the mapping-based extraction. On the other hand, new classes in Wikidata can be added continuously by the community. Furthermore, Wikidata is an inherently multilingual knowledge base. Both of these factors attract novel EL research over Wikidata in recent years cf. Figure 3.3. While Wikidata has its advantages regarding EL, exploiting those, for example in the form of hyper-relational structure (see Figure 3.4 for an example graph), is also challenging.

Primarily, this survey strives to expose the benefits and associated challenges stemming from the effective use of Wikidata as the target KG for EL. Additionally, the survey provides a concise overview of existing approaches, which is essential to (1) avoid duplicated research in the future and (2) enable a smoother entry into the field of Wikidata EL. Similarly, dataset landscape is structured, which helps researchers finding the correct dataset for their EL problem.

The focus of this survey lies on EL approaches, which operate on already marked mentions of entities, as the task of Entity Recognition (ER) is much less dependent on the characteristics of a KG. However, due to the only recent uptake of research on EL on Wikidata there is only a low number of EL-only publications. To broaden the survey's scope, we also consider methods that include the task of ER. We do not restrict ourselves to either rule-, statistical- or deep learning-based algorithms on Wikidata. This survey limits itself to the English language as it is the most dominant language in EL, and thus a better comparison of the approaches and

10. https://release-dashboard.dbpedia.org/

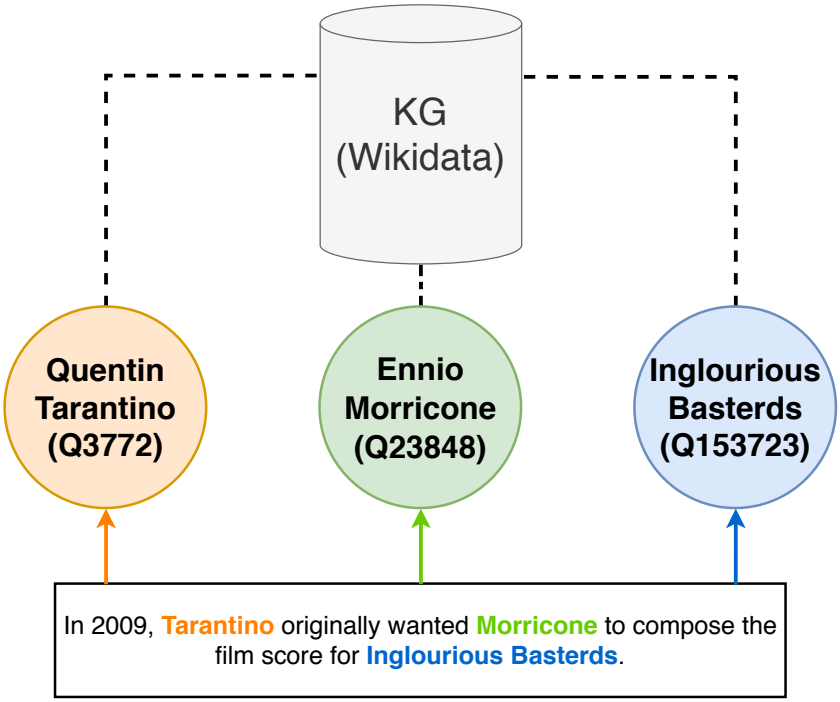


Figure 3.1: Entity Linking - Mentions in text are linked to the corresponding entities (color-coded) in a knowledge base (here: Wikidata).

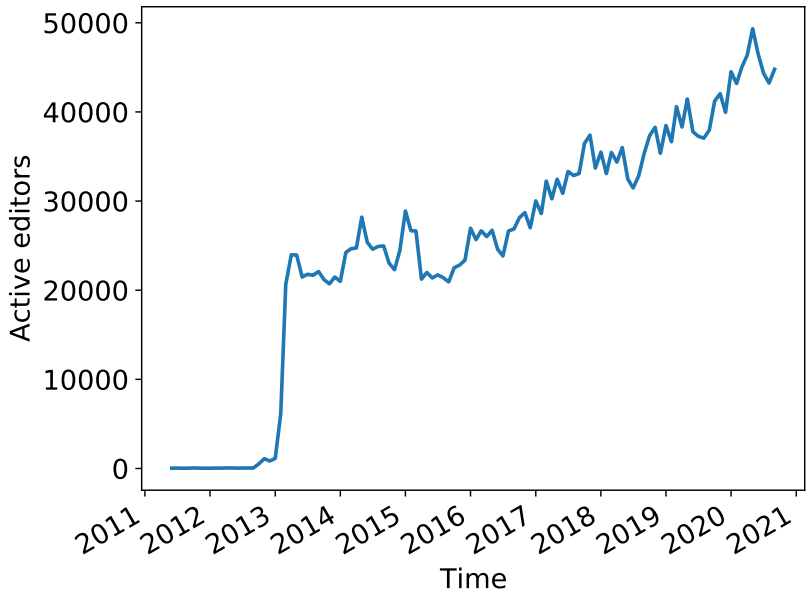


Figure 3.2: Active editors in Wikidata (Wikimedia Foundation, 2020b).

datasets is possible. Nevertheless, the topic of multilingualism is still of relevance in the analyses and discussions, as it is an essential characteristic of Wikidata. Since all multilingual Entity Linkers found also target English, none were excluded.

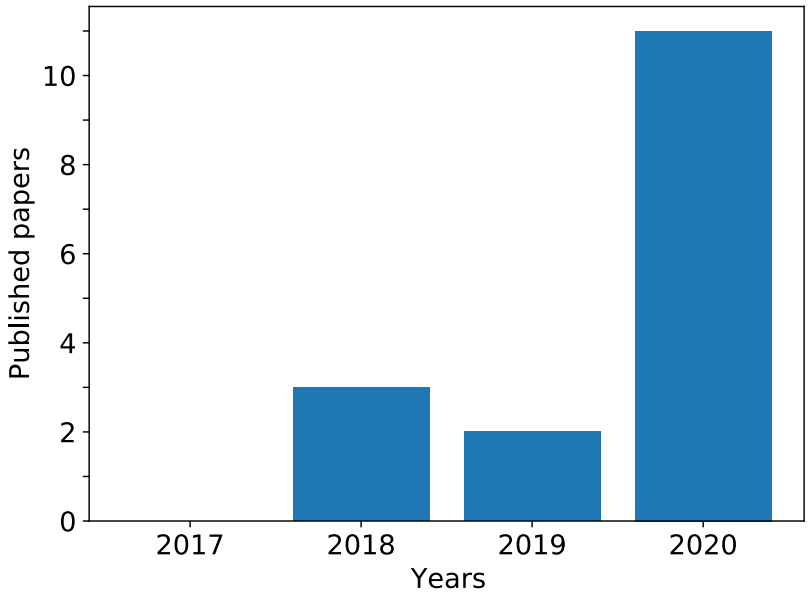


Figure 3.3: Publishing years of included Wikidata EL papers.

3.1.2 Research Questions and Contributions

EL approaches use many different kinds of information like labels, popularity measures, graph structure, and more. This multitude of possible signals raises the question of how the characteristics of Wikidata are used by the current state of the art of EL over Wikidata. Thus, the first research question is:

> **RQ 1**: How do current Entity Linking approaches exploit the specific characteristics of Wikidata?

In particular, which Wikidata-specific characteristics contribute to the solution? We answer this question by gathering all existing approaches working on Wikidata systematically (see Section 2) and analyzing them. The focus lies mainly on the usage of Wikidata's graph characteristics.

Secondly, we identify what kind of characteristics of Wikidata are of importance for EL but are insufficiently considered. This raises the second research question:

> **RQ 2**: Which unexploited Wikidata characteristics are are worth to consider for the Entity Linking task?

We tackle this question by giving an overview of the structure of Wikidata and the amount of information it contains, and then discussing the potential and challenges for EL.

Furthermore, we want to give an overview of which datasets for EL over Wikidata exist. Lastly, it is of interest if it is essential that datasets are designed with Wikidata in mind and if so, in what way? Thus, we post the following two research questions:

> **RQ 3**: Which Wikidata EL datasets exist, how widely used are they and how are they constructed?

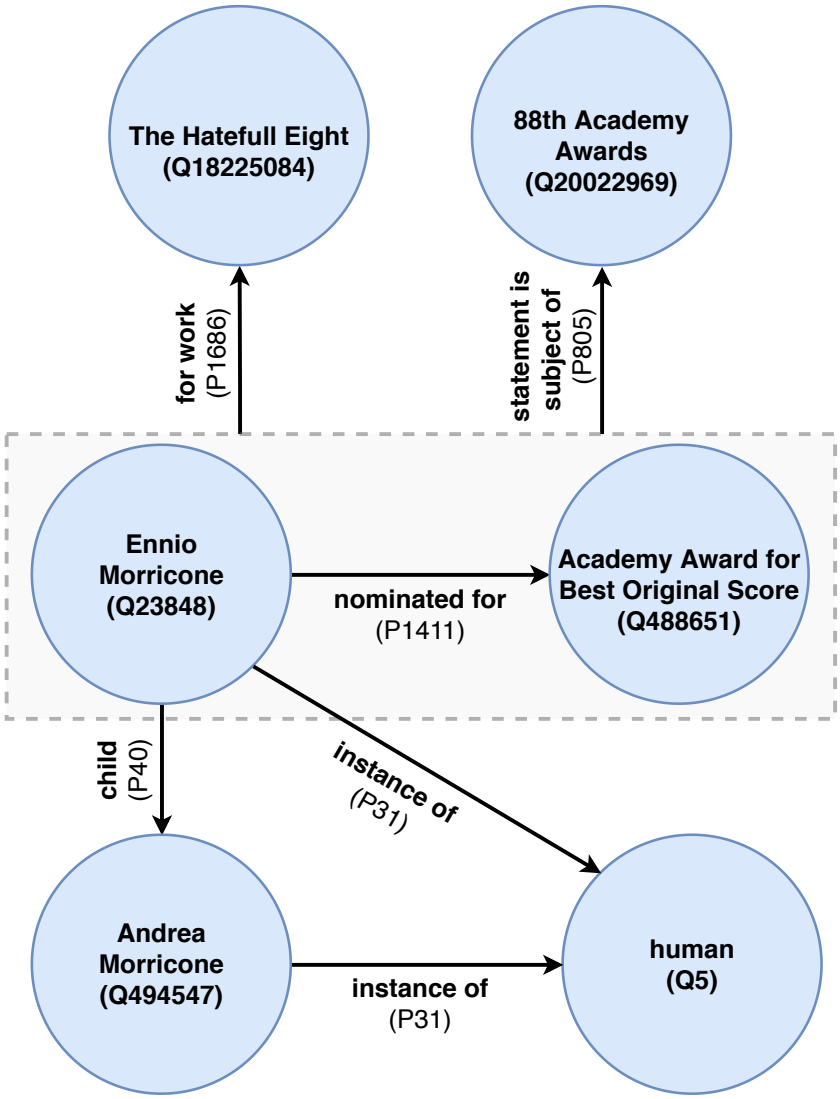


Figure 3.4: Wikidata subgraph - Dashed rectangle represents a claim with attached qualifiers.

> **RQ 4:** Do the characteristics of Wikidata matter for the design of EL datasets and if so, how?

To answer those two last research questions, all current Wikidata-specific EL datasets are gathered and analyzed with the research questions in mind. Furthermore, we discuss how the characteristics of Wikidata might affect the design of datasets.

This survey makes the following contributions:

- A concise list of future research avenues.
- A list and comparison of datasets focusing on Wikidata.
- An analysis of current evaluation results.
- A discussion of the relevance of Wikidata for Entity Linking.

3.2 Survey Methodology

There are several different types of surveys which desire to accomplish different contributions to the research field (Kitchenham, 2004):

1. Providing an overview of the current prominent areas of research in a field
2. Identification of open problems
3. Providing a novel approach tackling the extracted open problems (in combination with the identification of open problems)

Our related work section analyses different recent and older surveys on EL and highlights specific areas not covered and our survey's novelties. While some very recent surveys exist, they do not consider the different underlying Knowledge Graphs as a significant factor affecting the performance of EL approaches. Furthermore, barely any approaches included in other surveys are working on Wikidata and take the particular characteristics of Wikidata into account. To fill in the gaps, our survey gives an overview and examines all current EL approaches and datasets, focusing on Wikidata. Additionally, we identify less-utilized but promising characteristics of Wikidata regarding EL. Therefore, this survey provides contributions 2 and 3.

Until December 18, 2020, we continuously searched for existing and newly released scientific work suitable for the survey. Note, this survey includes only scientific articles that were accessible to the authors.[11]

3.2.1 Approaches

This survey's qualifying and disqualifying criteria for including papers can be found in Table 3.1. "Semi-structured" in this table means that the entity mentions do not occur in natural language utterances but more structured formats such as tables. The different approaches were searched for by using multiple different search engines (see Table 3.3).

Table 3.1: Qualifying and disqualifying criteria for approaches.

Criteria	
Must satisfy all	**Must not satisfy any**
• Approaches that consider the problem of unstructured EL over Knowledge Graphs • Approaches where the target Knowledge Graph is Wikidata	• Approaches conducting Semi-structured EL • Approaches not doing EL in the English language

To gather a wide choice of approaches the following filters were applied. Any approach where `Wikidata` was not occurring once in the full text was not con-

11. https://www.projekt-deal.de/max-planck-gesellschaft-verzichtet-ab-2019-auf-elsevier/

sidered. `Entity Linking` or `Entity Disambiguation` had to occur in the title of the paper. The publishing year was not a criterion due to the small number of valid papers and the relatively recent existence of Wikidata. The systematic search process resulted in 150 papers and theses (including duplicates).

Following this search, the resulting papers were filtered again using the qualifying and disqualifying criteria. This resulted in 16 papers and one master thesis in the end.

The search resulted in papers in the period from 2018 to 2020. While there exist EL approaches from 2016 (Almeida et al., 2016; Spitz et al., 2016) working on Wikidata, they did not qualify according to the criteria above.

3.2.2 Datasets

The dataset search was conducted in two ways. First, a search for potential datasets was performed using multiple search engines, see Table 3.3. Second, the datasets on which the approaches were evaluated were considered. The criteria for the inclusion of a dataset can be found in Table 3.2.

We scanned the dataset papers in the following way. First, in the title, `Entity Linking` or `Entity Disambiguation` had to occur once. Due to those keywords, other datasets suitable for EL but constructed for a different purpose like KG population were not included. Additionally, `dataset` must occur in the title and `Wikidata` has to appear at least once in the full text. This resulted in 20

Table 3.2: Qualifying and disqualifying criteria for the dataset search.

Criteria	
Must satisfy all	**Must not satisfy any**
• Datasets that are designed for EL or are used for evaluation of Wikidata EL • Datasets must include Wikidata identifiers from the start	• Datasets without English utterances

Table 3.3: Search engines.

Search Engines
• Google Scholar • Springer Link • Science Direct • IEEE Xplore Digital Library • ACM Digital Library

papers (including duplicates). Of those, only two included Wikidata identifiers and focused on English.

Eighteen datasets are accompanying the different approaches. Many of those did not include Wikidata identifiers from the start. This makes them less optimal for the examination of the influence of Wikidata on the design of datasets. They are included in the section about the approaches but not in the section about the Wikidata datasets.

After removal of duplicates, 11 Wikidata datasets are included in the end.

3.3 Problem Definition

EL is the task of linking an entity mention in unstructured or semi-structured data to the correct entity in a KG. The focus of this survey lies in unstructured data, namely natural language utterances.

An utterance is defined as a sequence of n words.

$$s = (w_0, w_1, ...w_{n-1})$$

Since not only approaches that solely do EL were included in the survey, Entity Recognition will also be defined.

There exists no universally agreed on definition of an entity. In general, named entities like a specific person or an organization are desirable to link. But sometimes, also common entities, such as `interview` or `theater`, are included. What exactly needs to be linked, depends on the use case (Rosales-Méndez et al., 2020).

Entity Recognition. ER is the task of identifying the spans

$$(w_i, ..., w_k)|0 \leq i \leq k \leq n-1$$

of all entities in an utterance u. Each such a span is called an entity mention m. The word or word sequence referring to an entity is also known as the surface form of the entity. An utterance can contain more than one entity, often also consisting of more than one word. Sometimes, also some broad type of an entity is classified too. Normally, those are `person`, `location` and `organization`. Some of the considered approaches do this classification task and also use it to improve the EL. It is also up to debate what an entity mention is. In general, a literal reference to an entity is considered a mention. But whether to include pronouns or how to handle overlapping mentions depends on the use-case.

Entity Linking. EL is the task of linking the recognized entity mention to the correct entity in a KG. A KG is defined as a directed graph $G = (V, E, \mathcal{R})$ consisting of vertices V, edges E and relations $\mathcal{R}$. Often, vertices correspond to entities $\mathcal{E}$ or literals $\mathcal{L}$, which are concrete values like the height or a name. E is a list $(e_1, \ldots, e_n)$ of edges with $e_j \in V \times \mathcal{R} \times V$ where relations $\mathcal{R}$ specify a certain meaning for the connection between entities. Such edges are also called triples. But there exists

no single definition of a KG; vertices and edges can also be defined differently. A concrete definition of the Wikidata KG is provided in the next section.

In general, EL takes the utterance u and all identified entity mentions $M = (m_1, ...m_n)$ in the utterance and links each of them to an element of the set $(\mathcal{E} \cup \{unknown\})$. The *unknown* element is added to the set of vertices to be able to map to an unknown entity that is not available in the KG. Such an entity is also called a NIL or an emerging entity (Hoffart et al., 2014).

The goal of EL is to find a mapping function that maps all found mentions to the correct KG entities and also to identify if an entity mention does not exist in the KG.

EL is often split into two subtasks. First, potential candidates for an entity are retrieved from a KG. This is necessary as doing EL over the whole set of entities is often intractable. *Candidate generation* is usually performed via efficient metrics measuring the similarities between entities in the utterance and entities in the KG. The result is a set of candidates $C = \{c_0, \cdots, c_l\}$ for each entity mention m in the utterance.

After limiting the space of possible entities, one of the available candidates is chosen for each entity. This is done via a *candidate ranking* algorithm, which assigns a rank to each candidate, signalizing how likely it is the correct one.

$$\begin{aligned} rank_{local} &: C \times M \to \mathbb{R} \\ &\text{given by } (c, m) \mapsto rank_{local}(c, m) \end{aligned}$$

where $rank_{local}$ is a ranking function of a candidate. The goal is then to optimize the objective function:

$$A^* = \arg\max_A \sum_{i=1}^{n} rank_{local}(a_i, m_i) | a_i \in C_i$$

where $A = \{a_1, ..., a_n\} \in \mathcal{P}(\mathcal{E})$ is an assignment of one candidate to each entity mention m_i. $\mathcal{P}(*)$ is the power set operator.

The rank calculation of the candidates of one entity is often not independent of the other entities' candidates. In this case, another global ranking function will include the whole assignment:

$$rank_{global} : \mathcal{P}(\mathcal{E}) \to \mathbb{R} \text{ given by } A \mapsto rank_{global}(A)$$

The objective function is then:

$$\begin{aligned} A^* = \arg\max_A \Bigg[\sum_{i=1}^{n} rank_{local}(a_i, m_i) \Bigg] \\ + rank_{global}(A) \mid a_i \in C_i \end{aligned}$$

Those two different categories of reranking methods are called *local* or *global* (Ratinov et al., 2011).

There exists also some ambiguity in the object of linking itself. For example, there exists an Wikidata entity `2014 FIFA World Cup` and an entity `FIFA World Cup`. There is no unanimous solution on how to link the entity mention in the utterance `In 2014, Germany won the FIFA World Cup`.

Sometimes EL is also called Entity Disambiguation, which we see more as part of EL, namely where entities are disambiguated via the candidate ranking.

3.4 Wikidata

Wikidata is a community-driven knowledge graph edited by humans and machines. As of July 2020, it contained around 87 million items of structured data about various domains. Seventy-three million items can be interpreted as entities due to the existence of a `is_instance` property. As a comparison, DBpedia contains around 5 million entities (Tanon et al., 2020). Note that the `is_instance` property includes a much broader scope of entities than the ones interpreted as entities for DBpedia. However, Wikidata contains around 8.5 million persons while DBpedia only contains around 1.8 million (in October 2020). Thus, a large difference in size is obvious.

Table 3.4: KG statistics by (Tanon et al., 2020).

KG	#Entities in million	#Labels/Aliases in million	last updated
Wikidata	78	442	always
DBpedia	5	22	monthly
Yago4	67	371	November 2019

3.4.1 Definition

Wikidata is a collection of *entities* where each such an entity has a page on Wikidata. An entity can be either an item or a property. Note that an entity in the sense of Wikidata is generally not the same as an entity one links to via EL. For example, Wikidata entities are also properties which describe relations between different items. Linking to such relations is closer to Relation Extraction (Bastos et al., 2021; Lin et al., 2017; Sorokin and Gurevych, 2017). Furthermore, many of the items are more abstract classes, which are usually also not considered as entities linked-to in EL. Note that if not mentioned otherwise, if we speak about entities, entities in the context of EL are meant.

Item. Topics, classes, or objects are defined as items. An example of an item can be found in Figure 3.5. An item is enriched with more information using statements about the item itself. In general, items consist of one label, one description, and aliases in different languages. An unique and language-agnostic identifier identifies items in the form `Q[0-9]+`.

For example, the item with the identifier `Q23848` has the label `Ennio Morricone`, two aliases, `Dan Savio` and `Leo Nichols`, and `Italian composer, orchestrator`

and conductor
(1928–2020) as description at the point of writing. The corresponding Wikidata page can also be seen in Figure 3.5.

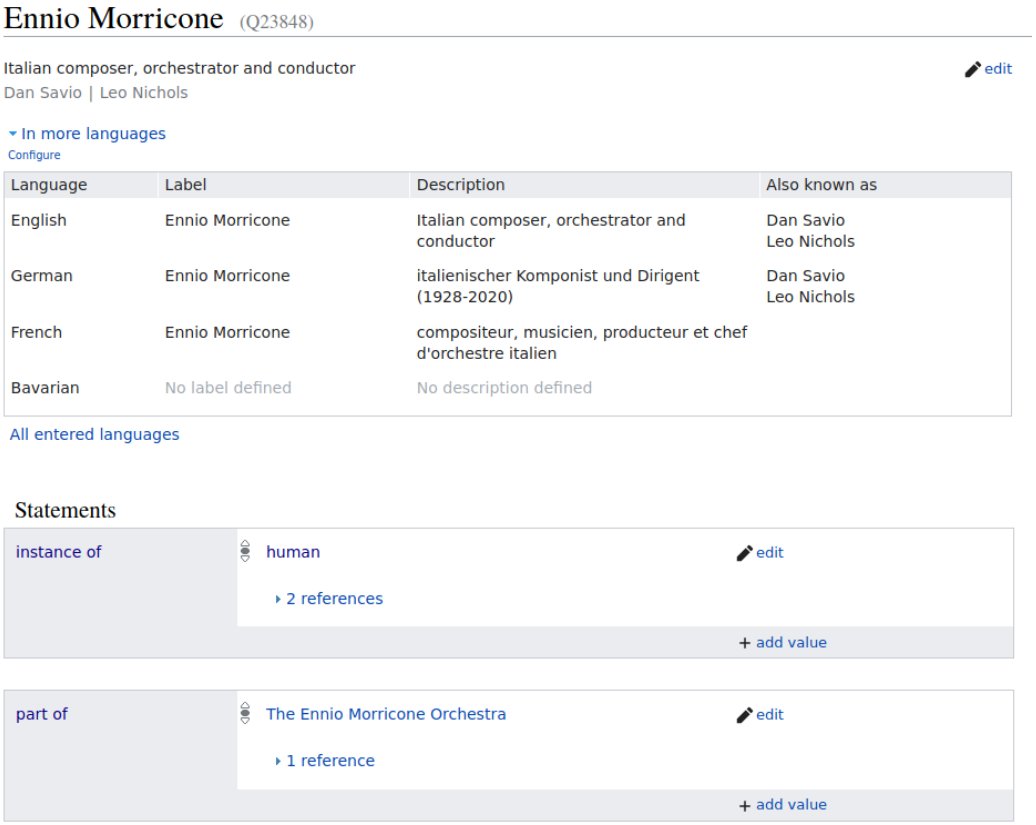


Figure 3.5: Example of an item in Wikidata

Not all items are entities in the context of EL. In general, items which are unique instances of some class are interpreted as entities. Of course, this also depends on the use case.

Property. A property specifies a relation between items/literals. Each property also has an identifier similar to an item, specified by $P[0-9]*$. For instance, a property P19 specifies the place of birth Rome for Ennio Morricone. In NLP, the term relation is commonly used to refer to a certain connection between entities. A property in the sense of Wikidata is a type of relation. To not break with the terminology used in the examined papers, when we talk about relations, we always mean Wikidata properties if not mentioned otherwise.

Statement. A statement introduces information by giving structure to the data in the graph. It is specified by a *claim*, and *references*, *qualifiers* and *ranks* related to the claim. Statements are assigned to items in Wikidata. A claim is defined as a pair of a property and some value. A value can be another item or some literal. Multiple values are possible for a property. Even an unknown value and a no value exists.

References point to sources making the claims inside the statements verifiable. In general, they consist of the source and date of retrieval of the claim. *Qualifiers* define the value of a claim further by contextual information. For example, a qualifier could specify how long one person was the spouse of another person. *Ranks* are used if multiple values are valid in a statement. If the population of a country is specified in a statement, it might be also useful to have the populations of past years available. The most up-to-date population information usually has then the highest rank and is thus usually the most desirable claim to use.

Statements can be also seen in Figure 3.5 at the bottom. For example, it is defined that `Ennio Morricone` is an `instance of` the class `human`. This is also an example for the different types of items. While `Ennio Morricone` is an entity in our sense, `human` is a class.

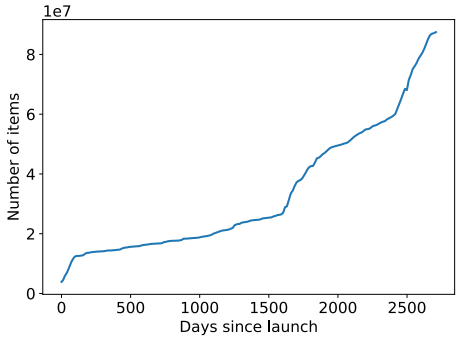
Hyper-Relational Graphs. Wikidata can thus be defined as a hyper-relational knowledge graph as statements can be specified by more information than a single claim. Multiple properties/relations are therefore part of a statement. In case of a hyper-relational graph $\mathcal{G} = (V, E, \mathcal{R})$, E is a list $(e_1, \ldots, e_n)$ of edges with $e_j \in V \times \mathcal{R} \times V \times \mathcal{P}(\mathcal{R} \times V)$ for $1 \leq j \leq n$, where $\mathcal{P}$ denotes the power set. A hyper-relational fact $e_j \in E$ is usually written as a tuple $(s, r, o, \mathcal{Q})$, where $\mathcal{Q}$ is the set of *qualifier pairs* $\{(qr_i, qv_i)\}$ with *qualifier relations* $qr_i \in \mathcal{R}$ and *qualifier values* $qv_i \in V$. (s, r, o) is referred to as the *main triple* of the fact. We use the notation Q_j to denote the qualifier pairs of e_j (Galkin et al., 2020). For example, under this representation scheme, the `nominated for` edge in Fig. 3.4 has two additional claims and would be represented as (`Ennio Morricone, nominated for, Academy Award for Best Original Score, (for work, The Hateful Eight), (statement is subject of, 88th Acade-my Awards)`) Structures similar to qualifiers exist also in some other knowledge graphs, such as the inactive Freebase in the form of Compound Value Types (Bollacker et al., 2008).

Other structural elements. The aforementioned elements are essential for Wikidata but more do exist. For example, there are entities (in the sense of Wikidata) corresponding to Lexemes, Forms, Senses or Schemas. Yet, as those are in general not of relevance for EL, we refrain from introducing them in more detail.

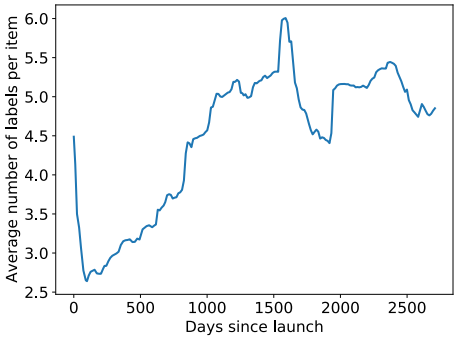
For more information on Wikidata, see the paper by Denny Vrandečić and Markus Krötzsch (Vrandečić and Krötzsch, 2014).

3.4.2 Discussion

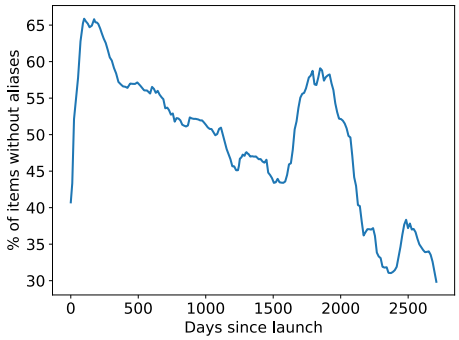
Novelties. As already mentioned, a useful characteristic of Wikidata is that the community can openly edit it. Another novelty is that there can be a plurality of facts, as contradictory facts based on different sources are allowed. Similarly, time-sensitive data can also be included easily by qualifiers and ranks. The population of a country, for example, changes from year to year which can be represented easily in Wikidata. Lastly, due to their language-agnostic identifiers, Wikidata is inherently multilingual. Language only starts playing a role in the labels and descriptions of an item.



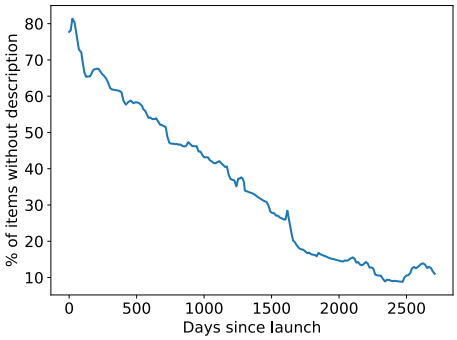
(a) Number of items of Wikidata since launch (Manske, 2020).



(b) Average number of labels (+ aliases) per item (Manske, 2020).



(c) Percentage of items without any aliases (Manske, 2020).



(d) Percentage of items without a description (Manske, 2020).

Figure 3.6: Statistics on Wikidata based on (Manske, 2020).

Strengths. Due to the inclusion of information by the community, recent events will always be included. The knowledge graph is thus much more up to date than other KGs. Freebase is unsupported for years now, and DBpedia updates its dumps only every month. Thus, Wikidata is much more suitable and useful for industry applications such as smart assistants since it is the most complete open accessible data source to date. In Figure 3.6a, one can see that number of items in Wikidata is increasing steadily. The existence of labels and additional aliases (see Figure 3.6b) helps EL as a too small amount of possible surface forms often lead to a failure in the candidate generation. DBpedia does for example not include aliases, only a single exact label; to compensate, additional resources like Wikipedia are often used to extract a label dictionary of adequate size (Moussallem et al., 2017). Even each property in Wikidata has a label (Vrandečić and Krötzsch, 2014). Fully language-model based approaches are therefore more naturally usable (Mulang, Singh, Vyas, et al., 2020). Also, nearly all items have a description, see Figure 3.6d. Thus, this short natural language phrase can be used for context similarity measures with the utterance. The inherent multilingual structure is intuitively useful for multilingual Entity Linking. Table 3.5 shows information about the use of different languages in Wikidata. As can be seen, are item labels/aliases available in up to 457 languages. Of course, not all items have labels in all languages. On average, labels/aliases/descriptions are available in 29.04 different languages. However, the median is only 6 languages. Many entities will therefore certainly not have information in many languages. The most dominant language is English but not

all elements have label/alias/description information in English. For less dominant languages, this is of course more severe. German labels exist for example only for 14 %, and Samoan labels for 0.3 %. Context information in the form of descriptions is also given in multiple languages but many languages are again not covered for each entity (as can be seen by a median of only 4). While the multilingual label and description information of items might be useful for language model based variants, the same information for properties enables multilingual language models. Because, on average, 21.18 different languages are available per property for labels, one could train multilingual models on the concatenations of the labels of triples to include context information. But of course, there are again many properties with a lower number of languages, as the median is also only 6 languages. Cross-lingual EL is therefore certainly necessary to use language-model based EL in multiple languages.

Table 3.5: Statistics - Languages Wikidata (Extracted from dump (Wikimedia Foundation, 2020a))

	Items	**Properties**
Number of languages	457	427
(average, median) of # languages per element (labels + descriptions)	29.04, 6	21.24, 13
(average., median) of # languages per element (labels)	5.59 , 4	21.18, 6
(average, median) of # languages per element (descriptions)	26.10, 4	9.77, 6
% elements without English labels	15.41%	0%
% elements without English descriptions	26.23%	1.08%

By using the qualifiers of hyper-relational statements more detailed information is available, useful not only for Entity Linking but also for other problems like Question Answering. The inclusion of hyper-relational statements is of course also more challenging. Novel graph embeddings have to be developed and utilized which can represent the structure of a claim enriched with qualifiers (Galkin et al., 2020; Rosso et al., 2020).

Table 3.6: Number of English labels/aliases pointing to a certain number of items in Wikidata (Extracted from dump (Wikimedia Foundation, 2020a))

# Labels/aliases	70,124,438	2,041,651	828,471	89,210	3329
# Items per label/alias	1	2	$3-10$	$11-100$	< 100

Weaknesses. However, this community-driven approach does also introduce challenges. For example, the list of labels of an item will not be exhaustive, as shown in Figures 3.6b and 3.6c. The graphs consider labels and aliases of all languages. While the average number of labels/aliases is around 5, not all are useful for Entity Linking in English. `Ennio Morricone` does not have an alias solely consisting of `Ennio` while he will certainly sometimes be referenced by that. Thus, one can not rely on the exact labels alone. But interestingly, Wikidata

has properties for the fore- and surname alone, just not as a label or alias. A close examination of what information to use is essential. However, this is also a problem in other KGs. Also, Wikidata often has items with very long, noisy, error-prone labels, which can be a challenge to link to (Mulang, Singh, Vyas, et al., 2020). Nearly 20 percent of labels have a length larger than 100 letters, see Figure 3.7. Due to the community-driven approach, false statements, due to errors or vandalism (Heindorf et al., 2016), also occur.

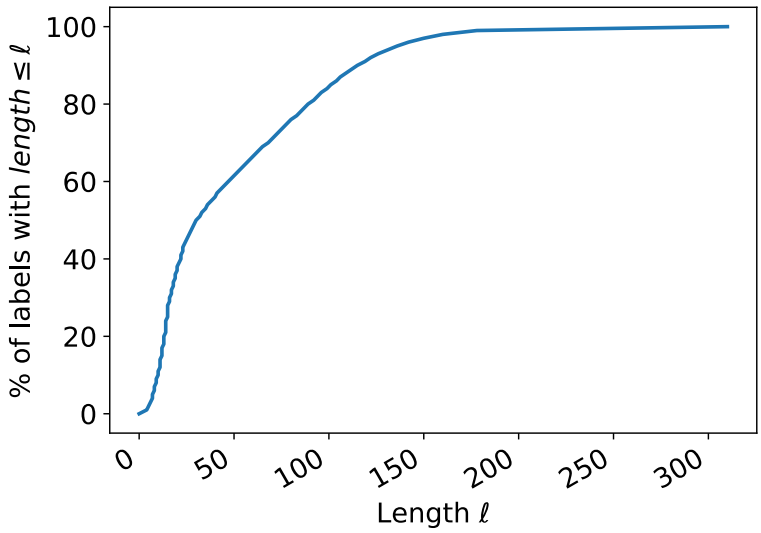


Figure 3.7: Percentiles of English label lengths (Extracted from dump (Wikimedia Foundation, 2020a))

Another problem may be the lack of facts (here defined as statements not being labels, descriptions, or aliases) for some entities. According to Tanon et al. (Tanon et al., 2020), in March 2020, DBpedia had, on average, 26 facts per entity while Wikidata had only 12.5. This is still more than YAGO4 with 5.1. However, those entities with fewer facts are probably also not occurring in DBpedia, which has a much lower amount of entities (Tanon et al., 2020). To tackle such long-tail entities, different approaches are necessary. The lack of descriptions can also be a problem. Currently, around 10% of all items do not have a description, as shown in Figure 3.6d. However, the situation is increasingly improving.

A general problem of Entity Linking is that a label or alias can reference multiple entities, see Table 3.6. While around 70 million mentions point each to an unique item, 2.9 million do not. Not all of those are entities by our definition but, e.g., also classes or topics. Also, longer labels or aliases often correspond to non-entity items. Thus, the percentage of entities with overlapping labels/aliases is certainly larger than for all items. To use Wikidata as a Knowledge Graph, one needs to be cautious of the items one will include as entities. For example, there exist `Wikimedia disambiguation page` items which often have the same label as an entity in the classic sense. Both, `Q76` vs `Q61909968` have `Barack Obama` as the label. Including those will make disambiguation more difficult. Also, the possibility of contradictory facts will make EL over Wikidata harder.

In Wikification, also known as EL on Wikipedia, large text documents for each entity exist in the knowledge base, enabling text-heavy methods (Wu, Petroni, et al., 2020). Such large textual contexts (besides the descriptions and the labels of triples itself) do not exist in Wikidata requiring other methods or the inclusion of Wikipedia. However, as Wikidata is closely related to Wikipedia, an inclusion is easily doable.

One can conclude that characteristics of Wikidata, like being up to date, multilingual and hyper-relational, introduce new possibilities while the existence of long-tail entities, noise or contradictory facts is also challenging. Thus, **RQ 2** is answered.

3.5 Approaches

3.5.1 Overview

Currently, the number of methods intended to work explicitly on Wikidata is still relatively small, while the amount of the ones utilizing the structure of Wikidata is even smaller.

There exist several KG-agnostic EL approaches (Moussallem et al., 2018; Usbeck, Ngomo, et al., 2014; Zwicklbauer et al., 2016). However, they were omitted as their focus is being independent of the KG. Of course, they do use Wikidata information like labels as this information also exists in other KGs, but it is no explicit usage of Wikidata-specific characteristics. While the approach by Zhou et al. (Zhou et al., 2020) does utilize Wikidata aliases in the candidate generation process, the target KB is Wikipedia and was therefore also excluded.

Tools without accompanying publications are not considered due to the lack of information about the approach and its performance. Hence, for instance, the Entity Linker in the DeepPavlov (Burtsev et al., 2018) framework is not included, though it targets Wikidata and appears to use label and description information successfully to link entities.

We distinguish three different kind of approaches: (1) Rule-based approaches, (2) approaches employing statistical methods and (3) neural network-based approaches. The vast amount of methods are using neural networks to solve the EL task (Banerjee et al., 2020; Boros et al., 2020; Botha et al., 2020; Cetoli et al., 2018; Huang et al., 2020; Klang and Nugues, 2020; Labusch and Neudecker, 2020; Mulang, Singh, Prabhu, et al., 2020; Mulang, Singh, Vyas, et al., 2020; Perkins, 2020; Provatorova et al., 2020; Raiman and Raiman, 2018; Sorokin and Gurevych, 2018). Some of those approaches solve the ER and EL jointly as an end-to-end task. Besides those, there exists one purely rule based approach (Sakor et al., 2020) and two based on statistical methods (Delpeuch, 2020; Lin et al., 2020).

The approaches mentioned above solve the EL problem as specified in Section 3.3. That is, other EL methods with a different problem definition also exist. For example, Almeida et al. (Almeida et al., 2016) try to link street names to entities in Wikidata by using additional location information and limiting the entities only to locations. As it uses additional information about the true entity via the location,

it is less comparable to the other approaches. Thawani et al. (Thawani et al., 2019) link entities only over columns of tables. It is not comparable since it does not use natural language utterances. The approach by Klie et al. (Klie et al., 2020) is concerned with Human-In-The-Loop EL. While its target KB is Wikidata, the focus on the inclusion of a human in EL process makes it incomparable to the other approaches. EL methods working on other languages than English (Ehrmann et al., 2020; Ellgren, 2020; Klang and Nugues, 2014; Vaigh et al., 2020; van Veen et al., 2016) were not considered but also did not use any novel characteristics of Wikidata. In connection to the CLEF HIPE 2020 challenge (Ehrmann et al., 2020), multiple Entity Linkers working on Wikidata were built. While short descriptions of the approaches are available in the challenge-accompanying paper, only approaches described in an own published paper were included in this survey. The approach by Kristanti and Romary (Kristanti and Romary, 2020) was not included as it used pre-existing tools for EL over Wikidata for which no sufficient documentation was available.

Due to the limited number of methods, we also evaluated methods that are not solely using Wikidata but also additional information from a separate KG or Wikipedia. This is mentioned accordingly. Approaches linking to knowledge graphs different from Wikidata, but for which a mapping between the knowledge graphs and Wikidata exists, are also not included. Such methods would not use the Wikidata characteristics at all and their performance depends only on the quality of the other KG and the mapping.

In the following, the different approaches are described and examined according to the characteristics of Wikidata used. For an overview, see Table 3.7.

Entity Linking

In the following, we will first focus on methods only doing EL.

In 2018, Cetoli et al. (Cetoli et al., 2018) evaluated how different types of basic neural networks perform solely over Wikidata. Notably, they compared the different ways to encode the graph context via neural methods, especially the usefulness of including topological information via GNNs (Sperduti and Starita, 1997; Wu, Pan, et al., 2020) and RNNs (Hochreiter and Schmidhuber, 1997). However, there is no candidate generation as it was assumed that the candidates are available. The process consists of combining text and graph embeddings. The text embedding is calculated by applying a Bi-LSTM over the Glove Embeddings of all words in an utterance. The resulting hidden states are then masked by the position of the entity mention in the text and averaged. A graph embedding is calculated in parallel via different methods utilizing GNNs or RNNs. The end score is the output of one feed-forward layer having the concatenation of the graph and text embedding as its input. It represents if the graph embedding is consistent with the text embedding. One crucial problem is that those methods only work for a single entity in the text. Thus, it has to be applied multiple times, and there will be no information exchange between the entities. While the examined algorithms do utilize the underlying graph of Wikidata, the hyper-relational structure is not taken into account. The paper is more concerned with comparing how basic neural

Table 3.7: Comparison between the utilized Wikidata characteristics of each approach.

Approach	Labels/ Aliases	Descr.	Knowledge graph structure	Hyper-relational structure	Types	Additional Information
OpenTapioca	✓	✗	✓	✓	✓	✗
NED using DL on Graphs	✓	✗	✓	✗	✗	✗
Falcon 2.0	✓	✗	✓[3]	✗	✗	✗
Arjun	✓	✗	✗	✗	✗	✗
DeepType	✓[1]	✗	✗	✗	✓[1]	Wikipedia[4]
Hedwig	✓	✓	✓	✗	✗	Wikipedia
VCG	✓	✗	✓	✗	✗	✗
KBPearl	✓	✗	✓	✗	✗	✗
PNEL	✓	✓	✓	✗	✗	✗
Mulang et al.	✓	✓[2]	✓	✗	✗	✗
Perkins	✓	✗	✓	✗	✗	✗
Huang et al.	✓	✓	✓	✗	✗	Wikipedia
Boros et al.	✗	✗	✗	✗	✓	Wikipedia, DBpedia
Provatorov et al.	✓	✓	✗	✗	✗	Wikipedia
Labusch and Neudecker	✗	✗	✗	✗	✗	Wikipedia
Botha et al	✗	✗	✗	✗	✗	Wikipedia
Tweeki	✓	✗	✗	✗	✓	Wikipedia

[1] In paper, just demonstrated for Wikipedia
[2] Appears in the set of triples used for disambiguation
[3] Only querying the existence of triples
[4] Wikidata not used in implementation/evaluation

networks work on the triples of Wikidata. Due to the pure analytical nature of the paper the usefulness of the designed approaches to a real-world setting is limited. The reliance on graph embeddings make it susceptible to change in the Wikidata KG.

Deeptype (Raiman and Raiman, 2018) is a novel approach using the type information of Wikidata or Wikipedia. Developed in 2018, first, a type system was optimized via stochastic optimization. A type system is a grouping of multiple type axes where a type axis is a set of mutually exclusive types. The idea is to classify entities according to the different type axes. Various methods to generate the type system were compared, such as a genetic algorithm. The objective was a type system which improves the EL performance while also being learnable. The learnability is important to guarantee that a classifier can be trained for the type system. After optimization, it consists of 128 different types. The authors do not mention how the candidates are generated. It is only stated that commonly it is

done via a dictionary, therefore, one can only assume that they used a dictionary. Then the words in an utterance are classified via a windowed Bi-LSTM according to the type system. The type probabilities are then used together with a link probability score to get the final score per candidate. This link probability a statistic on how often a mention is linked to an article of an entity in Wikipedia. The approach is multilingual as its learned type system is agnostic to language. Thus it can be easily used with entity mentions in different languages. It is important to note that they used Wikipedia categories to train their type system and Wikipedia articles to train the type classifier. However, the authors claim that the algorithm is easily changeable to Wikidata. Nevertheless, as it is also possible to adapt other algorithms, initially created for different KGs, to Wikidata, this method may not be suitable to be compared to the other algorithms. Assuming it could be used over Wikidata types, it seems to produce quite good results while only using a basic disambiguation algorithm besides the type classifier. The results show that incorporating detailed type information improves EL considerably. As Wikidata contains many more types (≈2,400,000) than other KGs, e.g., DBpedia (≈484,000) (Tanon et al., 2020), it seems to be more suitable for this fine-grained type classificaton. Yet, not only the amount of types plays a role but also how many types are assigned per entity. In this regard, Wikipedia provides much more type information per entity than Wikidata (Weikum et al., 2021). A shift to Wikidata is, therefore, not that simple. As Wikidata is growing every minute, it may also be challenging to keep the type system up to date.

The approach by Mulang et al. (Mulang, Singh, Prabhu, et al., 2020) is tackling the EL problem with Transformer (Vaswani et al., 2017) models. It is assumed that the candidate entities are given. For each entity, the labels of 1-hop and 2-hop triples are extracted. Those are then concatenated together with the utterance and the entity mention. The concatenation is the input of a pre-trained Transformer model. With a fully connected layer on top, it is then optimized according to a binary cross-entropy loss. This architecture results in a similarity measure between the entity and the entity mention. The examined models are the Transformer models Roberta (Liu et al., 2019), XLNet (Yang, Dai, et al., 2019) and the DCA-SL model (Yang, Gu, et al., 2019). There is no global coherence technique applied. Overall, up to 2-hop triples of any kind are used. For example, labels, aliases, descriptions, or general relations to other entities are all incorporated. It is not mentioned if the hyper-relational structure in the form of qualifiers were used. On the one hand, the purely language-based EL results in less need of retraining if the KG changes. On the other hand, the reliance on the triple information might be problematic for long-tail entities.

The master thesis by Perkins (Perkins, 2020) is performing candidate generation by using anchor link probability over Wikipedia and LSH over labels and mention bi-grams. Contextual word embeddings of the utterance (ELMo (Peters et al., 2018)) are used together with KG embeddings (TransE (Bordes et al., 2013)), calculated over Wikipedia and Wikidata, respectively. The context embeddings are sent through a recurrent neural network. The output is concatenated with the KG embedding and then fed into a feed-forward neural network giving a similarity measure between the KG embedding of the entity candidate and the utterance. The KG is used in the form of the calculated TransE embeddings. Hyper-relational

structures like qualifiers are not mentioned in the thesis and not considered by the TransE embedding algorithm. Thus, probably not included. The used KG embeddings make it necessary to retrain when the Wikidata KG changes as they are not dynamic.

The approach designed by Botha et al. (Botha et al., 2020) tackles multilingual EL. It is also crosslingual. That means, it can link entity mentions to entities in a knowledge graph in a language different to the utterance one. The idea is to train one model to link entities in utterances of 100+ different languages to a KG containing not necessarily textual information in the language of the utterance. While the target KG is Wikidata, they mainly use Wikipedia descriptions as input. This is the case as extensive textual information is not available in Wikidata. But as Wikipedia articles are easily linkable to the corresponding Wikidata entities, gathering the desired textual information is easy. Furthermore, as the Wikidata entities have language-agnostic identifiers, Wikidata is suited to be the target KG. The approach resembles the Wikification method by Wu et al. (Wu, Petroni, et al., 2020) but extends the training process to be multilingual and targets Wikidata. Candidate generation is done via a dual-encoder architecture. Here, two BERT-based Transformer models encode both the context-sensitive mentions and the entities to the same vector space. The mentions are encoded using local context, the mention and surrounding words, and global context, the document title. Entities are encoded by using the Wikipedia article description available in different languages. In both cases, the encoded CLS-token are projected to the desired encoding dimension. The goal is to embed mentions and entities in such a way that the embeddings are similar. The model is trained over Wikipedia by using the anchors in the text as entity mentions. Now, after the model is trained, all entities are embedded. The candidates are generated by embedding the mention and searching for the nearest neighbors. A certain number of neighbors are then the generated candidates. A cross-encoder is employed to rank the entity candidates, fed with the concatenation of the entity description and mention text. Final scores are obtained and the entity mention is linked. Wikidata information is only used to gather all the Wikipedia descriptions in the different languages for all entities. Besides that, one relies mainly on Wikipedia. While that is the case, it is also clear that Wikidata is very suitable as the target KG for multilingual EL as its entities themselves are language-agnostic. The approach was tested on zero- and few-shot settings showing that the model can handle an evolving knowledge base with newly added entities that were never seen before. This is also more easily achievable due to its missing reliance on the graph structure of Wikidata or the structure of Wikipedia. It is the case that some Wikidata entities do not appear in Wikipedia and are therefore invisible to the approach. But this is less problematic here than for other approaches. The model is trained over descriptions of entities in multiple languages. Other approaches only use the English Wikipedia, which misses entities available in other languages. Thus, the amount of available entities is larger.

Entity Recognition and Entity Linking

The following methods all include ER in their EL process.

In 2018, Sorokin and Gurevych (Sorokin and Gurevych, 2018) were doing joint end-to-end ER and EL on short texts. The algorithm tries to incorporate multiple context embeddings into a mention score, signaling if a word is a mention, and a ranking score, signaling the candidate's correctness. First, it generates several different tokenizations of the same utterance. For each token, a search is conducted over all labels in the KG to gather candidate entities. If the token is a substring of a label, the entity is added. Each token sequence gets then a score assigned. The scoring is tackled from two sides. On the utterance side, a token-level context embedding and a character-level context embedding (based on the mention) is computed. The calculation is handled via dilated convolutional networks (DCNN) (Yu and Koltun, 2016). On the KG side, one includes the labels of candidate entity, the labels of relations connected to a candidate entity, the embedding of the candidate entity itself, and embeddings of the entities and relations related to the candidate entity. This is again done by DCNNs and, additionally, by fully connected layers. The best solution is then found by calculating a ranking and mention score for each token for each possible tokenization of the utterance. All those scores are then summed up into a global score. The global assignment with the highest score is then used to select the entity mentions and entity candidates. The approach uses the underlying graph, label and alias information of Wikidata. Graph information is used via connected entities and relations. They also use TransE embeddings, and therefore no hyper-relational structure. Due to the usage of static graph embeddings, retraining will be necessary if Wikidata changes.

OpenTapioca (Delpeuch, 2020) is a mainly statistical EL approach published in 2019. While the paper never mentions ER, the approach was evaluated with it. In the code one can see that the ER is done by a SolrTextTagger analyzer of the Solr search platform[12]. The candidates are generated by looking up if the mention corresponds to an entity label or alias in Wikidata stored in a Solr collection. Entities are filtered out which do not correspond to the type person, location or organization. OpenTapioca is based on two main features, which are local compatibility and semantic similarity. First, local compatibility is calculated via a popularity measure and a unigram similarity measure between entity label and mention. The popularity measure is based on the number of sitelinks, PageRank scores, and the number of statements. Second, the semantic similarity strives to include context information in the decision process. All entity candidates are included in a graph and are connected via weighted edges. Those weights are calculated via a statistical similarity measure. This measure includes how likely it is to jump from one entity candidate to another while discounting it by the distance between the corresponding mentions in the utterance. The resulting adjacency matrix is then normalized to a stochastic matrix that defines a Markov Chain. One now propagates the local compatibility using this Markov Chain. Several iterations are then taken, and a final score is inferred via a Support Vector Machine. It supports multiple entities per utterance. OpenTapioca is only trained on and evaluated for three types of entities: locations, persons, and organizations. It facilitates Wikidata-specific labels, aliases, and sitelinks information. More importantly, it also uses qualifiers of statements in the calculation of the PageRank scores. But the qualifiers are only seen as additional edges to the entity. The usage

12. https://lucene.apache.org/solr/

in special domains is limited due to its restriction to only three types of entities but this is just an artificial restriction. It is easily updatable if the Wikidata graph changes as no immediate retraining is necessary.

Falcon 2.0 (Sakor et al., 2020) is a fully linguistic approach and a transformation of Falcon 1.0 (Sakor et al., 2019) to Wikidata. Falcon 2.0 was published in 2019 and its focus lies on short texts, especially questions. It links entities and relations jointly. Falcon 2.0 uses entity and relation labels as well as the triples itself. The relations and entities are recognized by applying linguistic principles. The candidates are then generated by comparing mentions to the labels using the Levenshtein distance. The ranking of the entities and relations is done by creating triples between the relations and entities and checking if the query is successful. The more successful the queries, the higher the candidate will be ranked. If no query is successful, the algorithm returns to the ER phase and splits some of the recognized entities again. As Falcon 2.0 is an extension of Falcon 1.0 from DBpedia to Wikidata, the usage of specific Wikidata characteristics is limited. Falcon 2.0 is tuned for EL on questions and short texts, as well as the English language. It is thus not very generalizable on longer, more noisy, non-question texts. As it only based on rules it is clearly independent of changes in the KG.

Arjun (Mulang, Singh, Vyas, et al., 2020) tries to tackle specific challenges of Wikidata like long entity labels and implicit entities. Published in 2020, Arjun is an end-to-end approach utilizing the same model for ER and EL. It is based on an Encoder-Decoder-Attention model. First, the entities are detected via feeding Glove (Pennington et al., 2014) embedded tokens of the utterance into the model and classifying each token as being an entity or not. Afterward, candidates are generated in the same way as in Falcon 2.0 (Sakor et al., 2020). The candidates are then ranked by feeding the mention, the entity label, and its aliases into the model and calculating the score. Thus, the model is a similarity measure between the mention and the entity labels. It does not use any global ranking. Wikidata information is used in the form of labels and aliases in the candidate generation and candidate ranking. As it relies purely on labels, it is not that susceptible to changes in the KG.

Hedwig (Klang and Nugues, 2020) is a multilingual entity linker specialized on the TAC 2017 task but published in 2020. Another entity linker (Klang et al., 2019), developed by the same authors, is not included in this survey as Hedwig is partly an evolution of it. The entities to be linked are limited to only a subset of all possible entity classes. Hedwig employs Wikidata and Wikipedia at the same time. The Entity Recgontion uses word embeddings, character embeddings, and dictionary features where the character embeddings are calculated via a Bi-LSTM. The dictionary features are class-dependent, but this is not defined in more detail. Those embeddings and features are computed and concatenated for each token. Afterward, the whole sequence of token features is fed into a Bi-LSTM with a linear chain Conditional Random Field (CRF) layer at the end to recognize the entities. The candidates for each detected entity mention are then generated by using a mention dictionary. The dictionary is created from Wikidata and Wikipedia information, utilizing labels, aliases, titles or anchor texts. The candidates are disambiguated by constructing a graph consisting of all candidate

entities, mentions, and occurring words in the utterance. The edges between entities and other entities, words, or mentions have the normalized pointwise mutual information (NPMI) assigned as their weights. The NPMI specifies how frequent two entities, an entity and a mention or an entity and a word, occur together. Those scores are calculated over a Wikipedia dump. Finally, the PageRank of each node in the graph is calculated via power iteration, and the highest-scoring candidates are chosen. In contrast to DeepType, the type classification is used to determine the types of entities, not mentions. As this is only relevant for the TAC2017 task, the classifier can be ignored. Labels and aliases of multiple languages are used. It also uses sitelinks to connect the Wikidata identifiers and Wikipedia articles. The paper also claims to use descriptions but does not describe anywhere in what way. No hyper-relational or graph features are used. As it employs class-dependent features, it is limited to the entities of classes specified in the TAC 2017 task. The NPMI weights have to be updated with the addition of new elements in Wikidata and Wikipedia.

KBPearl (Lin et al., 2020), published in 2020, utilizes EL to populate incomplete KGs using documents. First, a document is preprocessed via Tokenization, POS tagging, NER, noun-phrase chunking, and time tagging. Also, an existing Information Extraction tool is used to extract open triples from the document. Open triples are non-linked triples in unstructured text. The triples are processed further by filtering invalid tokens and doing canonicalization. Then, a graph of entities, predicates, noun phrases, and relation phrases is constructed. The candidates are generated by comparing the noun/relation phrases to the labels and aliases of the entities/predicates. The edges between the entities/relations and between entities and relations are weighted by the number of intersecting one-hop statements. The next step is the computation of a maximum dense subgraph. Density is defined by the minimum weighted degree of all nodes (Hoffart et al., 2011). As this problem is NP-hard, a greedy algorithm is used for optimization. New entities relevant for the task of Knowledge Graph Population are identified by thresholding the weighted sum of an entity's incident edges. Like used here, global coherence can perform sub-optimally since not all entities/relations in a document are related. Thus, two variants of the algorithm are proposed. First, a pipeline version that separates the full document into sentences. Second, a near neighbor mode, limiting the interaction of the nodes in the graph by the distances of the corresponding noun-phrases and relation-phrases. The approach includes label and alias information of entities and predicates. Additionally, one-hop statement information is used, but hyper-relational features are not mentioned. However, the paper does not claim that its focus is entirely on Wikidata. Thus, the weak specialization is understandable. While it utilizes EL, the focus of the approach is still knowledge base population. No training is necessary which makes the approach suitable for a dynamic graph like Wikidata.

PNEL (Banerjee et al., 2020) is an E2E model jointly solving ER and EL focused on short texts. PNEL employs a Pointer network (Vinyals et al., 2015) working on a set of different features. An utterance is tokenized into multiple different combinations. Each token is extended into the (1) token itself, (2) the token and the predecessor, (3) the token and the successor, and (4) the token with both predecessor and successor. For each token combination, candidates are

searched for by using the BM25 similarity measure. Fifty candidates are used per tokenization combination. Therefore, 200 candidates are found per token. For each candidate, features are extracted. Those range from the simple length of a token to the graph embeddings of the candidate entity. All features are concatenated to a large feature vector. Therefore, per token, a sequence of 200 such features vectors exist. Finally, the concatenation of those sequences of each token in the sentence is then fed into a Pointer network. At each iteration of the Pointer network, it points to one candidate in the network or an `END` token marking no choice. The entity descriptions, labels and aliases are all used. Additionally, the graph structure is included by TransE graph embeddings, but no hyper-relational information was incorporated. E2E models often can improve the performance of the ER. Most EL algorithms employed in industry often use older ER methods decoupled from the EL process. Thus, such an E2E EL approach can be of use. Nevertheless, due to its reliance on static graph embeddings, complete retraining will be necessary if Wikidata changes.

The approach designed by Huang et al. (Huang et al., 2020) is utilizing deep and shallow models together. It specialized in short texts. The ER is performed via a pre-trained BERT model (Devlin et al., 2019) with a single classification layer on top, determining if a token belongs to an entity mention. The candidate search is done via an ElasticSearch[13] index, comparing the entity mention to labels and aliases by exact match and Levenshtein distance. The candidate ranking uses three similarity measures to calculate the final rank. A CNN is used to compute a character-based similarity between entity mention and candidate label. This results in a similarity matrix whose entries are calculated by the cosine similarity between each character embedding of both strings. The context is included in two ways. First, between the utterance and the entity description, by embedding the tokens of each sequence through a BERT model. Again, a similarity matrix is built by calculating the cosine similarity between each token embedding of both utterance and description. The KG is also considered by including the triples containing the candidate as a subject. For each such a triple a similarity matrix is calculated between the label concatenation of the triple and the utterance. All measures are then combined and fed into a two-layer perceptron. Wikidata labels, aliases and descriptions are utilized. Additionally, the KG structure is incorporated through the labels of candidate-related triples. This is similar to the approach by Mulang et al. (Mulang, Singh, Prabhu, et al., 2020), but only 1-hop triples are used. There are also no hyper-relational information considered. Due to its reliance on text alone, it is less susceptible to the changes of Wikidata.

In connection to the *CLEF 2020 HIPE challenge* (Ehrmann et al., 2020), multiple approaches for ER and EL of historical newspapers on Wikidata were developed. Documents were available in English, French and German. Three approaches with a focus on the English language are described in the following. The documents are noisy as the OCR method for transcribing the newspapers produced errors. The authors often constructed different methods for different languages. From now on, only the English models are described. Differences in the usage of Wikidata between the languages did not exist. Yet, the approaches were not multilingual

13. https://www.elastic.co/elasticsearch/

as different models were used and/or a retraining was necessary for different languages.

Boros et al. (Boros et al., 2020) tackled ER by using a BERT model with a CRF layer on top, which recognizes the entity mentions and classifies the type. During the training, the regular sentences are enriched with misspelled words to make the model robust against noise. For the EL, a knowledge base is built from Wikipedia, containing Wikipedia titles, ids, disambiguation pages, redirects and calculating link probability between mentions and Wikipedia pages. The link probability between anchors and Wikipedia pages is used to gather entity candidates for a mention. The disambiguation approach follows an already existing method (Kolitsas et al., 2018). Here, the utterance tokens are embedded via a Bi-LSTM. The token embeddings of a single mention are combined. Then similarity scores between the resulting mention embedding and the entity embeddings of the candidates are calculated. The entity embeddings are computed according to Ganea and Hofmann (Ganea and Hofmann, 2017). These similarity scores are combined with the link probability and long-range context attention, calculated by taking the inner product between an additional context-sensitive mention embedding and an entity candidate embedding. The resulting score is a local ranking measure and is again combined with a global ranking measure considering all other entity mentions in the text. In the end, additional filtering is applied by comparing the DBpedia types of the entities to the ones classified during the ER. If the type does not match or other inconsistencies apply, the entity candidate gets a lower rank. Here, they also experimented with Wikidata types, but this resulted in a performance decrease. As can be seen, technically, no Wikidata information besides the unsuccessful type inclusion is used. Thus, the approach resembles more of a Wikification algorithm. Yet, they do link to Wikidata as the HIPE task dictates it and therefore, the approach was included in the survey. New Wikipedia entity embeddings can be easily added (Ganea and Hofmann, 2017) which is an advantage when Wikipedia changes. Also, its robustness against erroneous texts makes it ideal for real-world use.

Labusch and Neudecker (Labusch and Neudecker, 2020) also applied a BERT model for ER. For EL, they used mostly Wikipedia, similar to Boros et al. (Boros et al., 2020). They built a knowledge base containing all person, location and organization entities from the German Wikipedia. Then it was converted to an English knowledge base by mapping from the German Wikipedia Pages via Wikidata to the English ones. This mapping process resulted in the loss of numerous entities. The candidate generation is done by embedding all Wikipedia page titles in an Approximative Nearest Neighbour index. Using this index, the neighboring entities to the mention embedding are found and used as candidates. For ranking, anchor-contexts of Wikipedia pages are embedded and fed into a classifier together with the embedded mention-context, which outputs whether both belong to the same entity. This is done for each candidate for around 50 different anchor-contexts. Then, multiple statistics on those similarity scores and candidates are calculated, which are used in a Random Forest model to compute the final ranks. Similar to the previous approach, Wikidata was only used as the target knowledge base, while information from Wikipedia was used for all the EL work. Thus, no special characteristics of Wikidata were used. The approach is less

Table 3.8: Results: EL only.

	DeepType [1]	Mulang et al.	LSH-ELMo model	NED using DL on Graphs [2]	Botha et al.
AIDA-CoNLL	0.949 [3]	0.9494 [3,4]	0.73	-	-
ISTEX-1000	-	0.9261 [5]	-	-	-
Wikidata-Disamb	0.924 [3]	0.9235 [6]	-	0.916	-
Mewsli-9	-	-	-	-	0.91 [7]

[1] Only evaluated on Wikipedia
[2] Model with best result
[3] Accuracy instead of F_1
[4] DCA-SL used
[5] XLNet used
[6] Roberta used
[7] Recall instead of F_1

affected by a change of Wikidata due to similar reasons as the previous approach. Also, this approach lacks performance compared to the state of the art in the HIPE task. The knowledge base creation process produces a disadvantageous loss of entities, but this might be easily changed.

Provatorov et al. (Provatorova et al., 2020) used an ensemble of fine-tuned BERT models for ER. The ensemble is used to compensate for the noise of the OCR procedure. The candidates were generated by using an ElasticSearch index filled with Wikidata labels. The candidate's final rank is calculated by taking the search score, increasing it if a perfect match applies and finally taking the candidate with the lowest Wikidata identifier number. They also created three other methods of the EL approach: (1) The ranking was done by calculating cosine similarity between the embedding of the utterance and the embedding of the same utterance with the mention replaced by the Wikidata description. Furthermore, the score is increased by the Levenshtein distance between the entity label and the mention. (2) A variant was used where the candidate generation is enriched with historical spellings of Wikidata entities. (3) The last variant used an existing tool, which included contextual similarity and co-occurrence probabilities of mentions and Wikipedia articles. Also, a global ranking was applied. The approach uses Wikidata labels and descriptions in one variant of candidate ranking. Beyond that, no other characteristics specific to Wikidata were considered. Overall, the approach is very basic and uses mostly pre-existing tools to solve the task. The approach is not susceptible to a change of Wikidata as it is mainly based on language and does not need retraining. However, its poor performance in the HIPE challenge makes it a less desirable method to employ.

Tweeki (Harandizadeh and Singh, 2020) is an approach focusing on unsupervised EL over tweets. The ER is done by a pre-existing Entity Recognizer (Gardner et al., 2018) which also tags the mentions. The candidates are generated by first calculating the link probability between Wikidata aliases over Wikipedia and then searching for the aliases in a dictionary. The ranking is done using the link probabilities while pruning all candidates that do not belong to the type

provided by the Entity Recognizer. It is a relatively simple approach that does not need to be trained, making it very suitable for linking entities in tweets. In that document type, often novel entities with minimal context exist. Regarding features of Wikidata, it uses label, alias and type information. Due to it being unsupervised, changes to the KG do not affect it.

3.5.2 Evaluation

Table 3.8 and Table 3.9 give an overview of all available results for the approaches described in the previous section. The first gives information for EL only approaches and the second for approaches evaluating EL together with ER. The micro F_1 scores are given:

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r}$$

where p is the precision $p = \frac{tp}{tp+fp}$ and r is the recall $r = \frac{tp}{tp+fn}$. tp are here the amount of true positives, fp the amount of false positives and fn the amount of false negatives. Micro F_1 means that the scores are calculated over all linked entity mentions and not separately for each document and then averaged. True positives are the correctly linked entity mentions, false positives incorrectly linked entities which do not occur in the set of valid entities and false negatives entities which occur in the set of valid entities but are not linked to (Cornolti et al., 2013). The approaches were evaluated on many different datasets, which makes comparison very difficult. Additionally, many approaches are evaluated on datasets designed for knowledge graphs different to Wikidata and then mapped. Often, the approaches are evaluated on the same dataset but over different subsets, which complicates a comparison even more. The method by Perkins (Perkins, 2020) was also evaluated on the Kensho Derived Wikimedia Dataset (Kensho R&D group, 2020), but it was only used to compare different variants of the designed approach and focussed on different amounts of training data. Thus, inclusion in the evaluation table is not reasonable.

Inferring the utility of a Wikidata characteristic from the different approaches' F_1-measures is inconclusive due to the sparsity of results. For EL-only, AIDA-CoNLL results are available for three of five approaches, but the results for two are the accuracies instead of the F_1-measures. However, considering the results of Deeptype (Raiman and Raiman, 2018) for Wikidata-Disamb, it becomes apparent that the inclusion of type information might help a lot. Still, it was only used with Wikipedia categories. The available labels for each item and property make language-model-based approaches possible that perform quite well (Mulang, Singh, Prabhu, et al., 2020). No approaches are available to compare to the one by Botha et al. (Botha et al., 2020), but the result demonstrates the promising performance of multilingual EL with Wikidata as the target KG. For ER + EL approaches, most results were available for LC-QuAD 2.0. Yet, no conclusion can be drawn as many approaches were evaluated on different subsets of the dataset. Falcon 2.0 performs well, but it does not substantially rely on Wikidata characteristics. The performance is good as it is designed for simple questions that follow its rules very closely. Arjun performs well on T-REx by mainly using label information, but the amount of methods tested on the T-REx dataset is too low to be conclusive.

Table 3.9: Results: ER + EL.

	OpenTapioca	Falcon 2.0	Arjun	VCG	KBPearl [1]	PNEL	Huang et al.	Boros et al.	Provatorov et al.	Labusch & Neudecker	Hedwig	Tweeki
AIDA-CoNLL	0.482	-	-	-	-	-	-	-	-	-	-	-
Microposts 2016	0.087, 0.148	-	-	-	-	-	-	-	-	-	-	0.248
ISTEX-1000	0.87	-	-	-	-	-	-	-	-	-	-	-
RSS-500	0.335	-	-	-	-	-	-	-	-	-	-	-
LC-QuAD 2.0	0.301	0.445	-	0.47	-	0.589 [2]	-	-	-	-	-	-
LC-QuAD 2.0 [3]	0.25	0.68	-		-	-	-	-	-	-	-	-
LC-QuAD 2.0 [4]	-	0.320	-	-	-	0.629 [2]	-	-	-	-	-	-
Simple-Question	0.20	0.41	-	-	-	0.68 [5]	-	-	-	-	-	-
Simple-Question [6]	-	0.63	-	-	-	-	-	-	-	-	-	-
T-REx	0.579	-	0.713	-	-	-	-	-	-	-	-	-
T-REx [7]	-	-	-	-	0.421	-	-	-	-	-	-	-
WebQSP	-	-	0.730	-	-	0.712 [8]	0.780	-	-	-	-	-
CLEF HIPE 2020	-	-	-	-	-	-	-	0.531 [9]	0.300 [9]	0.141 [9]	-	-
TAC2017	-	-	-	-	-	-	-	-	-	-	0.582	
Graph-Questions	-	-	0.442	-	-	-	-	-	-	-	-	-
QALD-7-WIKI	-	-	-	-	0.679	-	-	-	-	-	-	-
NYT2018	-	-	-	-	0.575	-	-	-	-	-	-	-
ReVerb38	-	-	-	-	0.653	-	-	-	-	-	-	-
Knowledge Net	-	-	-	-	0.384	-	-	-	-	-	-	-
TweekiGold	0.291	-	-	-	-	-	-	-	-	-	-	0.65
Derczynski	0.14	-	-	-	-	-	-	-	-	-	-	0.371

[1] NN model
[2] L model
[3] 1000 sampled questions from LC-QuAD 2.0
[4] LC-QuAD 2.0 test set used in KBPearl paper
[5] S model
[6] Probably evaluated on train and test set
[7] Evaluation on subset of T-REx data different to the subset used in Arjun paper
[8] W model
[9] Strict mention matching

Besides that, PNEL and the approach by Huang et al. also achieve good results; both include a broader scope of Wikidata information in the form of labels, descriptions and graph structure. As HIPE challenge approaches are using Wikidata only marginally and the difference in performance depends more on the robustness against the OCR-introduced noise, comparing them is not providing information on the relevance of Wikidata characteristics.

While some algorithms (Mulang, Singh, Vyas, et al., 2020) do try to examine the challenges of Wikidata, like more noisy long entity labels, many fail to use most of the advantages of Wikidata's structure. If the approaches are using even more information than just the labels of entities and relations, they mostly only include simple n-hop triple information. Hyper-relational information like qualifiers is only used by OpenTapioca but still in a simple manner. This is surprising, as they can provide valuable additional information. As one can see in Figure 3.8, around half of the statements on entities occurring in the LC-QuAD 2.0 dataset have one or more qualifiers. These percentages differ from the ones in all of Wikidata, but when entities are considered, appearing in realistic use cases like QA, qualifiers are much more abundant. Thus, dismissing the qualifier information might be critical. The inclusion of hyper-relational graph embeddings could improve the performance of many approaches already using non-hyper-relational ones. Rank information of statements might be useful to consider, but choosing the best one will probably often suffice.

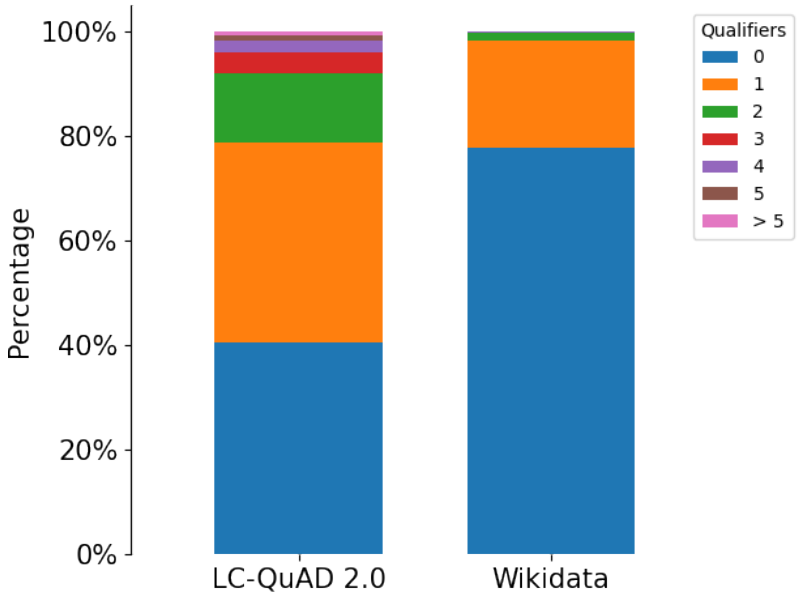


Figure 3.8: Percentage of statements having the specified number of qualifiers for all LC-QuAD 2.0 and Wikidata entities.

Of all approaches, only two algorithms (Banerjee et al., 2020; Huang et al., 2020) use descriptions explicitly. Others incorporate them through triples too, but more on the side (Mulang, Singh, Prabhu, et al., 2020). Descriptions can provide valuable context information and many items do have them; see Figure 3.6d. Hedwig (Klang and Nugues, 2020) claims to use descriptions but fails to describe how. Three approaches (Botha et al., 2020; Klang and Nugues, 2020; Raiman and Raiman, 2018)

demonstrated the usefulness of the inherent multilingualism of Wikidata, notably in combination with Wikipedia.

As Wikidata is always changing, approaches robust against change are preferred. A reliance on transductive graph embeddings (Banerjee et al., 2020; Cetoli et al., 2018; Perkins, 2020; Sorokin and Gurevych, 2018), which need to have all entities available during training, makes repeated retraining necessary. Alternatively, the used embeddings would need to be replaced with graph embeddings, which are efficiently updatable or inductive (Albooyeh et al., 2020; Baek et al., 2020; Hamaguchi et al., 2017; Teru et al., 2020; Wang et al., 2019; Wang, Gao, et al., 2021; Wu, Petroni, et al., 2020). The rule-based approach Falcon 2.0 (Sakor et al., 2020) is not affected by a developing knowledge base but only usable for correctly-stated questions. Methods only working on text information (Huang et al., 2020; Mulang, Singh, Prabhu, et al., 2020; Mulang, Singh, Vyas, et al., 2020) like labels, descriptions or aliases do not need to be updated if Wikidata changes, only if the text type or the language itself does. For approaches (Harandizadeh and Singh, 2020; Huang et al., 2020; Klang and Nugues, 2020) that rely on statistics over Wikipedia, new entities may in Wikidata may sometimes not exist in Wikipedia to a satisfying degree. The approaches by Boros et al. (Boros et al., 2020), and Labusch and Neudecker (Labusch and Neudecker, 2020) are mostly using Wikipedia information. They are, therefore, susceptible to changes in Wikipedia, especially specific statistics calculated over Wikipedia pages. Botha et al. (Botha et al., 2020) also mainly depends on Wikipedia and thus on the availability of the desired Wikidata entities in Wikipedia itself. But as it uses Wikipedia articles in multiple languages, it encompasses many more entities than the previous approaches that focus on Wikipedia. As it was designed for the zero- and few-shot setting, it is quite robust against changes in the underlying knowledge base. Deeptype (Raiman and Raiman, 2018) relies on a fine-grained type system. As the categories of Wikidata are not evolving as fast as novel entities appear, it is relatively robust against a changing knowledge base. However, it was not yet tested on Wikidata, which's type assignments differs vastly from Wikipedia. Statistical approaches (Delpeuch, 2020; Lin et al., 2020) need to update the underlying statistics, but this might be efficiently doable. Overall, the robustness against change is most negatively affected by static/transductive graph embeddings.

This summary and evaluation of the existing Wikidata Entity Linkers answers **RQ 1**.

3.5.3 Reproducibility

Not all algorithms are available as an Web API or even as source code. An overview can be seen in Table 3.10. The amount of approaches for Wikidata having an accessible Web API is meager. While the code for some methods exists, this is still just the case for less than half. The effort to set up the different approaches also varies significantly due to missing instructions or data. Thus, we refrained from evaluating and filling the missing results for all the datasets in Tables 3.8 and 3.9.

Table 3.10: Availability of approaches.

Approach	Code	Web API
OpenTapioca (Delpeuch, 2020)	✓	✓
NED using DL on Graphs (Cetoli et al., 2018)	✓	✗
Falcon 2.0 (Sakor et al., 2020)	✓	✓
Arjun (Mulang, Singh, Vyas, et al., 2020)	✓	✗
DeepType (Raiman and Raiman, 2018)	✓	✗
Hedwig (Klang and Nugues, 2020)	✗	✗
VCG (Sorokin and Gurevych, 2018)	✓	✗
KBPearl (Lin et al., 2020)	✗	✗
PNEL (Banerjee et al., 2020)	✓	✗
Mulang et al. (Mulang, Singh, Prabhu, et al., 2020)	✓	✗
Perkins (Perkins, 2020)	✗	✗
Huang et al. (Huang et al., 2020)	✗	✗
Boros et al. (Boros et al., 2020)	✗	✗
Provatorov et al. (Provatorova et al., 2020)	✗	✗
Labusch and Neudecker (Labusch and Neudecker, 2020)	✗	✗
Botha et al. (Botha et al., 2020)	✗	✗
Tweeki (Harandizadeh and Singh, 2020)	✗	✗

3.6 Datasets

3.6.1 Overview

This section is concerned with analyzing the different datasets which are used for Wikidata EL. A comparison can be found in Table 3.11. Here, information about the purpose, release year, domain and more is given. The majority of datasets on which existing Entity linkers were evaluated, were originally constructed for KGs different from Wikidata. Such a mapping can be problematic as some entities labeled for other KGs could be missing in Wikidata. Or some NIL entities that do not exist in other KGs could exist in Wikidata. Eleven datasets were found for which Wikidata (Botha et al., 2020; Delpeuch, 2020; Dubey et al., 2019; Ehrmann et al., 2020; Elsahar et al., 2018; Harandizadeh and Singh, 2020; Kensho R&D group, 2020; Lin et al., 2020; Mesquita et al., 2019; Noullet et al., 2020) identifiers were available from the start.

LC-QuAD 2.0 (Dubey et al., 2019) is a dataset semi-automatically created for Complex Questions Answering providing complex natural language questions. For each question, Wikidata and DBpedia identifiers are provided. The questions are generated from subgraphs of the Wikidata KG. The dataset does not provide annotated mentions.

T-REx (Elsahar et al., 2018) was constructed automatically over Wikipedia abstracts. Its main purpose is Knowledge Base Population. According to Mulang et al. (Mulang, Singh, Vyas, et al., 2020), this dataset describes the challenges of Wikidata, at least in the form of long, noisy labels, the best.

Table 3.11: Comparison of used datasets.

Dataset	Domain	Year	Purpose	Annotated mentions	Identifiers
ISTEX-1000	Research articles	2019	EL	✓	Wikidata
Wikidata-Disamb (based on Wiki-Disamb30(Ferragina and Scaiella, 2010))	Wikipedia articles	2018 [1]	EL	✗	Wikidata [2]
LC-QuAD 2.0	General complex questions	2019	Question Answering (QA)	✗	DBpedia, Wikidata
T-REx	Wikipedia abstracts	2015	Knowledge Base Population (KBP), Relation Extraction (RE), Natural Language Geneation (NLG)	✓	Wikidata
Knowledge Net	Wikipedia abstracts, biographical texts	2019	KBP	✓	Wikidata
NYT2018	News	2018	EL	✓	Wikidata, DBpedia
KORE50DYWC	News	2019	EL	✓	Wikidata, DBpedia, YAGO, Crunchbase
Kensho Derived Wikimedia Dataset	Wikipedia	2020	Natural Language Processing (NLP)	✓	Wikidata, Wikipedia
CLEF HIPE 2020	Historical newspapers	2020	ER, EL	✓	Wikidata
Mewsli-9	News in multiple languages	2020	Multilingual EL	✓	Wikidata
TweekiData	Tweets	2020	EL	✓	Wikidata
TweekiGold	Tweets	2020	EL	✓	Wikidata

[1] data from 2010

[2] Original dataset on Wikipedia

ISTEX-1000 (Delpeuch, 2020) is a research-focused dataset containing 1000 author affiliation strings. It was manually annotated to evaluate the OpenTapioca entity linker.

KnowledgeNet (Mesquita et al., 2019) is a Knowledge Base Population dataset with 9073 manually annotated sentences. The text was extracted from biographical documents from the web or Wikipedia articles.

NYT2018 (Lin and Chen, 2019; Lin et al., 2020) consists of 30 news documents that were manually annotated on Wikidata and DBpedia. It was constructed for KBPearl, so its main focus is also KBP which is a downstream task of EL.

One dataset, KORE 50 DYWC (Noullet et al., 2020), was found, which was not used by any of the approach papers. It is an annotated EL dataset based on the KORE50 dataset, a manually annotated subset of the AIDA corpus. All sentences are reannotated with DBpedia, Yago, Wikidata and Crunchbase entities.

The Kensho Derived Wikimedia Dataset (Kensho R&D group, 2020) is an automatically created condensed subset of Wikimedia data. It consists of three levels: Wikipedia text, annotations with Wikipedia pages and links to Wikidata items. Thus, mentions in Wikipedia articles are annotated with Wikidata items. However, as some Wikidata items do not have a corresponding Wikipedia page, the annotation is not exhaustive. It was constructed for NLP in general.

Table 3.12: Ambiguity of mentions (existence of a match does not correspond to a correct match).

Dataset	Average number of matches	No match	Exact match	More than one match
ISTEX-1000 (train)	23.23	8.06%	26.34%	65.61%
ISTEX-1000 (test)	25.85	10.30%	23.88%	65.82%
Wiki-Disamb30 (train)	25.06	0.36%	1.26%	98.38%
Wiki-Disamb30 (dev)	30.39	0.40%	1.18%	98.42%
Wiki-Disamb30 (test)	30.18	0.30%	1.44%	98.26%
Knowledge Net (train)	21.90	10.41%	22.29%	67.3%
T-REx	4.79	31.36%	32.98%	35.65%
KORE50DYWC	28.31	3.93%	7.49%	88.60%
Kensho Derived Wikimedia Dataset	8.16	35.18%	30.94%	33.88%
CLEF HIPE 2020 (en, dev)	24.02	35.71%	11.51%	52.78%
CLEF HIPE 2020 (en, test)	17.78	43.82%	6.74%	49.44%
Mewsli-9 (en)	11.09	16.80%	34.90%	47.30%
TweekiData	19.61	19.98%	12.01%	68.01%
TweekiGold	16.02	7.41%	20.25%	72.34%

CLEF HIPE 2020 (Ehrmann et al., 2020) is a dataset based on historical newspapers in English, French and German. Only the English dataset will be analyzed in

the following. This dataset is of great difficulty due to many errors in the text, which originates from the OCR method used to parse the scanned newspapers. For the English language, only a development and test set exist. In the other two languages, a training set is also available. It was manually annotated.

Mewsli-9 (Botha et al., 2020) is a multilingual dataset automatically constructed from WikiNews. It includes nine different languages. A high percentage of entity mentions in the dataset do not have corresponding English Wikipedia pages, and thus, cross-lingual linking is necessary.

TweekiData and TweekiGold (Harandizadeh and Singh, 2020) are an automatically annotated corpus and a manually annotated dataset for EL over tweets. TweekiData was created by using other existing tweet-based datasets and linking them to Wikidata data via the Tweeki EL. TweekiGold was created by an expert, manually annotating tweets from another dataset with Wikidata identifiers and Wikipedia page-titles.

Table 3.13 shows the number of documents, the number of mentions, emerging entities and unique entities, and the mentioned ratio. What classifies as a document in a dataset depends on the dataset itself. For example, for T-REx, a document is a whole paragraph of a Wikipedia article, while for LC-QuAD 2.0, a document is just a single question. Due to this, the average amount of entities in a document also varies, e.g., LC-QuAD 2.0 with 1.47 entities per document and T-REx with 11.03. If a dataset was not available, information from the original paper was included. If dataset splits were available, the statistics are also shown separately. The majority of datasets do not contain emerging entities. For the Tweeki datasets, it is not mentioned which Wikidata dump was used to annotate. For a dataset that contains emerging entities, this is problematic. On the other hand, the dump is specified for the CLEF HIPE 2020 dataset, making it possible to work on the Wikidata version with the correct entities missing.

To get an overview how widespread they datasets are in use, see the section 3.5.2. Thus, **RQ 3** is answered.

3.6.2 Evaluation

The difficulty of the different datasets was measured by the accuracy of a simple EL method (Table 3.14) and the ambiguity of mentions (Table 3.12). The simple EL method searches for entity candidates via an ElasticSearch index, including all English labels and aliases. It then disambiguates by taking the one with the largest tf-idf based BM25 similarity measure score and the lowest Q-identifier number resembling the popularity. Nothing was done to handle inflections.[14] Here, only datasets were included which were accessible. As one can see, is the accuracy positively correlated with the number of exact matches. The more ambiguous the underlying entity mentions are, the more inaccurate a simple similarity measure between label and mention becomes. In this case, more context information is necessary. The simple Entity Linker was only applied to datasets that were feasible

14. All source code, plots and results can be found on https://github.com/cedricm-research/ELEnglishWD

Table 3.13: Comparison of the datasets with focus on the number of documents and Wikidata entities.

Dataset	# documents	# mentions	Emerging entities	Wikidata entities	Unique Wikidata entities	Mentions per document
ISTEX-1000 (train)	750	2073	0%	100%	53.7%	2.76
ISTEX-1000 (test)	250	670	0%	100%	65.8%	2.68
Wikidata-Disamb (train)	100,000	100,000	0%	100%	27.2%	1.0
Wikidata-Disamb (test)	10,000	10,000	0%	100%	57.3%	1.0
Wikidata-Disamb (dev)	10,000	10,000	0%	100%	56.2%	1.0
LC-QuAD 2.0	6046	44,529	0%	100%	51.2%	1.47
T-REx	4,650,000	51,297,484	0%	100%	9.1%	11.03
Knowledge Net (train)	3977	13,039	0%	100%	30%	3.28
NYT2018	30	-	-	-	-	-
KORE50DYWC	50	307	0%	100%	72.0%	6.14
Kensho Derived Wikimedia Dataset	14,255,258	121,835,453	0%	100%	3.7%	8.55
CLEF HIPE 2020 (en, dev)	80	470	46.4%	53.6%	31.9%	5.88
CLEF HIPE 2020 (en, test)	46	134	33.6%	66.4%	42.5%	2.91
Mewsli-9 (en)	12,679	80,242	0%	100%	48.2%	6.33
TweekiData	5,000,000	5,038,870	61.2%	38.8%	5.4%	1.01
TweekiGold	500	958	11.1%	88.9%	66.6%	1.92

to disambiguate in that way. T-REx and the Kensho Derived Wikimedia Dataset were too large. According to the EL performance, ISTEX-1000 is the easiest dataset. Many of the ambiguous mentions reference the most popular one, while also many exact unique matches exist. T-REx, the Kensho Derived Wikimedia Dataset and the Mewsli-9 training dataset have the largest percentage of exact matches for labels. The largest number of ambiguous mentions have the Wiki-Disamb30 datasets, resulting in a low EL but not the lowest accuracy. Deciding on the most prominent entity appears to produce good EL results. This is also the case for the TweekiGold dataset. While the KORE50DYWC dataset is less ambiguous than Wiki-Disamb30, it performs the worst due to references to unpopular entities. The CLEF HIPE 2020 dataset also has a low EL accuracy but not due to ambiguity but many mentions with no exact match. The reason for that is the noise created by OCR. Only the English dataset was examined. The second column of Table 3.14 specifies the accuracy with all unique exact matches removed. This is based on the intuition that exact matches without any competitors are usually correct. In general, the removal does decrease the accuracy with one exception. The Wiki-Disamb30 datasets constantly achieve better accuracy as a large percentage of the unique exact matches appear to point to wrong entities. Thus, the true entity does not have the label it is referenced by.

Table 3.14: EL accuracy - Kensho Derived Wikimedia Dataset, T-REx and TweekiData are not included due to size, **Acc. filtered** has all exact matches removed.

Dataset	Acc.	Acc. filtered
ISTEX-1000 (train)	0.744	0.716
ISTEX-1000 (test)	0.716	0.678
Wiki-Disamb30 (train)	0.597	0.600
Wiki-Disamb30 (dev)	0.580	0.584
Wiki-Disamb30 (test)	0.576	0.580
Knowledge Net (train)	0.371	0.285
KORE50DYWC	0.225	0.187
CLEF HIPE 2020 (en, dev)	0.333	0.287
CLEF HIPE 2020 (en, test)	0.258	0.241
TweekiGold	0.565	0.520
Mewsli-9 (en)	0.602	0.490

Two main characteristics of Wikidata may affect the design of Wikidata EL datasets. First, multilingualism is the main focus of Wikidata, and thus, multilingual datasets should also be a focus. Unfortunately, only two datasets (Botha et al., 2020; Ehrmann et al., 2020) focus on the multilingualism of Wikidata. The CLEF HIPE 2020 dataset is designed for Wikidata and has documents for the languages English, French and German, but each language has a different corpus of documents. The same is the case for the Mewsli-9 dataset, while here, documents in nine languages are available. A dataset similar to VoxEL (Rosales-Méndez et al., 2018), which is defined for Wikipedia, would be helpful. Here, each utterance is translated into multiple languages, which eases the comparison of the multilingual EL performance. Having the same corpus of documents in different languages would

allow a better comparison of a method's performance in various languages. Of course, such translations will never be perfectly comparable.

The second characteristic is the large rate of change of Wikidata. Thus, it would also be advisable that the datasets specify the Wikidata dumps they were created, similar to Petroni et al. (Petroni et al., 2021). Many of the existing datasets do that, yet not all. In current dumps, entities, which were available while the dataset was created, could have been removed. It is even more probable that emerging entities could now have a corresponding entity in an updated Wikidata dump version. If the EL approach now would detect it as an emerging entity, it is evaluated as correct, but in reality, this is false and vice versa. Concerning emerging entities, another variant of an EL dataset could be useful. Two Wikidata dumps from different time points could be used to label the utterances. Such a dataset would be valuable in the context of Knowledge Graph Population when emerging entities are inserted into the KG. With the true emerging entity available, one could measure the quality of the insertion. Also, constraining that the method needs to perform well on both KG dumps would force EL approaches to be less reliant on a fixed graph structure. This answers **RQ 4**.

3.7 Related work

While there are multiple recent surveys on EL, none of those are specialized in analyzing the area of EL on Wikidata.

The extensive survey by Sevgili et al. (Sevgili et al., 2019) is giving an overview of all neural approaches from 2015 to 2020. It compares 30 different approaches on nine different datasets. Of those, only Deeptype can be seen as focused on Wikidata. The survey also discusses the current state of the art of domain-independent and multi-lingual neural EL approaches. However, the influence of the underlying KG was not of concern to the authors. It is not described in detail how they found the considered approaches.

In the survey by Al-Moslmi et al. (Al-Moslmi et al., 2020), the focus lies on ER and EL approaches over KGs in general. It considers approaches from 2014 to 2019. It gives an overview of the different approaches of ER, Entity Disambiguation, and EL. A distinction between Entity Disambiguation and EL is made, while our survey sees Entity Disambiguation as a part of EL. The roles of different domains, text types, or languages are discussed. The authors considered 89 different approaches and tools. Most approaches were designed for DBpedia or Wikipedia, some for Freebase or YAGO, and some to be KG-agnostic. Again, the only Wikidata contender was Deeptype. F_1 scores were gathered on 17 different datasets. Fifteen algorithms, for which an implementation or a WebAPI was available, were evaluated using GERBIL (Röder et al., 2018).

Another survey (Oliveira et al., 2020) examines recent approaches, which employ holistic strategies. Holism in the context of EL is defined as the usage of domain-specific inputs and metadata, joint ER-EL approaches and collective disambiguation methods. Thirty-six research articles were found which had any holistic aspect - none of the designed approaches linked explicitly to Wikidata.

A comparison of the number of approaches and datasets included in the different surveys can be found in Table 3.15.

If we go further into the past, the existing surveys (Ling et al., 2015; Shen et al., 2015) are not considering Wikidata at all or only in a small amount as it is still a rather recent KG in comparison to the other established ones like DBpedia, Freebase or YAGO. For an overview on different KGs on the web, we refer the interested reader to the one by Heist et al. (Heist et al., 2020).

No found survey focused on the differences of EL over different knowledge graphs, respectively, on the particularities of EL over Wikidata.

Table 3.15: Survey Comparison

Survey	# Approaches	# Wikidata Approaches	# Datasets	# Wikidata Datasets
Sevgili et al. (2019)	30	1	9	0
Al-Moslmi et al. (2020)	39	1	17	0
Oliveira et al. (2020)	36	0	32	0
This survey	17	17	21	11

3.8 Discussion

3.8.1 Current Approaches, Datasets and their Drawbacks

Approaches. The number of algorithms using Wikidata is small; the number of algorithms using Wikidata solely is even smaller. Most algorithms employ labels and alias information contained in Wikidata. Some deep learning-based algorithms leverage the underlying graph structure, but the inclusion of that information is often superficial. The same information is also available in other KGs. Additional statement specific information like qualifiers is used by only one algorithm (OpenTapioca), and even then, it only interprets qualifiers as extra edges to the item. Thus, there is no inclusion of the actual structure of a hyper-relation. Information like the descriptions of items which are providing valuable context information is also used seldom. Wikidata includes type information, but almost none of the existing algorithms utilize it to do more than to filter out entities that are not desired to link in general. An exception is Tweeki, which uses it together with ER, and perhaps DeepType, though the evaluated model used Wikipedia categories.

One could claim that the current algorithms are mostly trying to map algorithms also usable on other KGs to Wikidata. Besides utilizing specific characteristics of Wikidata, it is also notable that there is no clear focus on one of the essential characteristics of Wikidata, the continual growth. Many approaches use static graph embeddings, which need to be retrained if the KG changes. EL algorithms working on Wikidata, which are not usable on future versions, seem unintuitive. But there also exist some approaches which can handle change. They often rely on

more extensive textual information, which is again challenging due to the limited amount of such data in Wikidata. Wikidata descriptions do exist, but only short paragraphs are provided, in general, insufficient to train a language model. To compensate, Wikipedia is included, which provides this textual information. It seems like Wikidata as the target KG with its language-agnostic identifiers and the easily connectable Wikipedia with its multilingual textual information are the perfect pair.

Most of the approaches tried to use Wikidata due to it being up to date while not utilizing its structure. With small adjustments, many would also work on any other KG. None of the investigated approaches tried to examine the performance between different versions of Wikidata. As continuous evolution is a central characteristic of Wikidata, a temporal analysis would be reasonable.

This survey aimed to identify the extent to which the current state of the art in Wikidata EL is utilizing the characteristics of Wikidata. As only a few are using more information than on other established KGs, there is still much potential for future research.

Datasets. Only a limited amount of datasets were created entirely with Wikidata in mind exist. Many datasets used are still only mapped versions of datasets created for other knowledge bases. Multilingualism is present so far that some datasets contain documents in different languages. However, only different documents for different languages are available. Having the same documents in multiple languages would be more helpful for an evaluation of multilingual Entity Linkers. The fact that the Wikidata is ever-changing is also not genuinely considered in any datasets. Always providing the dump version on which the dataset was created is advisable. Great is that datasets from very different domains like news, forums, research, tweets exist. The utterances can also vary from shorter texts with only a few entities to large documents with many entities. The difficulty of the datasets significantly differs in the ambiguity of the entity mentions. The datasets also differ in quality. Some were automatically created and others annotated manually by experts. There are no unanimously agreed upon datasets used for Wikidata EL. Of course, a single dataset can not exist as different domains and text types make different approaches, and hence datasets necessary.

3.8.2 Future Research Avenues

In general, Wikidata EL could be improved by including:

- Hyper-relational statements which provide additional information
- Type information for more than limiting the candidate space
- Inductive or efficently trainable knowledge graph embeddings
- Item label and description information in multiple languages for multilingual EL

The qualifier and rank information of Wikidata could be also suitable to do EL on time-sensitive utterances (Agarwal et al., 2018). The problem evolves around

utterances which talk about entities from different time points and spans and thus, the referred entity can significantly diverge.

The usefulness of other characteristics of Wikidata, e.g., references, may be limited but could make EL more challenging due to the inclusion of contradictory information. Therefore, research into the consequences and solutions of conflicting information would be advisable.

To reiterate, due to the fast rate of change of Wikidata, approaches are necessary, which are more robust to such a dynamic KG. Continuously retraining transductive embeddings is intractable, so more sophisticated methods like inductive or efficiently retrainable graph embeddings are a necessity.

Multilingual or cross-lingual EL is already tackled with Wikidata but currently mainly by depending on Wikipedia. Using the available multilingual label/description information in a structured form together with the rich textual information in Wikipedia could move the field forward.

It seems like there exist no commonly agreed on Wikidata EL datasets as shown by a large number of different datasets the approaches were tested on. Such datasets should try to represent the challenges of Wikidata like the time-variance, contradictory triple information, noisy labels, and multilingualism.

4

Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering using only Knowledge Graphs

Bibliographic Information

Cedric Möller and Ricardo Usbeck. 2024. Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering Using Only Knowledge Graphs. In *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands,* edited by Angelo A. Salatino, Mehwish Alam, Femke Ongenae, Sahar Vahdati, Anna Lisa Gentile, Tassilo Pellegrini, and Shufan Jiang, 60:88–105. Studies on the Semantic Web. IOS Press

Abstract

Entity Linking is crucial for numerous downstream tasks, such as question answering, knowledge graph population, and general knowledge extraction. A frequently overlooked aspect of entity linking is the potential encounter with entities not yet present in a target knowledge graph. Although some recent studies have addressed this issue, they primarily utilize full-text knowledge bases or depend on external information such as crawled webpages. Full-text knowledge bases are not available in all domains and using external information is connected to increased effort. However, these resources are not available in most use cases. In this work, we solely rely on the information within a knowledge graph and assume no external information is accessible.

To investigate the challenge of identifying and disambiguating entities absent from the knowledge graph, we introduce a comprehensive silver-standard benchmark dataset that covers texts from 1999 to 2022. Based on our novel dataset, we develop an approach using pre-trained language models and knowledge graph embeddings without the need for a parallel full-text corpus. Moreover, by assessing the influence of knowledge graph embeddings on the given task, we show that implementing a sequential entity linking approach, which considers the whole sentence, can outperform clustering techniques that handle each mention separately in specific instances.

4.1 Introduction

Entity Linking (EL) [15] is an essential part of numerous downstream tasks, such as question answering (Lan et al., 2021), knowledge graph population (Ji and Grishman, 2011) or relation extraction (Ji et al., 2022). Yet, EL is still accompanied by several challenges. The main problem is the **ambiguity of entity mentions**. If multiple entities in a knowledge graph (KG) can be referred to by the same name, deciding on the correct one becomes increasingly difficult. The inclusion of context information in the KG or in the input text is usually employed to solve this.

A second problem, only rarely thoroughly considered is the possibility of out-of-KG entities. These are entities that are referred to in the input text but do not actually exist in the KG yet. [16] Consider for example the news message "The President of the Japan Football Association and deputy Olympic Committee chief Kozo Tashima tests positive for COVID-19. Japan insists the 2020 Summer Olympics will still go ahead as planned.". This message is from the beginning of 2020. The mentioned entity "COVID-19" might not yet have existed in a target KG. Hence, an entity linker might link it to a different (likely coronavirus-related) and thus incorrect entity. Most methods ignore this case by assuming that all mentions truly refer to an entity in the KG.

We developed an integrated method that can identify and cluster **out-of-KG entities**. While some methods do exist which consider this task jointly, they either rely on external information (Hoffart et al., 2014; Wu et al., 2016; Zhang et al., 2019) such as crawled webpages or exclusively focus on encyclopedias such as Wikipedia as the underlying knowledge base (Blissett and Ji, 2019; Cassidy et al., 2011; Dutta and Weikum, 2015; Fahrni et al., 2013; Graus et al., 2012; Greenfield et al., 2016; Huynh et al., 2013; Monahan et al., 2011; Tamang et al., 2012). We want to offer an alternative utilizing purely KGs (Singhal, 2012), in our case Wikidata, without using any external data.

Our contributions are as follows:

15. Note that in literature there is a separation between Entity Linking and Entity Disambiguation. The latter means only the disambiguation part and not the entity mention recognition part. According to this difference, we target Entity Disambiguation in this work, however, refer to it as Entity Linking as used in many computational linguistics communities.

16. We abbreviate entities not in the KG as out-of-KG and entities in the KG as in-KG.

1. A novel, openly available Entity Linking dataset containing out-of-KG entities;
2. A sequential Entity Linking method supporting
   (a) Detection of out-of-KG entities;
   (b) Clustering of out-of-KG entities;

4.2 Method

4.2.1 Problem definition

Given a document $d = (t_1, t_2, \dots, t_n)$ represented as a sequence of tokens t_i, where each token can be classified as part of a mention m_j or not, the objective is to construct a mapping function $f : M \to E$ that accurately associates each mention $m_j \in M$ to its corresponding entity $e_k \in E$. Entities can either be present in the Knowledge Graph (KG) or absent from it. For mentions referring to entities not in the KG, the aim is to associate them with one another.

Formally, let $M = \{m_1, m_2, \dots, m_p\}$ be the set of mentions in the document, and $E = E_{\text{in-KG}} \cup E_{\text{out-of-KG}}$ be the set of entities, where $E_{\text{in-KG}}$ is the set of entities in the KG and $E_{\text{out-of-KG}}$ is the set of entities not in the KG. The goal is to find an optimal mapping function f such that:

1. For each mention m_j referring to an entity in the KG, $f(m_j) = e_k \in E_{\text{in-KG}}$.
2. For each mention m_j referring to an entity not in the KG, $f(m_j) = e_k \in E_{\text{out-of-KG}}$, and all mentions referring to the same entity outside the KG are assigned to the same e_k.

4.2.2 Candidate Generation

As we focus on a KG-only use case, we can not rely on an existing entity mention dictionary as utilized in other EL works (Le and Titov, 2018). [17] Hence, we can only use information from the KG. This mainly restricts us to labels and aliases existing for each entity. Thus, for candidate generation, we fill an ElasticSearch index with all labels and aliases. We query this index using a combination of TF-IDF and fuzzy search to compensate for less frequent words and possible typos or other small variations. We retrieve a candidate set of size 100, giving us an acceptable candidate recall. Due to this process, our method does not rely on a parallel text corpus.

4.2.3 Entity Linker

The entity linker is a bi-encoder together with an additional ranking model. A bi-encoder was chosen instead of a cross-encoder as it assumed that one needs

17. An exception in our work is the evaluation of the AIDA-CoNLL dataset. As many EL works utilize the existing candidate set by Le and Titov (Le and Titov, 2018), we also relied on it to be able to compare and verify the performance in a replicable way.

to link against a large number of candidates. This is necessary to guarantee a large enough recall in the candidate generation. A cross-encoder is deemed too expensive in such a case as one would need to encode the text together with the entity candidate of each mention multiple times. In a bi-encoder, the text and all candidates are encoded separately. It consists of a mention encoder and an entity encoder. Figure 4.1 depicts the model architecture.

Mention Encoder

The mention encoder is based on a pre-trained RoBERTa model.[18] For efficiency, we opt for fine-tuning only bottleneck adapters (Houlsby et al., 2019) instead of the whole model. The input to the model is the tokenized input text. All embedded tokens of each entity are averaged and taken as their embedded representation. Furthermore, the embedded representation is scaled via a linear layer to project it to the same space as the entity embedding space. The final embedded vector is defined as e_m.

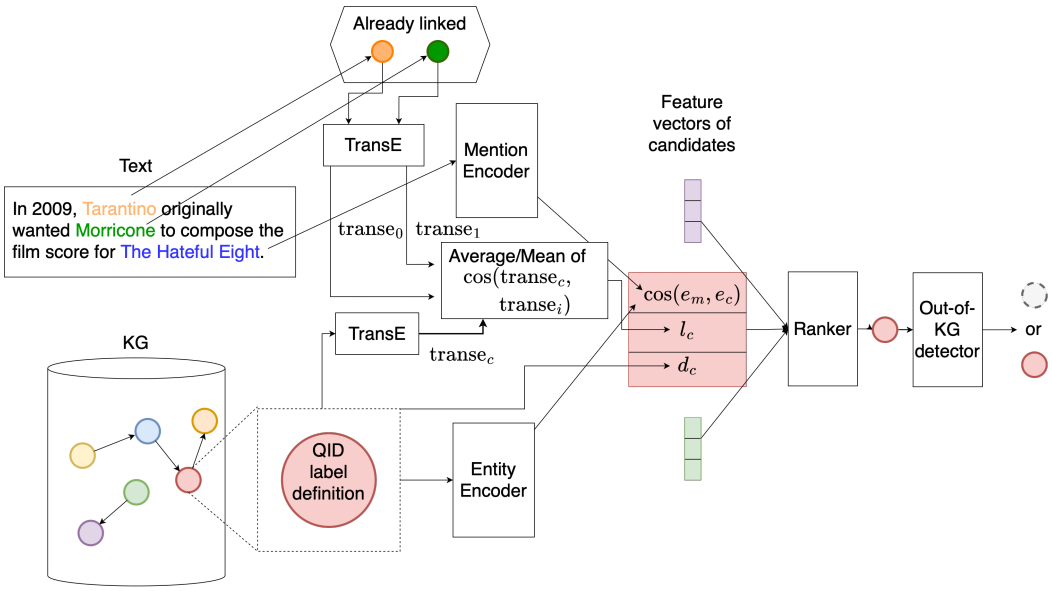


Figure 4.1: First stage - Entity linking and out-of-KG detection of the entity mention "The Hateful Eight". The mention is encoded and compared against the entity encoding. The out-degree of the candidate entities are retrieved. Furthermore, the KG embedding of the candidate is compared against already linked entities. All features are fed into a ranker which determines the correct candidate or detects the mention as out-of-KG. The different colors represent different entities.

Entity Encoder

The entity encoder creates a latent representation of the KG entity by embedding its definition. We define the definition of each entity as the value of a `schema:description` triple in Wikidata. Note that we use the term *definition* instead of *description* here as entity linking methods often refer to the first paragraph

18. We chose RoBERTa-base due to its improved performance over BERT and resource reasons (Liu et al., 2019).

of Wikipedia articles as descriptions. These are much longer than the short descriptions in Wikidata.[19] The definition embedding is generated by encoding a concatenation of the entity's main label and definition and feeding it into an adapter-equipped RoBERTa model. The embedded vector of the [CLS] token is taken as the representation of the entity and projected to the same vector space as the mention embedding. The final definition vector of a node c is denoted as e_c.

We consider three additional features for each entity: First, the popularity of a node d_c is measured by the number of outgoing edges of the entity in the KG. The second feature is the TransE-embedded vector transe_c of the node (Bordes et al., 2013). And the last feature is the type information of an entity.

Ranker

After computing all mention encodings and all assigned candidate entity encodings, as well as their features, it is necessary to combine and rank them. The highest-ranked entity will be the one to which we link.

Linking In-KG candidates The final ranker is a linear layer combining all the aforementioned features. For each candidate-mention-pair, the following inputs are fed into a linear layer.

1. Cosine similarity $\cos(e_c, e_m) = \frac{<e_c, e_m>}{\|e_c\|_2 \|e_m\|_2}$ ($< \cdot, \cdot >$ represents the dot product) between mention embedding e_m and entity definition embedding e_c
2. Node popularity in the form of out-degree d_c
3. Average cosine similarity of the candidate TransE embedding to the TransE embeddings of past linking decisions $l_c = \overline{\cos(\text{transe}_c, \text{transe}_i)} | i \in D$ where D is the set of the past linked entity identifiers

The final logits are calculated as follows (with $\oplus$ denoting vector concatenation):

$$r_c = \text{Linear}(\cos(e_c, e_m) \oplus d_c \oplus l_c)$$

Note that the fourth feature utilizes the TransE embeddings of past linking decisions which introduces *sequentiality*. This implies that each linking decision is influenced by the preceding decision within the same document.

out-of-KG decision The out-of-KG entity detection decision is determined by looking at the maximum-scored entity candidate and deciding whether it is similar enough to the mention. If not, it is an out-of-KG entity. During training, we rely on softmax, to consider all candidates. First, $\sigma(r_c)$ over all candidates is calculated and then multiplied with the feature concatenation $\frac{<e_c, e_m>}{\|e_c\|_2 \|e_m\|_2} \oplus d_c \oplus l_c$ of all candidates to get an accumulated feature vector a. σ stands here for the softmax operation over all candidates. This vector is fed through another single-layer network to get an additional scalar:

$$r_{\text{out-of-KG}} = \text{Linear}(a)$$

19. Of course, the model is compatible with long descriptions but we wanted to focus on the difficulties of only using the information in a KG.

By introducing this additional decision, we are able to detect out-of-KG entities directly without relying on a validation dataset to tune a threshold. Also, it is not necessary to train the model with actual out-of-KG entities. During training, the model is trained by randomly including or excluding the true candidate from the candidate list.

4.2.4 Clustering out-of-KG entities

A detected out-of-KG entity is represented by its mention embedding e_m, and its surrounding linked entities. To identify whether two out-of-KG entities refer to each other, we apply a linear layer to two features: 1) the cosine similarity between both entity embeddings, and 2) the mean cosine distance between the TransE embeddings of the linked entities that surround the mentions. To obtain informative cosine similarities, we further optimize the model to return larger cosine similarities for mentions pointing to the same entity and smaller cosine similarities for mentions pointing to different entities. It is trained via cross-entropy loss where negative mentions are all other mentions in the same batch not referring to the same entity. Using the pairwise scores output by the linear layer, we cluster all out-of-KG detected entities via DBSCAN clustering.[20] The process is depicted in Figure 4.2.

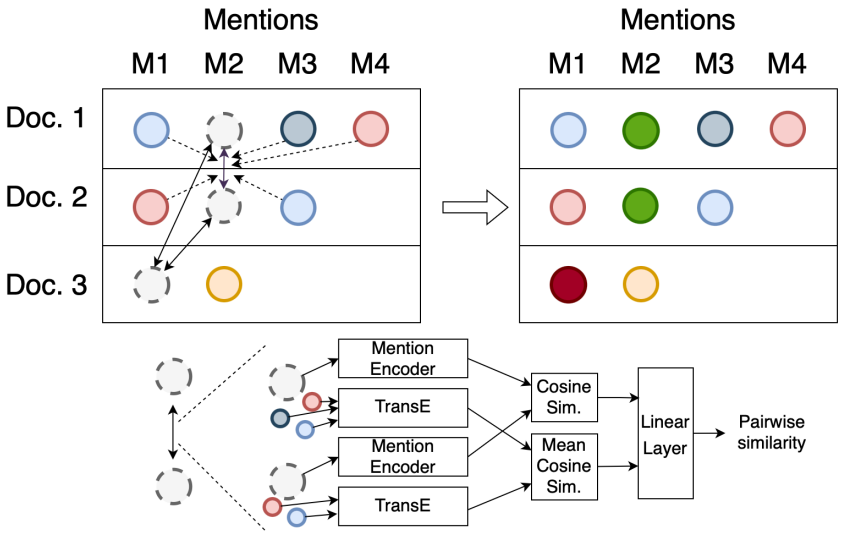


Figure 4.2: Second stage - Clustering the out-of-KG detected entities. Left: Mentions before clustering with three ambiguous mentions. Right: Mentions after clustering with one pair grouped together (green) and another being a singleton (red). Circles with dotted borders illustrate out-of-KG entity mentions. Dotted arrows signalize the impact of already linked entities on the similarity measure between mentions. Note that not all dotted arrows are drawn to simplify the figure.

20. We also evaluated agglomerative average-linkage, maximum-linkage and single-linkage clustering but achieved worse results.

4.2.5 Training

The final loss function consists of multiple aggregated separate loss functions. For entity linking, mention-mention similarity and mention-entity similarity cross-entropy loss was employed. All losses are aggregated equally to return the final loss. Early stopping with a patience of 10, warm-up and a linear learning rate decay are employed. Due to the size of Wikidata, training our own embeddings was intractable. Hence, we opted for the trained set provided by PyTorch-BigGraph (Lerer et al., 2019). The model was trained for 30 epochs on a single NVIDIA RTX A6000 machine.

4.2.6 Inference

When the TransE embeddings are used as a feature in the ranker, beam search with 10 beams is employed to identify the best sequence of linked entities. We evaluated different window sizes for the included surrounding entities in both, the sequential linking and the out-of-KG entity clustering, and determined a window size of 6 as the best.

4.3 Experiments

The experiments are split into three parts. First, we evaluated the performance of the chosen entity linking method in regard to the used features. The best-performing model was then used further. Secondly, we evaluated the impact of the out-of-KG detection mechanism. After the best model was determined, the capability of methods to not only detect out-of-KG entities but also cluster them is examined.

4.3.1 Methods

We compare our sequential method to three different clustering-based methods which cluster all mentions and entities at once. They are the state-of-the-art NASTyLinker by Heist et al. (Heist and Paulheim, 2023) (denoted NASTyLinker), the top-down clustering approach by Kassner et al. (Kassner et al., 2022) (denoted Edin) and the bottom-up clustering approach by Agarwal et al. (Agarwal et al., 2022) (denoted bottom-up).[21] All clustering methods are evaluated by using the trained bi-encoders for computing the similarities between the mentions and the ranker for the mention-candidate similarities.

4.3.2 Datasets

To examine how to handle out-of-KG entities in the task of EL, we created an entity linking dataset from the current-events page of Wikipedia [22], dubbed

21. Note that the original NASTyLinker used an additional cross-encoder. While we do not, a cross-encoder is orthogonal to our changes and can be incorporated in our method as well. By NASTyLinker we refer here to the employed clustering method, not the mention-entity scoring mechanism.

22. https://en.wikipedia.org/wiki/Portal:Current_events

Table 4.1: Statistics of Wikievents dataset

	train	dev	test	overall
# examples	63,623	11,205	11,206	86,034
# mentions	185,039	32,652	32,660	250,351
# out-of-KG mentions	0	2,579	2,519	5,098
# unique entities	38,066	9,349	9,386	45,655
# unique out-of-KG entities	0	751	734	1,221
Average of # mentions per example	2.9	2.91	2.91	2.9

`Wikievents`. On the current-events page, short news snippets stating recent events are available. These texts contain hyperlinks to articles in Wikipedia. We crawled the current-events page texts between 1999-12-29 and 2022-10-01. Each hyperlink was identified and taken as an entity mention. The corresponding page title of the Wikipedia article was mapped to the Wikidata QID. Furthermore, all entity mentions retrieved were filtered further by only keeping those which are `instance of` (P31) of some class and were no `subclass of` (P279) of any other class. The data was split into a train, development, and test set according to the ratios (0.74, 0.13, 0.13). The cutoff date for the knowledge graph and the examples of the development and test sets are 2019-01-28. The development and test sets are created by randomly splitting all examples after the cutoff date. The statistics of the dataset can be found in Table 4.1. Note, the training dataset contains no out-of-KG entities as the included texts all are from before the cutoff date.

Also, three examples from the dataset can be seen in Figure 4.3.

Additionally, we artificially added out-of-KG entities to the well-known `AIDA-CoNLL` dataset (Hoffart et al., 2014). This dataset was chosen as it is a popular dataset in the entity linking domain. The original dataset had only links to Wikipedia which we mapped to Wikidata. out-of-KG entities were added by gathering all occurring entities and randomly selecting 10% of all entities that occurred and declaring them out-of-KG. We removed each mention of such an entity from the AIDA-CoNLL training set.

Also, we removed all already existing out-of-KG entities to only focus on the artificial ones. Note that those already existing were without identifiers and

> **Example 1**: **Hoda Muthana**, an **Alabama** woman who joined the **Islamic State**, is banned from entering the **United States**. .
>
> **Example 2**: **Peter Kaiser** wins the 2019, 1000-mile Iditarod, arriving in **Nome** in 9 days, 12 hours and 39 minutes..
>
> **Example 3**: In the aftermath of **Cyclone Idai**, those infected by cholera jump to 139 confirmed cases in **Mozambique**..

Figure 4.3: Wikievents example sentences. Entities marked in bold with Out-of-KG entities being underlined.

Table 4.2: Statistics of artificially created out-of-KG-entity enriched AIDA-CoNLL dataset (statistics of the original dataset in brackets)

	train	dev	test
# examples	946 (946)	216 (216)	231 (231)
# mentions	34,268 (46,678)	9,558 (11,824)	8,942 (11,206)
# out-of-KG mentions	0 (9,710)	952 (2,252)	900 (2,262)
# unique entities	3,935 (4,065)	1,638 (1,641)	1,530 (1,532)
# unique out-of-KG entities	0 (0)	166 (0)	155 (0)
Average of # mentions per example	36.2 (49.3)	44.2 (54.7)	38.7 (48.5)

Table 4.3: Percentage of entities occurring in clusters of different sizes

Dataset	Entity type	1	2	3	4	5	6-10	11-20	21-50	50-
Wikievents train	in-KG	62.6	14.4	6.5	3.4	2.3	5.0	2.7	1.9	1.2
	out-of-KG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wikievents dev	in-KG	65.6	13.9	5.9	3.5	2.0	4.6	2.5	1.3	0.7
	out-of-KG	75.4	12.1	4.9	1.5	0.9	2.5	0.7	1.1	0.9
Wikievents test	in-KG	65.8	13.9	5.8	3.1	2.2	4.7	2.2	1.6	0.6
	out-of-KG	75.7	12.5	2.9	2.5	1.4	1.9	1.1	1.1	1.0
AIDA-CoNLL train	in-KG	45.6	19.1	9.3	7.2	3.5	8.8	3.3	2.4	0.9
	out-of-KG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AIDA-CoNLL testa	in-KG	54.4	21.2	6.5	4.8	2.4	6.7	2.5	1.2	0.3
	out-of-KG	47.6	24.1	7.2	4.8	2.4	10.2	3.0	0.6	0.0
AIDA-CoNLL testb	in-KG	48.4	22.3	9.7	7.1	3.7	5.1	2.0	1.5	0.2
	out-of-KG	48.4	25.8	7.7	3.9	4.5	5.8	2.6	0.6	0.6

therefore not of use for our setting. The statistics of the AIDA datasets can be found in Table 4.2. In brackets, the original statistics can be found. Note that this version of AIDA-CoNLL is already mapped to Wikidata. Additional statistics on the mention clusters in the datasets can be found in Table 4.3.

4.3.3 Evaluation metrics

To evaluate the performance of the EL without out-of-KG detection, we report the in-KG linking accuracy. It is calculated by dividing the number of correctly linked mentions by the number of all mentions. EL with out-of-KG detection is evaluated by calculating the Precision, Recall and F-measure. The true positives are entity mentions detected correctly as being in the KG and correctly linked. False positives are entity mentions which were incorrectly linked to entities in the KG. This encompasses entity mentions referring to entities in the KG and entity

Table 4.4: Comparison of entity linking performance with different features on AIDA-CoNLL (repeated with three different seeds)

Model	Accuracy
Mention Encoder	0.852 ± 0.002
Pop.	0.615 ± 0.000
TransE.	0.643 ± 0.000
Mention Encoder + Pop.	0.850 ± 0.002
Mention Encoder + TransE	0.866 ± 0.003
Mention Encoder. + Pop. + TransE	**0.868 ± 0.004**

mentions not existing in the KG yet. Lastly, false negatives are entity mentions which do refer to being in the KG but are detected as being out-of-KG.

Lastly, to evaluate the clustering performance, we measured the CEAF, MUC and B^3 as commonly employed in coreference resolution (Moosavi and Strube, 2016). Furthermore, we report the in-KG F-measure, the out-of-KG F-measure (defined by Kassner et al. (Kassner et al., 2022)) and the combination of them $\frac{F_{\text{in-KG}}F_{\text{out-of-KG}}}{2}$. These measures are reported for out-of-KG entity mentions and entity mentions in the KG.

4.3.4 Results

Entity Linking Performance

In the first step, we evaluated how the different features affected the entity linking performance on the modified AIDA-CoNLL dataset. In this section, our primary goal is to select a suitable entity linking method that will be effective in the later stages of our research, particularly for handling out-of-KG entities. As such, we have opted not to compare our approach with other existing methods at this point, instead concentrating on identifying the specific features and determining the potential advantages of a sequential linking method. As can be seen in Table 4.4, the mention encoder itself already contributes the most to the overall performance. The popularity contributes only slightly. Nevertheless we see that popularity is still a feature disambiguating nearly 61.5% of all mentions.

out-of-KG detection

In the Tables 4.5, we show the precision, recall and F-measure of the models trained with different out-of-KG ratios (how likely the true candidate is removed from the candidate set). As can be seen, the recall is the largest if the model can ignore to detect out-of-KG entities. This is the case as no entity mentions are filtered out and all are linked to an entity in the KG. However, the precision is lower as all out-of-KG entities are misdetected as being in the KG and hence automatically also mislinked. The larger the ratio is, the higher the precision becomes. Of course, when only out-of-KG entities occur in the training data, the precision decreases again as the model does not learn to link at all. Interestingly, the maximum F-measure is reached at different points for the different datasets. For AIDA-CoNLL,

Table 4.5: Effect of different sample ratios for out-of-KG entities on entity linking performance

(a) AIDA-testb

out-of-KG ratio	P	R	F1
0.0	0.795	**0.867**	0.829
0.05	0.860	0.844	**0.852**
0.1	0.860	0.827	0.843
0.2	0.875	0.815	0.844
0.3	0.875	0.804	0.838
0.4	0.872	0.788	0.828
0.5	**0.880**	0.756	0.814
1.0	0.447	0.071	0.122

(b) Wikievents-test

out-of-KG ratio	P	R	F1
0.0	0.771	**0.834**	0.801
0.05	0.819	0.831	0.825
0.1	0.832	0.829	0.830
0.2	0.851	0.824	0.837
0.3	0.861	0.820	0.840
0.4	0.875	0.816	0.845
0.5	**0.887**	0.809	**0.846**
1.0	0.011	0.000	0.001

the largest F-measure is reached at a small out-of-KG ratio of 0.05 while for the Wikievents-test set it is reached at 0.50 (From 0.60 on it decreased for Wikievents). We suspect that this is the case as the candidate sets in both cases have a different candidate gold-candidate recall rate (how often the true entity is in the candidate set). For AIDA-testb, the gold-candidate recall is 0.97, while for Wikievents it is 0.89.[23] Hence in AIDA, a larger ratio leads to more entity mentions detected as out-of-KG while the true entity is in the candidate set. **Nevertheless, it is clear, that incorporating the detection of out-of-KG entities during training leads to a higher F-measure**.

Clustering out-of-KG entities

For all four methods, the hyperparameters were tuned on the validation set and then used on the test set in regard to the combined F-measure of out-of-KG entity linking and in-KG entity linking. The metrics are presented separately for in-KG entities and out-of-KG entities.[24]

As can be seen in Table 4.6a for AIDA-CoNLL, the clustering methods often outperform the sequential method on the clustering metrics for the out-of-KG entities. For the in-KG entities the sequential method comes close but does not outperform the best performing clustering methods. In regard to the F-measure, the in-KG entity linking performance of the sequential method outperforms all others. For the out-of-KG entity linking, the best performing model is the Edin clustering method. The sequential method has the second-best performance here. The SOTA NASTyLinker is outperformed by both. The additional KG information does help the entity linking process of the in-KG entities while it does not help the clustering of out-of-KG entities.

23. These rates are not in conflict with the previous statement that Wikievents mentions are less ambiguous. Different candidate generation methods were applied as mentioned in 4.2.2 resulting in different gold-candidate recall rates.

24. in-KG entities are considered here as well as they can also occur in the out-of-KG entity detection. We also conducted the experiments while assuming a perfect out-of-KG detection result and achieved similar results.

Table 4.6: Clustering performance (Thresholds determined on validation set). Best in bold, second best underlined.

(a) AIDA-CoNLL testb

	Sequential	Seq. w/o TransE	Bottom-up	Edin	NASTyLinker
$CEAF_{inkg}$	0.929 ± 0.002	0.929 ± 0.002	0.888 ± 0.010	**0.938 ± 0.005**	0.932 ± 0.001
MUC_{inkg}	0.989 ± 0.000	0.989 ± 0.000	0.985 ± 0.001	**0.993 ± 0.000**	0.991 ± 0.001
$B3_{inkg}$	0.938 ± 0.001	0.938 ± 0.001	0.916 ± 0.005	**0.952 ± 0.003**	0.944 ± 0.002
MUC_{ookg}	0.981 ± 0.001	0.981 ± 0.001	0.957 ± 0.002	**0.992 ± 0.001**	0.989 ± 0.001
$B3_{ookg}$	0.866 ± 0.004	0.864 ± 0.006	0.594 ± 0.025	**0.958 ± 0.003**	0.929 ± 0.017
$CEAF_{ookg}$	0.821 ± 0.009	0.819 ± 0.011	0.450 ± 0.022	**0.945 ± 0.004**	0.911 ± 0.021
$F1_{inkg}$	**0.843 ± 0.002**	**0.843 ± 0.002**	0.826 ± 0.001	0.836 ± 0.005	0.831 ± 0.009
$F1_{ookg}$	0.605 ± 0.014	0.603 ± 0.016	0.048 ± 0.004	**0.613 ± 0.021**	0.534 ± 0.047
$F1_{combined}$	0.724 ± 0.007	0.723 ± 0.009	0.437 ± 0.001	**0.725 ± 0.013**	0.682 ± 0.027

(b) Wikievents-test

	Sequential	Seq. w/o TransE	Bottom-up	Edin	NASTyLinker
$CEAF_{inkg}$	0.894 ± 0.002	0.894 ± 0.001	0.875 ± 0.008	0.906 ± 0.004	**0.927 ± 0.001**
MUC_{inkg}	0.951 ± 0.000	0.951 ± 0.000	0.939 ± 0.001	0.950 ± 0.000	**0.963 ± 0.000**
$B3_{inkg}$	0.911 ± 0.001	0.910 ± 0.001	0.882 ± 0.005	0.924 ± 0.002	**0.933 ± 0.001**
MUC_{ookg}	0.915 ± 0.001	0.912 ± 0.001	0.900 ± 0.002	0.862 ± 0.000	**0.931 ± 0.001**
$B3_{ookg}$	0.714 ± 0.003	0.726 ± 0.004	0.680 ± 0.019	0.693 ± 0.003	**0.758 ± 0.016**
$CEAF_{ookg}$	0.621 ± 0.008	0.628 ± 0.006	0.653 ± 0.024	0.652 ± 0.004	**0.738 ± 0.017**
$F1_{inkg}$	0.840 ± 0.002	0.840 ± 0.001	0.805 ± 0.001	0.844 ± 0.004	**0.847 ± 0.007**
$F1_{ookg}$	0.299 ± 0.014	0.303 ± 0.009	0.180 ± 0.002	0.303 ± 0.021	**0.403 ± 0.043**
$F1_{combined}$	0.570 ± 0.009	0.572 ± 0.005	0.493 ± 0.001	0.573 ± 0.014	**0.625 ± 0.015**

For the Wikievents dataset, the sequential method is outperformed by the clustering methods on both, the clustering metrics and the F-measures (see Table 4.6b). We suspect that this is the case due fewer entities co-occurring in the Wikievents dataset. This delivers less context in the form of already linked entities when using the TransE encodings. This leads to fewer benefits for the in-KG entity linking as well as more noise for clustering the out-of-KG entity mentions. The best performing model is the NASTyLinker. Additionally, we examined the wrong clusters when using the mention encoder and identified that most often COVID-related entities are clustered together. As the model was trained on data before COVID-19 had an impact it struggles to differentiate entities related to this as they are syntactically very close. The clustering performance for all methods is reduced in comparison to the AIDA-CoNLL dataset. **Evidently, the clustering of out-of-KG entities is more challenging on the Wikievents dataset**. This introduces a new research avenue.

In contrast to the positive impact of knowledge graph (KG) information on in-KG entity linking, incorporating it into the clustering of out-of-KG entities has a negligible effect. This is the case for both datasets.

4.4 Related Work

Entity linking methods can be categorized into two types. First, discriminative methods that are based on the bi-encoder / cross-encoder pairing (Ayoola et al., 2022; Logeswaran et al., 2019; Wu, Petroni, et al., 2020). Both encoders are commonly BERT-like models. The bi-encoder encodes the description of each entity and matches it to the text by using an approximate nearest neighbor search. This is important as the next step, the cross-encoding, is expensive. Here, those neighbors are reranked by applying a cross-encoder to the concatenation of both, the input text and the entity description. The highest-ranked entity is then the final linked one. Another type of entity linker is based on generative models (Cao et al., 2021; De Cao et al., 2022). Here, instead of using some external description of an entity, the whole model memorizes the knowledge graph (KG) during training. The linked entity is then directly generated by the model. Such methods skip the problem of mining negatives which are crucial for a good performance of bi-encoder-based methods. Our method follows the discriminative paradigm.

State-of-the-Art entity linking methods often do not consider out-of-KG entities. This is most apparent when examining the most common EL evaluation dataset. AIDA-CoNLL (Hoffart et al., 2014) does contain out-of-KG entities but even in the original paper, they were ignored during evaluation. As a consequence of that, most subsequent methods ignore them as well. However, there exist certain entity linking subtasks where out-of-KG entities are of importance.

NIL-Clustering, introduced at TAC-2011 (Ji and Grishman, 2011), focuses on linking a whole batch of documents at once. Notably, the entity mentions occurring also contain out-of-KG entities, here called NIL entities. The goal is to not only link the in-KG entities but also the out-of-KG entities. In essence, this leads to a clustering of all documents. Naturally, most methods employ clustering techniques (Blissett and Ji, 2019; Cassidy et al., 2011; Dutta and Weikum, 2015; Fahrni et al., 2013; Graus et al., 2012; Greenfield et al., 2016; Huynh et al., 2013; Monahan et al., 2011; Tamang et al., 2012). These methods focus on Wikipedia while we focus on Wikidata as a proper KG.

Hoffart et al. (Hoffart et al., 2014), introduced the task of emerging entity discovery in 2014. Here, the goal is to link entities, occurring in incoming texts, while also being able to discover emerging entities. These are entities that are out-of-KG and recently created. For example, news articles might contain emerging entities as certain events occurring are entities but might not yet be added to the KG. To solve this problem of discovering emerging entities, auxiliary information is considered. The auxiliary information used is often retrieved from external documents such as crawled webpages (Hoffart et al., 2014; Wu et al., 2016; Zhang et al., 2019). In our work, we avoid external documents and solely focus on detecting out-of-KG entities by checking the candidates. Hoffart et al. published the AIDA-EE dataset but it is not freely available.

Since 2022, several new works on the subject matter were published. The EDIN benchmark (Kassner et al., 2022) focuses on the adaptation of an entity linking model to support unknown entities. Here, the training is split into two parts, a regular entity linking training and an adaptation phase where unknown entities

are encountered. The EDIN benchmark focuses on Wikipedia and introduces an adaptation phase. In contrast to that, we do not expect any adaptation data to be available.

TempEL (Zaporojets et al., 2022) is a benchmark focusing on the linking of evolving and emerging entities. However, the assumption here is that long well-formulated descriptions of emerging entities do exist. It is a more specific case of the zero-shot setting. No entities are encountered, for which no description exists. This differs from this work as we do not assume that any additional information is available for the out-of-KG entities.

Agarwal et al. (Agarwal et al., 2022) consider the detection of out-of-KG entities but they again assume long well-formulated descriptions of the in-KG entities.

The NILK dataset (Iurshina et al., 2022) is a dataset similar to ours but it is not accompanied by an entity linking model. The dataset was excluded from our evaluation due to its mention-focused construction. Specifically, each instance in the dataset consists of a single mention and its corresponding context. The dataset's train/dev/test split was created by partitioning the set of identified mentions. As a result, mentions of same sentences may appear in multiple splits, which poses a challenge for our approach since it depends on contextual information from other mentions within the same sentence. Consequently, the NILK dataset is not suitable for evaluating our method.

The NASTyLinker by Heist et al. (Heist and Paulheim, 2023) introduced a new clustering method and incorporated the scores of a cross-encoder in the clustering process. It was evaluated on the task of NIL-Clustering and focused on the NILK dataset with Wikipedia descriptions.

The work by Pozzi et al. (Pozzi et al., 2022) focuses on Wikipedia and examines the detection and clustering of out-of-KG as well. They modify an existing dataset to include out-of-KG entities as well. We additionally offer a dataset with true out-of-KG entities and provide its KG. Also, we do not assume to have knowledge about out-of-KG entities during training. In our case, they are only encountered during the evaluation. Finally, we put a greater focus on the out-of-KG detection mechanism.

The method by Dong et al. (Dong et al., 2023) relies on the availability of out-of-KG entities during training time. This differs from our method as we do not assume that this is the case.

The clustering of out-of-KG entities is related to cross-document coreference resolution (Bagga and Baldwin, 1998; Cattan et al., 2021; Monahan et al., 2011; Singh et al., 2011). However, we limit the clustering only to out-of-KG detected entity mentions and include information available in KG to support the clustering.

There exist several other methods (Ristoski et al., 2021; Sevgili et al., 2019) which include KG embeddings into the EL process. However, we are the first to examine their impact on the clustering of out-of-KG entities.

4.5 Future Work

In the future, we want to improve upon the results by using a more sophisticated training regime. For example, hard-negative sampling or arborescence sampling (Agarwal et al., 2022) could be employed which could improve the performance. Also, to improve the performance of the out-of-KG detection, it would make sense to introduce more methods from novelty detection (Yang et al., 2024) or open-set recognition (Geng et al., 2021) which specifically focus on the detection of instances of classes not encountered during training. Furthermore, the candidate generation can be improved by not relying on TF-IDF but embedding all entity definitions in a latent space and retrieving candidates by a k-nearest neighbor search.

Most importantly, we will look into creating another suitable dataset containing out-of-KG entities. While the Wikievents dataset includes them in a natural way, it is limited to short texts with only a small amount of context information. Alternatively, models which can cope with small amounts of input data need to be developed. Furthermore, Wikievents and AIDA-CoNLL focus on the news domain. Cleaner and less ambiguous texts are more common here. This can be seen by the good performance of the lexical similarity measures when clustering. However, out-of-KG entities also exist in other contexts like historical documents which tend to be noisier and thus challenging (Menzel et al., 2021).

4.6 Conclusion

We developed the Wikievents entity linking dataset, which contains out-of-KG entities, and demonstrated that it presents a significant challenge for clustering such entities. Moreover, we designed and assessed a sequential method that initially links entities or identifies them as out-of-KG, and subsequently clusters all out-of-KG entities for disambiguation.

Our findings reveal that our sequential method's ability to consider the entire sentence allows it to perform on par with or even surpass methods that cluster all mentions jointly in some cases. We also demonstrated the feasibility of learning out-of-KG entity detection during training and highlighted the importance of incorporating them to an appropriate degree.

Thus, we were able to show, that our approach which relies exclusively on information within a knowledge graph, eliminating dependency on lengthy textual descriptions is an alternative for Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering.

4.7 Limitations

One limitation of this study is that only one type of graph embedding was considered, and different embeddings like DistMult, ComplEx, etc., might result in different performance outcomes. Additionally, due to hardware limitations, we couldn't perform pre-training of the Entity Linker (EL) on a large corpus like

Wikipedia, which is common in many entity linking methods today. Finally, it's important to note that we assumed entity mentions in the text are already detected, but detecting entirely new entity mentions is a challenge and crucial for real-world applications.

Supplemental Material Statement: Source code and datasets are available at https://github.com/semantic-systems/out-of-kg-el.

Acknowledgments

This project was supported by the Hub of Computing and Data Science (HCDS) of Hamburg University within the Cross-Disciplinary Lab program. Additionally, support was provided by the Ministry of Research and Education within the SifoLIFE project "RESCUE-MATE: Dynamische Lageerstellung und Unterstützung für Rettungskräfte in komplexen Krisensituationen mittels Datenfusion und intelligenten Drohnenschwärmen" (FKZ 13N16836).

5

Incorporating Type Information into Zero-Shot Relation Extraction

Bibliographic Information

Cedric Möller and Ricardo Usbeck. 2024. Incorporating Type Information into Zero-Shot Relation Extraction. In *Joint proceedings of the 3rd International workshop on knowledge graph generation from text (TEXT2KG) and Data Quality meets Machine Learning and Knowledge Graphs (DQMLKG) co-located with the Extended Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 26-30, 2024,* edited by Sanju Tiwari, Nandana Mihindukulasooriya, Francesco Osborne, Dimitris Kontokostas, Jennifer D'Souza, Mayank Kejriwal, Maria Angela Pellegrino, Anisa Rula, José Emilio Labra Gayo, Michael Cochez, and Mehwish Alam, 3747:10. CEUR Workshop Proceedings. CEUR-WS.org

Abstract

The task of zero-shot relation extraction focuses on the extraction of relations not seen during training time. Commonly, additional information about the relation such as the relation name or a description of the relation is utilised. In this work, we analyze whether a relation extractor can benefit from the inclusion of fine-grained type information about the involved entities. This is based on the intuition that relation descriptions might contain ontological information on the domain and range of the entity types that are usually put into relation. For that, we follow a cross-encoding setup where we encode both, the entity information and relation information, as one sequence and learn to score the representation. We examine this method on several datasets and show that the inclusion of the fine-grained type information leads to an improvement in performance.

5.1 Introduction

Identifying the relation that is expressed between entities is a very important subproblem of various downstream tasks. For instance, it is critical to handle semantic-web-related tasks such as knowledge graph question answering or knowledge graph population. Usually, it is assumed that the encountered relations are known before. Zero-shot relation extraction breaks with this assumption. During inference time, the goal is to extract entirely new relations not seen before during training time.

With the establishment of pre-trained models, this goal becomes achievable. Those models are trained on large corpora of textual data in an unsupervised way. In zero-shot relation extraction, one assumes that some information on the new relations is available. The simplest kind of information is a label describing the relation. But this only works if the relation label co-occurs with a similar context as encountered during the training of the pre-trained models. If this is not the case, using additional information such as a description of the relation is necessary.

In this work, we analyse the impact of combining fine-grained type information and the relation description on the relation extraction performance. This is based on the assumption that the descriptions contain valuable information on the types of the involved entities. For example, the description of the relation `director` states `director(s) of film, TV-series, stageplay, video game or similar`. Therefore it is clear, that the relation should not be used when talking about board members of a company, also sometimes referred to as directors. We incorporate fine-grained type information extracted from **Wikidata** together with the relation descriptions in the relation extraction process. [25]

The contributions are:

- Zero-shot relation extraction model using fine-grained type information and relation descriptions
- Ablation study on the impact of fine-grained type information and relation descriptions on the performance

5.2 Methodology

5.2.1 Problem Definition

The problem of relation extraction can be defined as follows: Given an input text c, an annotated head h and tail entity e, identify the correct relation r as expressed in the text. Zero-shot relation extraction separates the set of relations encountered during training from the ones encountered during inference. Hence, during training time, the set of available relations is R_{train}, while during test time, the set is R_{test}. It holds that $R_{\text{train}} \cap R_{\text{test}} = \emptyset$. Also, no annotated examples containing any relations in set R_{test} are available during inference time. Additional

25. Code/Data available at: https://github.com/semantic-systems/zero-shot-re

information defining the relation is available. We assume labels, descriptions and type information on entities to be available.

5.2.2 Method

To study the impact of fine-grained type information, we opt to extend a simple but powerful model introduced by Lan et al. (Lan et al., 2023). Hence, we cross-encode the information of the text and the relation information in a single input. Different from their work, we do not solely rely on the relation label but also include the relation description. Additionally, we assume the existence of fine-grained types for both, the head and the tail relation, extracted using the `P31` relation in Wikidata. We include the relation description under the assumption that it contains valuable ontological information referring to the fine-grained types of the considered entities. For example, for the relation `shipping port`, the description is

```
shipping port of the vessel (if different from
"ship registry"): For civilian ships, the
primary port from which the ship operates ...
```

We denote the types of the head entity by $\mathcal{T}_h$ and the types of the tail entity by $\mathcal{T}_t$. Additionally, for each type of an entity, we extract the label describing the type (e.g., `human` for `Q5`). The input x to the model then consists of four different segments. The **first segment** describes the head entity:

```
Head Entity : {l_h} with Types : {T_h}
```

and the **second segment** describes the tail entity:

```
Tail Entity : {l_t} with Types : {T_t}
```

where $l_\square$ denotes the label of the head entity h or tail entity t. $T_\square$ is the concatenation of the labels of the types of the head or tail entity $T_\square = \bigoplus_{u \in \mathcal{T}_\square} l_u$.

The **third segment** gives information on the input text:

```
Context : {c}
```

The **final segment** gives information on the relation:

```
{l_r} defined as {d_r}
```

where l_r denotes the label of the relation r and d_r is the description of the relation r.

All segments are then combined into a single coherent text as follows:

```
[CLS] Given the Head Entity : {l_h} with
Types : {T_h}, Tail Entity : {l_t} with
Types : {T_t} and Context : {c}, the context
expresses the relation [SEP] {l_r} defined
as {d_r} [SEP]
```

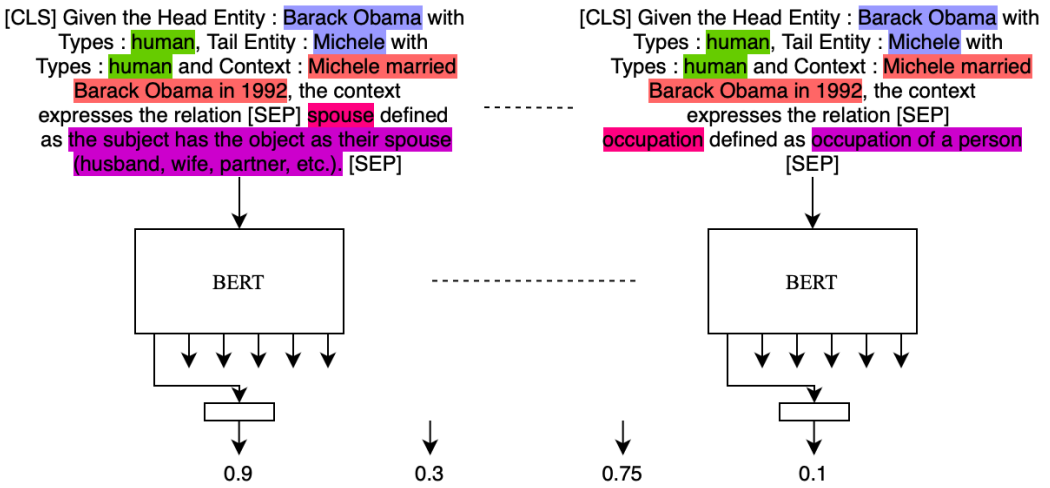


Figure 5.1: Model overview: Green specifies the types, blue the entities, orange the context, red the relation label and Purple the description of the relation.

The whole text is then fed into an encoder-only model $f(x)$ which returns a sequence of vectors $e_{[CLS]} \dots e_{[SEP]}$. The vector $e_{[CLS]}$ is then taken and fed a input to a linear layer which then returns a final score.

$$s_r = l(e_{[CLS]})$$

This is done for each potential relation, which gives us $|R_{\text{test}}|$ scores. The highest score is taken as the predicted relation. All potential relations are known beforehand. During training, the model is optimized using cross-entropy loss. Each example contains a single positive relation. The model trains to differentiate it against other relations by comparing it against incorrect relations. For that, n other relations are sampled and used as negative examples. [26]

An overview of the model can be found in Figure 5.1.

5.3 Evaluation

We evaluate the model on two popular datasets, FewRel and Wiki-ZSL. Both datasets were annotated on Wikipedia article texts. FewRel is originally a few-shot relation extraction dataset annotated by Han et al. (Han et al., 2018). The dataset was modified for zero-shot purposes by Chia et al. (Chia et al., 2022). They split the training, validation and test examples by their relations into disjoint sets. Wiki-ZSL is a zero-shot relation extraction dataset created by Chen et al. (Chen and Li, 2021) based on the Wiki-KB (Sorokin and Gurevych, 2017). As the entities and relations in both datasets are linked to Wikidata, we focus on it as the knowledge graph providing the fine-grained entity types.

In each dataset, the set of relations in the training and test dataset is disjoint and randomly assigned. Three different settings are examined per dataset. Each setting considers a different number of relations in the train, validation and test set. The

26. In our experiments we set $n = 5$.

number of relations in the validation/test set varies between $m = 5$, $m = 10$ and $m = 15$ relations. These relations are randomly picked and the remaining relations are assigned to the training set.

To handle the considerable noise induced by the random selection of the relations, the dataset for $m = 5$, $m = 10$ and $m = 15$ were randomly split into train, validation and test sets for five times. A method is evaluated on each split and the results are averaged.

As metrics, precision, recall and F1 are calculated. All metrics are computed in a macro setting which means that for each relation the precision, recall and F1 are calculated and then averaged over all relations.

We compare our method, called TMC-BERT, against several methods: CIM (Rocktäschel et al., 2016) solves the task as a textual entailment problem where the relation descriptions and the input sentence are given to a Natural Language Inference model to classify whether the input sentence entails the relation description. This is done for all potential relations and the highest scoring is taken. ZS-BERT (Chen and Li, 2021) encodes the input sentence as well as the relation descriptions into a dense vector space. An nearest neighbor search is conducted over all the encodings of the relation descriptions given the input sentence. The closest relation encoding is the final relation. Tran et al. (2022) (Tran et al., 2022) again encode the input sentence and relation descriptions into a dense vector space. They additionally employ a contrastive-learning inspired loss on the input sentence and relation encodings. The final scoring is achieved by concatenating the relation encoding and the sentence encoding and feeding it into a linear layer. RE-Matching (Zhao, Zhan, et al., 2023) encodes the input sentence and relation descriptions as well but uses feature distillation to calculate a similarity score based on more fine-grained feature interactions. RelationPrompt (Chia et al., 2022) relies on a generative model to generate synthetic data as additional training samples. At the same time, the generative model is also used to generate a relation given the sentence and the two entities as input. We compare against the model with (RelationPrompt) and without (RelationPrompt NG) synthetic training data. MC-BERT (Lan et al., 2023) models the relation extraction similar to us as a multiple-choice problem where the input sentence and the relation label are rearranged together in a natural sentence, encoded and scored. DSP-ZRSC (Lv et al., 2023) solves the problem via Discriminative Soft Prompting where the input text, the entities and all relation labels are concatenated, fed into a prompt discriminative language model and each relation label is scored. Tran et al. (2023) (Tran et al., 2023) solve it as a representation learning problem and introduce a second loss term incorporating the degree of correlation between sentences and relations.

BERT-base-case was used as the model to stay comparable to MC-BERT. The model was fine-tuned on two NVIDIA A6000s with a batch size of 48 and a learning rate of $5e-5$.

5.3.1 Results

As can be seen in Table 5.1, the incorporation of type-related information leads to a large increase in performance on several datasets in comparison to regular

Table 5.1: Results on FewRel and Wiki-ZSL

		Wiki-ZSL			FewRel		
m	Model	P	R	F1	P	R	F1
5	CIM	49.63	48.81	49.22	58.05	61.92	59.92
	ZS-BERT	71.54	72.39	71.96	76.96	78.86	77.90
	Tran et al. (2022)	87.48	77.50	82.19	87.11	86.29	86.69
	RelationPrompt NG	51.78	46.76	48.93	72.36	58.61	64.57
	RelationPrompt	70.66	83.75	76.63	90.15	88.50	89.30
	RE-Matching	78.19	78.41	78.30	92.82	92.34	92.58
	DSP-ZRSC	94.1	77.1	84.8	93.4	92.5	92.9
	Tran et al. (2023)	**94.50**	**96.48**	**95.46**	**96.36**	**96.68**	**96.51**
	MC-BERT	80.28	84.03	82.11	90.82	90.13	90.47
	TMC-BERT	90.11	87.89	88.92	93.94	93.30	93.62
10	CIM	46.54	47.90	45.57	47.39	49.11	48.23
	ZS-BERT	60.51	60.98	60.74	56.92	57.59	57.25
	Tran et al. (2022)	71.59	64.69	67.94	64.41	62.61	63.50
	RelationPrompt NG	54.87	36.52	43.80	66.47	48.28	55.61
	RelationPrompt	68.51	74.76	71.50	80.33	79.62	79.96
	RE-Matching	74.39	73.54	73.96	83.21	82.64	82.93
	DSP-ZRSC	80.0	74.0	76.9	80.7	**88.0**	84.2
	Tran et al. (2023)	**85.43**	**88.14**	**86.74**	81.13	82.24	81.68
	MC-BERT	72.81	73.96	73.38	**86.57**	85.27	**85.92**
	TMC-BERT	81.21	81.27	81.23	84.42	84.99	85.68
15	CIM	29.17	30.58	29.86	31.83	33.06	32.43
	ZS-BERT	34.12	34.38	34.25	35.54	38.19	36.82
	Tran et al. (2022)	38.37	36.05	37.17	43.96	39.11	41.36
	RelationPrompt NG	54.45	29.43	37.45	66.49	40.05	49.38
	RelationPrompt	63.69	67.93	65.74	74.33	72.51	73.40
	RE-Matching	67.31	67.33	67.32	73.80	73.52	73.66
	DSP-ZRSC	77.5	64.4	70.4	**82.9**	78.1	80.4
	Tran et al. (2023)	64.68	65.01	65.30	66.44	69.29	67.82
	MC-BERT	65.71	67.11	66.40	80.71	79.84	80.27
	TMC-BERT	**73.62**	**74.07**	**73.77**	82.11	**79.93**	**81.00**

MC-BERT. On Wiki-ZSL, the performance increases vary between 6 and nearly 8 F1 points. The type-related information has a great impact on, both, recall and precision. On FewRel, the performance increases when considering 5 or 15 unseen relations. However, the performance increases are less pronounced. In comparison to the current SOTA method by Tran et al. (Tran et al., 2023), TMC-BERT considerably surpasses its performance when confronted with 15 unseen relations. This is the most complex setting as much fewer examples and relations are available during training while more potential relations are encountered during inference. Here, the additional type information helps a lot. Furthermore, the inclusion of fine-grained type information is orthogonal to the properties of the

method by Tran et al. (Tran et al., 2023). Their method could benefit from it as well.

5.3.2 Ablation study

To examine what changes lead to the large increase in performance, we conducted an ablation study on the incorporation of different kinds of information. Here, we differentiated between three cases:

1. TMC-BERT
2. TMC having the types of the subject and object entity removed (TMC-BERT w/o types)
3. TMC having the description of the relation removed (TMC-BERT w/o desc.)

Table 5.2: Ablation study of on FewRel and Wiki-ZSL

		Wiki-ZSL			FewRel		
m	Model	P	R	F1	P	R	F1
5	TMC-BERT w/o desc.	85.56	84.07	84.74	**93.96**	93.26	93.61
	TMC-BERT w/o types	85.00	84.41	84.68	93.33	92.50	92.91
	TMC-BERT	**90.11**	**87.89**	**88.92**	93.94	**93.30**	**93.62**
10	TMC-BERT w/o desc.	77.26	78.16	77.70	**85.24**	83.29	84.25
	TMC-BERT w/o types	74.89	76.05	75.46	85.16	83.36	84.24
	TMC-BERT	**81.21**	**81.27**	**81.23**	84.42	**84.99**	**85.68**
15	TMC-BERT w/o desc.	72.33	71.16	71.73	79.22	76.46	79.79
	TMC-BERT w/o types	68.53	69.81	69.16	79.22	78.19	78.69
	TMC-BERT	**73.62**	**74.07**	**73.77**	**82.11**	**79.93**	**81.00**

As can be seen in Table 5.2, the addition of the relation description alone was the least beneficial type of information. Adding information on relation types leads to a larger improvement, probably as the pre-trained model already associates specific types with certain relation labels. Finally, the ablation study shows that the relation description and fine-grained entity type information complement each other, as using each separately does not lead to as large a decrease in performance as using them together.

5.3.3 Entity Linking impact

As it is not realistic that fine-grained type information is always available, we also evaluate the model when identifying entity types using an entity linker (EL). For that, we train the model with known entity types but evaluate with the entity types as retrieved by an entity linker. As an entity linker, we use ReFinED (Ayoola et al., 2022). As can be seen in Table 5.3, the performance diminishes when using types identified through entity linking. On Wiki-ZSL, the performance is still surpassing the existing SOTA results at all times. On FewRel, the performance is still greater when only confronted with five relations but decreases more when

having to predict 10 or 15 relations. One reason might be that the entity linking performance is lower on FewRel than on Wiki-ZSL.

Table 5.3: Results on FewRel and Wiki-ZSL when using an entity linker

		Wiki-ZSL			FewRel		
m	Model	P	R	F1	P	R	F1
5	TMC-BERT	**90.11**	**87.89**	**88.92**	**93.94**	93.30	93.62
	TMC-BERT + EL	88.44	87.07	87.73	**93.94**	**93.35**	**93.64**
10	TMC-BERT	**81.21**	**81.27**	**81.23**	**84.42**	**84.99**	**85.68**
	TMC-BERT + EL	81.16	81.22	81.18	84.88	83.43	84.14
15	TMC-BERT	**73.62**	**74.07**	**73.77**	**82.11**	**79.93**	**81.00**
	TMC-BERT + EL	73.53	73.96	73.67	80.87	78.74	79.78

5.3.4 Case study

Table 5.4 illustrates two instances where the inclusion of type information or relation descriptions proved beneficial. In the first case, specifying that MMORPG belongs to the video game genre facilitated the correct classification of the **genre** relation. In the second example, highlighting that bass is a voice type aligned the type label precisely with the **voice type** relation label. Additionally, the relation description directly addressed the voice type of bass.

Table 5.4: Comparison of the performance of TMC-BERT and MC-BERT on two different examples. Ground-truth relations are shown in bold. The interacting entities and their types are shown in dictionaries following the sentences.

Method	TMC-BERT	MC-BERT
Sentence	Gravity Corporation is a South Korean video game corporation primarily known for the development of the MMORPG Ragnarok Online. {MMORPG: video game genre; Ragnarok Online: video game}	
Classified Relation	**genre**	manufacturer
Description of Relation	creative work's genre or an artist's field of work	main use of the subject (includes current and former usage)
Sentence with entity types	Putnam Griswold (1875-1914) was an American opera singer (bass), born in Minneapolis, Minnesota. {Putnam Griswold: human, bass: voice type}	
Classified Relation	**voice type**	use
Description of Relation	person's voice type. expected values: soprano, mezzo-soprano, contralto, countertenor, tenor, baritone, bass (and derivatives)	main use of the subject (includes current and former usage)

5.4 Related Work

Commonly, relation extraction is tackled as classification problem. Usually, the input text is encoded and a classification head is attached. To encode text, CNNs (Zeng et al., 2014), RNNs (Miwa and Bansal, 2016) or transformers (Zhong and Chen, 2021) are usually employed. Recently, pre-trained models have been fine-tuned on the relation extraction task. Due to the fixed classification head, such trained models are not flexible enough to handle new relations (Wu and He, 2019). Hence, when targeting zero-shot relation extraction other methods are necessary. Representation-learning-based methods (Chen and Li, 2021; Tran et al., 2023; Tran et al., 2022; Zhao, Zhan, et al., 2023) try to embed the textual information and the relational information in the same vector space. For that, relational information such as labels or descriptions are usually used to get a representation of the relation. The goal is to learn representations such that the representation of the true relation resides close to a representation of the text in the vector space while the representation of false relations is far away. Recently, generative language models have been increasingly utilized for the task (Chen et al., 2022; Chia et al., 2022; Lv et al., 2023; Ni et al., 2022). Here, the model is prompted with the input text as well as information on the potential relations. The model is then fine-tuned to either generate the relation as expressed in the input text or a full triple consisting of the two entities and relation. For example, Chen et al. (Chen et al., 2022) model

it as solving a Masked Language Modelling problem. Also, generative models were applied to generate synthetic training data for relation extraction (Chia et al., 2022).

Type information was considered in previous works focusing on relation extraction but these works either used very broad types or did not tackle zero-shot relation extraction (Koch et al., 2014; Liu et al., 2014; Wu and Chen, 2020).

Some methods model the problem as a textual entailment problem (Obamuyide and Vlachos, 2018; Rahimi and Surdeanu, 2023; Sainz et al., 2021). Here, the idea is that a model that is pre-trained on the textual entailment task is directly applied to the relation extraction task. The assumption is that the model can identify whether the textual information entails the relation description.

The method by Lan et al (Lan et al., 2023) models relation classification as a multiple-choice problem where the text is encoded with relation information and a score is calculated. This is done for all relations and the relation with the highest score is taken. We extend this approach.

5.5 Conclusion and Future Work

In this work, we examined the impact of fine-grained type information on the zero-shot relation extraction problem. Different from past methods, we employed fine-grained type information as additional information and showed that combining this information with the description of the relation leads to a synergistic effect, improving the performance overall. We believe that this is the case because the description information provides valuable ontological information on the domain and range of a relation. This domain and range are then compared against the fine-grained type information of the interacting entities. Furthermore, we validated whether the increase in performance did indeed spring from the combination of type and relation description information which is indeed the case. Finally, we studied the impact of using an entity linker to retrieve the entity types. While it leads to a decrease, the performance often still surpasses the current SOTA in the most complex setting considerably.

In future works, we want to tackle multiple problems. First, it is not certain that one has access to fine-grained type information during inference. Therefore, we want to examine, whether the performance of a trained entity typer is sufficient to produce similar results. Secondly, the current architecture follows a cross-encoding approach. While this is not a problem when one encounters only a few relations during inference, in real-world use cases this is not typically the case. There are hundreds of potential different relations that could be encountered during inference. Cross-encoding the text with each one leads to a substantial computational effort. We want to examine whether the relation candidate generation module might also benefit from fine-grained type information. Also, the training process currently only trains the model by randomly sampling other relations. Choosing the relationships in a smarter way might lead to additional improvement. Finally, the impact of fine-grained entity types from other knowledge graphs needs to be evaluated.

6

DISCIE - Discriminative Closed Information Extraction

Bibliographic Information

Cedric Möller and Ricardo Usbeck. 2024. DISCIE-Discriminative Closed Information Extraction. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II,* edited by Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan, 15232:23–40. Lecture Notes in Computer Science. Springer

Abstract

This paper introduces a novel method for closed information extraction. The method employs a discriminative approach that incorporates type and entity-specific information to improve relation extraction accuracy, particularly benefiting long-tail relations. Notably, this method demonstrates superior performance compared to state-of-the-art end-to-end generative models. This is especially evident for the problem of large-scale closed information extraction where we are confronted with millions of entities and hundreds of relations. Furthermore, we emphasize the efficiency aspect by leveraging smaller models. In particular, the integration of type-information proves instrumental in achieving performance levels on par with or surpassing those of a larger generative model. This advancement holds promise for more accurate and efficient information extraction techniques.

6.1 Introduction

Today, our ability to generate data far surpasses our ability to understand it, particularly when that data is in textual form. As a potential solution, knowledge graphs (KGs), structured representations of data as interconnected nodes and links, offer the promise of making complex information machine-readable and easily interpretable (Ji et al., 2022).

However, the process of automatically transforming unstructured text into a meaningful KG is a significant unsolved problem. It encompasses numerous complex subproblems such as entity recognition, relation extraction and semantic understanding, where each represents a substantial field of study.

In general, this means that a text is translated to a set of triples. Each triple consists of a subject, a predicate and an object. Each subject and object is an entity while the predicate corresponds to a relation between the two entities.

In this work, we focus on Closed Information Extraction (CIE) (Josifoski et al., 2022). Here, triples are extracted which are grounded in an underlying KG. This means that each of the extracted entities and relations have unique identifiers assigned. An example[27] is:

"Barack Obama was born in Hawaii" → `[Q76, P19, Q782]`

Recent methods like the State-of-the-Art model GenIE interpreted the task as an end-to-end machine translation task where the input is the text and the output are the triples. Generative models are employed to translate text directly to triples. While generative models proved to be very powerful, it is harder to incorporate external information (e.g., the underlying KG) into the generation process (Josifoski et al., 2022). Hence, the generative model is forced to learn the entire KG during training. As the size of such a KG can be huge, this can inhibit performance. Also, this means such generative methods are not able to use an evolving version of the KG. Furthermore, the sequential nature of the decoding process often to leads lower efficiency, which is critical given the large amount of textual data available today. Lastly, the authors reported a lower performance on long-tail relations.[28]

Instead of using generative models, this work employs discriminative models. This means, we first identify salient segments of the input text. Subsequently, external information is introduced to distinguish these segments within a predefined set of classes. The discriminative process encompasses tasks such as recognizing mentions, disambiguating entities, and extracting relations. Also, in many subtasks relevant to the task of end-to-end entity linking are non-generative models still state-of-the-art (Ma et al., 2023; Shavarani and Sarkar, 2023). This allows us to tackle three shortcomings of the generative State-of-the-Art model: efficiency, inclusion of external information and performance on long-tail relations.

We employ lightweight models in each step of our method which gives us a large efficiency boost. While such methods often perform worse than their larger

27. Using QIDs and PIDs from www.wikidata.org. QIDs are the identifiers of entities and PIDs are the identifiers of relations.

28. Relations rarely occurring.

counterparts, we investigate whether the utilization of fine-grained entity type information as external information into relation extraction step can alleviate the performance gap. Lastly, we explore whether this information has a positive impact on the performance on long-tail relations as well. The primary emphasis of this paper is centered around enhancements made to the relation extraction component. In terms of mention recognition and entity linking, our approach trains and uses models that have demonstrated high performance.

The contributions of this paper are as follows:

- Show that the inclusion of coarse-grained type information is not sufficient;
- Show that the inclusion of fine-grained type information has a large impact on relation extraction and hence CIE in general (in particular long-tail relations);
- Show that efficient lightweight discriminative models can outperform large-scale generative models when using fine-grained type information.

In the following, we will develop such a discriminative method and especially focus on the incorporation of type information into the relation extraction step.[29]

6.2 Method

6.2.1 Problem Definition - Closed Information Extraction

We define the problem as follows: Given are a text t and a KG $\mathcal{G} = (V, R, E)$ where V are all entities in the graph, R all relations in the graph and $E \subseteq (V \times R \times V)$ all edges each of which connects two entities via a relation. Each text t contains triples of form $< v, r, w >$ with $v, w \in V$ and $r \in R$. The goal is to extract these triples from text.

6.2.2 Model

Mention Recognizer

The mention recognizer is an encoder-only model that accepts the tokenized input text $t_1, \ldots, t_i, \ldots, t_n$. It encodes the whole sequence to get an embedded representation for each token $k_1, \ldots, k_i, \ldots, k_n$, where $k_i \in \mathbb{R}^d$. Then, each pair of subsequent tokens is combined by concatenation and fed into a linear layer $s_{i,j} = l(k_i \oplus k_j) \in \mathbb{R}$, classifying whether the pair denotes the first and last token of a mention or not. Overall it outputs $\frac{n(n+1)}{2}$ scores for a sequence of length n. All scores surpassing an initial threshold are taken as mention candidates and forwarded to the entity candidate generation module.[30] The model is trained with the binary cross entropy loss function.

29. The code can be found in: https://github.com/semantic-systems/discie

30. Usually, mention recognition is solved by applying BIO sequence tagging. We trained and evaluated such a method but achieved a lower performance in comparison to the token-pair-based approach described above.

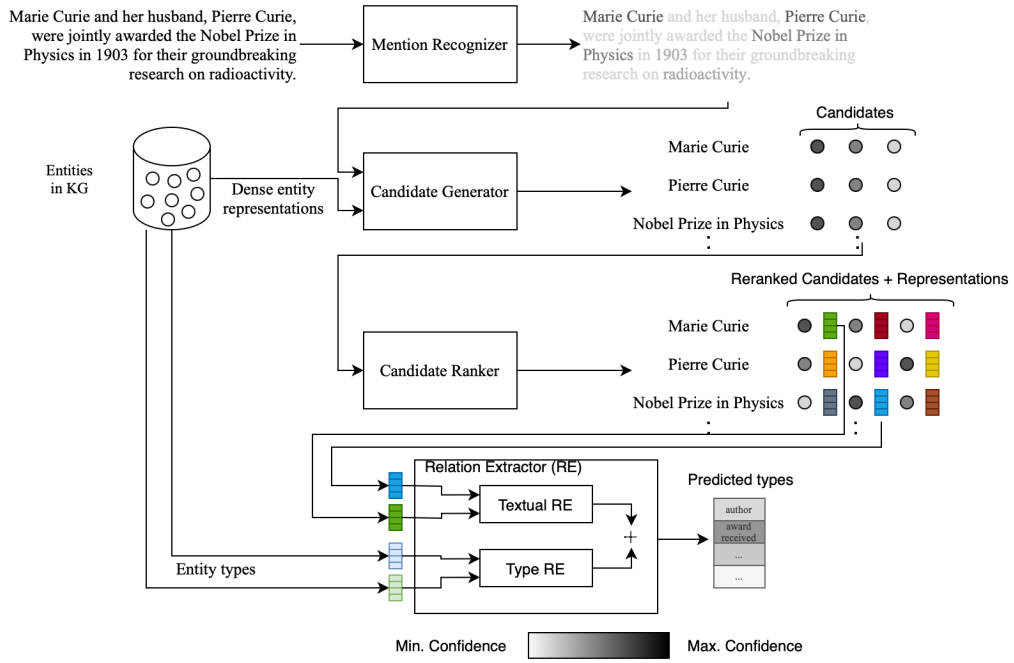


Figure 6.1: DISCIE - Architecture. The intensity of the colors indicate the scores. Higher intensity resolves to a higher score. The likely outcome would be the triples: [(Q7186:Marie Curie, P166:award received, Q38104:Nobel Prize in Physics), (Q7186:Marie Curie, P26:spouse, Q37463:Pierre Curie)]

Entity Candidate Generator

The entity candidate generator is based on the bi-encoder architecture, more specifically a Siamese network (Chicco, 2021). The Siamese network is an encoder-only model. It encodes the textual mention representation and the textual entity representation into dense representations. The textual mention representation is of form

[CLS] `{mention}` `[CTX_L]` `{context_left}` `[CTX_R]` `{context_right}` [SEP]

where `context_left` and `context_right` is the text of a certain window size left and right of the identified mention. [CTX_L] and [CTX_R] are special tokens denoting the context. The textual entity representation is of form

[CLS] `{label}` `[DESC]` `{desc}` [SEP]

where `{label}` is the English label of the entity and `{desc}` is the short description text as available in Wikidata via the predicate `schema:description`.[31] Both, the mention and entity textual representations, are fed into the same encoder-only model and encoded to retrieve the final mention/entity representation, which is here taken as the embedded `[CLS]` token. We denote the embedded mention

31. This could be replaced with any other KG containing descriptions.

representations as b_m and the embedded entity representation as b_c. Finally, both representations, are compared via cosine similarity

$$\frac{< b_m, b_c >}{\|b_m\|\|b_c\|} \in \mathbb{R}$$

where $< \cdot, \cdot >$ denotes the dot product and $\|\cdot\|$ the euclidean norm.

The model is trained with the binary cross-entropy loss using in-batch negatives and mined hard-negatives (Wu, Petroni, et al., 2020). When doing in-batch negative-based learning, all other entities in the current batch are interpreted as negatives. For mined hard-negatives, all entities are embedded after β epochs and for each training example, all γ nearest entities are found. All entities not being the ground-truth entity are now taken as negatives. The method returns for each mention m a set of candidates C_m. During training, $\beta = 1$ and $\gamma = 10$. After training, all entities are embedded and inserted into a vector index for fast retrieval.[32]

Entity Candidate Ranker

While the entity candidate generator alone could be used for entity disambiguation, it is usually less accurate. That is why in a subsequent step an entity candidate ranker is used that is less efficient but more accurate. It is applied to the subset of entities retrieved by the previous step. The candidate ranker re-ranks all the candidates C_m retrieved for a mention. It is based on the cross-encoder architecture (Wu, Petroni, et al., 2020). It takes the concatenated textual representations of the mention and entity candidate and feeds it into an encoder-only model. The cross-encoder architecture allows cross-attention between the entity representation and the input text. This usually leads to higher performance than just comparing the bi-encoder representations directly (Wu, Petroni, et al., 2020).[33] Hence, the input is of form:

[CLS] `{label} [DESC] {desc} [SEP] {mention} [CTX_L] {context_left} [CTX_R] {context_right}` [SEP]

[DESC] is a special token denoting the entity description. The embedded `[CLS]` token is taken (denoted as $b_{m,c}$) and fed into a final linear layer to get a similarity score

$$s_{m,c} = h(b_{m,c}) \in \mathbb{R}$$

.

During training, for each entity mention a set of hard negative entity candidates is sampled by using the entity candidate generator and the vector index. The model is trained via binary cross entropy loss including all the hard negatives and the positive entity candidate.

32. https://faiss.ai
33. This was also observable in our use case.

Relation Extractor

Textual information. The relation extractor accepts a pair of entity mentions. Instead of only focusing on the input text, we incorporate candidate information as well. We take each mention m and its highest scoring candidate c, and combine both $f(m, c)$. Then, each $f(m, c)$ is compared to all other $f(m', c')$ where $m \neq m'$. As the combination of each mention and its candidate $f(m, c)$ we simply use the embedded `[CLS]` token output by the candidate ranker, so $f(m, c) = b_{m,c}$. The predicted relation is scored by first calculating whether a subject-object relationship holds between a pair

$$< l_s(b_{m,c}), l_o(b_{m',c'}) > \in \mathbb{R}$$

where l_s and l_o are learnable linear layers. Then, a score for each potential relation is calculated as

$$W_r \left[b_{m,c} + b_{m',c'} \right]$$

where $W_r \in \mathbb{R}^{|R| \times d}$ is a learnable matrix and d is the dimension of $b_{m,c}$ and $b_{m',c'}$.

Both scores are then combined to get the final relation score

$$g[b_{m,c}, b_{m',c'}] = W_r \left[b_{m,c} + b_{m',c'} \right] + < l_s(b_{m,c}), l_o(b_{m',c'}) > \mathbf{1}$$

where $\mathbf{1} \in \mathbb{R}^{|R|}$ is a vector of ones. It holds that $g[b_{m,c}, b_{m',c'}] \in \mathbb{R}^{|R|}$.

Type information. Additionally, we also incorporate fine-grained type information into the relation extraction process. This is based on the intuition that certain relations are usually restricted to combinations of certain entity types. To learn these dependencies, we calculate relation classification logits separately from the textual representations just using the type information of each candidate. Each entity candidate c has a set of types $T_c \subseteq T$ where T is the set of types available in Wikidata. Now, we assign each type $t \in T$ a learnable vector $e_t \in \mathbb{R}^{d_T}$. As an entity might have multiple types, we create a condensed representation of the candidate as $t_c = \frac{1}{|T_c|} \sum_{t \in T_c} e_t$.

Then, we calculate the type-based relation logits by feeding the concatenation of t_c and another candidate $t_{c'}$ into a linear layer:

$$h(t_c \oplus t_{c'}) \in \mathbb{R}^{|R|}$$

Finally, we sum up the contextual logits and the type logits to get the final logits:

$$k(m, c, m', c') = h(t_c \oplus t_{c'}) + g[f(m, c) \oplus f(m', c')]$$

The relation extractor is trained via binary cross-entropy loss.

Inference

First, we retrieve a set of suitable mentions by applying the mention recognizer. After mapping its output to $(0, 1)$ by applying the sigmoid function, we retrieve a score $s^m_{i,j}$ for each possible span. Now, all spans surpassing a threshold ϵ_m are taken

as mention candidates. For each such mention, the entity candidate generator is applied to retrieve a set of candidates. Each candidate is reranked by applying the entity candidate reranker. Its scores $s^c_{m,c}$ are again mapped to $(0, 1)$. The final candidate score is then the average $s = \frac{s^m_{i,j} + s^c_{m,c}}{2}$. If the maximum score of all candidates surpasses a threshold ϵ_c, the candidate and its mention are accepted. Finally, the relations of each pair of mention-candidate combinations are calculated by using the relation extractor to get the relation scores s_r. Each relation score surpassing the final relation threshold ϵ_r is accepted.

Table 6.1: Results on REBEL and FewRel (Micro)

	REBEL				FewRel
Model	P	R	$F1$	$F2$	R
SOTA-Pipeline	43.30±0.15	41.73±0.13	42.50±0.13	-	17.89±0.24
GenIE	68.02±0.15	69.87±0.14	68.93±0.12	-	30.77±0.27
GenIE - PLM	59.32±0.13	77.78±0.12	67.31±0.10	-	46.95±0.27
DISCIE (F2 calibrated)	62.13±0.10	**81.93±0.07**	70.67±0.08	**77.02±0.06**	**47.10±0.28**
DISCIE (F1 calibrated)	**77.41±0.11**	72.68±0.08	**74.97±0.08**	73.58±0.07	34.39±0.29

Table 6.2: Results on GeoNRE and WikipediaNRE (Micro)

	GeoNRE			WikipediaNRE		
Model	P	R	$F1$	P	R	$F1$
SOTA-Pipeline	66.65±1.47	66.22±1.46	66.43±1.45	65.17±0.27	54.40±0.20	59.30±0.21
SetGenNet	86.89±0.51	85.31±0.47	86.10±0.34	82.75±0.11	77.55±0.27	80.07±0.27
GenIE	91.77±0.98	**93.20±0.83**	**92.48±0.88**	91.39±0.15	**91.58±0.15**	91.48±0.12
DISCIE	**92.4±0.90**	87.2±1.02	89.71±0.86	**91.57±0.16**	91.53±0.13	**91.55±0.12**

6.3 Evaluation

For evaluation, we used four different datasets: REBEL, WikipediaNRE, GeoNRE and FewRel. Here, REBEL (Huguet Cabot and Navigli, 2021) is a large-scale dataset while WikipediaNRE, GeoNRE (Trisedya et al., 2019) and FewRel (Han et al., 2018) are of smaller size. In regard to relations, REBEl contains 857 different relations while the other three datasets all contain fewer than 157 relations. During the evaluation, FewRel is used as a recall-only benchmark dataset as it is not exhaustively annotated (Josifoski et al., 2022). See Table 6.3 for information on the datasets. For the candidate representations, we use the concatenation of the Wikipedia title and the Wikidata description of the entity. The used Wikdata dump is from 2022. As for type information, we use the fine-grained types as given by the `P31` relation in Wikidata. Also, we extract for each type all supertypes and consider them as valid types of an entity. Finally, we restrict the set of types to the set as defined by Ayoola et al. (Ayoola et al., 2022) due to them showing great performance utilising them in the task of entity linking.[34]

34. 930 types are used in total. They were filtered by exploring how useful they are for disambiguating between different entities.

Table 6.3: Statistics of the datasets with T, E and R standing for the number of triples, entities and relations, respectively

Dataset	Examples			T			E	R
	Train	Dev	Test	Train	Dev	Test		
Rebel	1,899,331	104,960	105,516	5,147,836	284,268	284,936	1,498,143	857
WikipediaNRE	223,536	980	29,619	298,489	1,317	39,678	278,204	157
GeoNRE	-	-	1,000	-	-	1,000	124	11
FewRel	-	-	27,650	-	-	27,650	64,762	80

Similar to Josifoski et al. (Josifoski et al., 2022) we follow two training regimes: For REBEL and FewRel, we train on the training dataset of REBEL and evaluate on the REBEL test and FewRel test set. For WikipediaNRE and GeoNRE, we finetune the already REBEL-trained model on the training dataset of WikipediaNRE and then evaluate on the test sets of WikipediaNRE and GeoNRE.[35]

The thresholds ϵ_m, ϵ_c and ϵ_r necessary for inference are tuned on the validation sets of REBEL, respectively WikipediaNRE.

We use `distilbert-base-cased` for the mention recognizer, and `all-MiniLM-L12-v2` for the bi-encoder, cross-encoder and relation extractor. While larger models potentially perform better, due the efficiency objective and the fact that we are a small university lab, we rely on such lightweight models. We train each model for 10 epochs on two NVIDIA A6000s and select the best-performing model by evaluating on the validation datasets. We use a learning rate of $2 \cdot 10^{-5}$ for all models.

6.3.1 CIE Evaluation

For the closed information extraction task, we compare our trained model, denoted as DISCIE, to the same models as used in the works by Josifoski et al. (Josifoski et al., 2022). GenIE is the SOTA model by Josifoski et al. utilizing a generative model trained from scratch. GenIE-PLM is the same model but initialized from pre-trained BART (Lewis et al., 2020). SetGenNet (Sui et al., 2021) is a encoder-decoder-based model utilising bi-partite matching for extracting triples. Finally, SOTA-Pipeline is a pipeline-based model by Josifoski et al. relying on a sequence of SOTA models for the tasks of mention recognition, entity linking and relation extraction. For more information on this pipeline, please refer to their paper. [36]

We report micro/macro precision, recall and F1 for all models as well as F2 for our model. Micro refers here to calculating the metric over all examples while macro calculates the metrics first for each relation separately and then averages them.

35. When evaluating on GeoNRE or WikipediaNRE, we limited the set of available predictable relations and entities to the same set as used in the work by Josifoski et al. (Josifoski et al., 2022). Therefore, we set prediction scores for out-of-scope relations to 0.0.

36. We did not compare to SCICERO (Dessì et al., 2022) as we were not able to adapt their code to our datasets.

Similar to Josifoski et al., we report the metrics with a 1-standard-deviation confidence interval constructed from 50 bootstrap samples of the data for all results.

In Table6.1, we show the results on the REBEL and the FewRel datasets. Our method outperforms the best-performing method GenIE by more than 5 F1-measure points.

It can be seen that DISCIE has much higher precision while lacking recall in comparison to GenIE. When tuning the thresholds for F2 instead F1 on the validation dataset [37], we see that the recall on the subset of data surpasses GenIE while also surpassing it in overall F1. On the recall-only benchmark FewREL, the F2-calibrated DISCIE performs slightly better than GenIE-PLM while the F1-calibrated DISCIE performs much worse. This is the case as the F1-calibrated DISCIE puts more emphasis on precision which leads to a reduced recall.

Table 6.2 presents the results for GeoNRE and WikipediaNRE. On GeoNRE, DISCIE performs nearly 3 F1 points worse while on WikipediaNRE it is only slightly better.

On REBEL, macro F1 of DISCIE surpasses the second-best method GenIE by nearly 7 points (see Table 6.4) with the F1 calibrated method and by more than 9 points with the F2 calibrated one. This means DISCIE performs more uniformly than GenIE across all relation types. This is especially important due to the large number of relations occurring in the dataset where many are long-tail relations.[38]

Table 6.4: Results on REBEL (Macro)

Model	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$	$F2_{\text{Macro}}$
SOTA-Pipeline	$12.20_{\pm 0.35}$	$10.44_{\pm 0.22}$	$9.48_{\pm 0.21}$	-
GenIE	$33.90_{\pm 0.73}$	$30.48_{\pm 0.65}$	$30.46_{\pm 0.62}$	-
DISCIE (F2 calibrated)	$35.84_{\pm 0.59}$	$\mathbf{43.99_{\pm 0.61}}$	$\mathbf{39.50_{\pm 0.56}}$	$\mathbf{42.08_{\pm 0.57}}$
DISCIE (F1 calibrated)	$\mathbf{44.05_{\pm 0.84}}$	$42.29_{\pm 0.62}$	$37.27_{\pm 0.67}$	$34.11_{\pm 0.63}$

On WikipediaNRE and GeoNRE, while DISCIE sometimes underperformed or only matched the performance of GenIE on micro metrics, we see that in regard to macro metrics, it outperforms GenIE (see Table 6.5). On WikipediaNRE it outperforms GenIE by 8 points on F1. On GeoNRE, DISCIE surpasses GenIE by more than 3 F1 points. DISCIE is therefore also performing more uniformly on those datasets.

Figure 6.2 compares the F1 for all relations separated by their number of occurrences in the training data on REBEL. As can be seen, the performance of DISCIE is surpassing the performance of GenIE consistently. For long-tail entities with an occurrence count between 16 and 64 (2^4 and 2^6), the performance sometimes nearly doubles.

37. Hence putting more emphasis on recall.
38. They occur only rarely in the training data.

Table 6.5: Results on GeoNRE and WikipediaNRE (Macro)

Model	GeoNRE			WikipediaNRE		
	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$
SOTA-Pipeline	38.67±5.72	34.49±5.99	35.14±5.09	24.12±1.46	16.55±1.00	17.76±1.01
GenIE	**75.77±7.80**	71.60±7.95	72.59±7.32	52.55±2.12	45.95±1.67	47.08±1.68
DISCIE	73.65±6.61	**76.72±6.54**	**75.05±6.01**	**53.76±2.14**	**51.80±2.05**	**52.75±1.90**

6.3.2 Ablation

To identify what aspects of the relation extractor contributed the most to the performance, we conducted an ablation study on REBEL. Table 6.6 compares regular DISCIE to DISCIE without any type information (w/o types), DISCIE without candidate descriptions (w/o desc.) and DISCIE with coarse-grained types (w/ coarse).

Excluding type information (w/o types) leads to a large decrease in performance. Especially the precision decreases by many points. Therefore, implicit KG information given by type information helps the model to more precisely decide on the right relation while filtering out relations not compatible with the types provided by the entities.

Not using candidate information (w/o candidate description) and only relying on the textual information at hand leads to a slight decrease in performance by around 0.5 F1 points. Hence, the information contained in the description is not fully replaced by the available type information.

Table 6.6: Ablation study of the relation extractor evaluated over REBEL dataset (w/o types: relation extractor does not use type information, w/o desc.: relation extractor does not use candidate descriptions, w/o text: relation extractor does neither use candidate descriptions nor the input text, w/ coarse: regular relation extractor but coarse-grained types are used)

Model	P	R	F1
DISCIE w/o types	62.41±0.07	69.08±0.08	65.58±0.06
DISCIE w/o desc.	76.82±0.11	72.14±0.07	74.41±0.07
DISCIE w/o text	59.75±0.24	35.87±0.09	44.83±0.10
DISCIE w/ coarse	68.32±0.08	68.31±0.08	68.32±0.06
DISCIE	**77.41±0.11**	**72.68±0.08**	**74.97±0.08**

Only using type information (w/o text) and not relying on any textual or candidate description information leads to the largest decrease in performance. The model is still able to often predict the correct relation by just using the available type information. Some combination of types therefore strongly predict the occurrence of certain relations in the text. However, the task is not trivial and therefore textual information at hand is still a necessity.

Lastly, replacing the fine-grained types with coarse-grained types of form `PER`, `ORG`, `LOC`, `MISC` (w/ coarse) leads to an increase in performance in comparison to

not using type information at all. Nevertheless, using fine-grained types increases the performance much more.

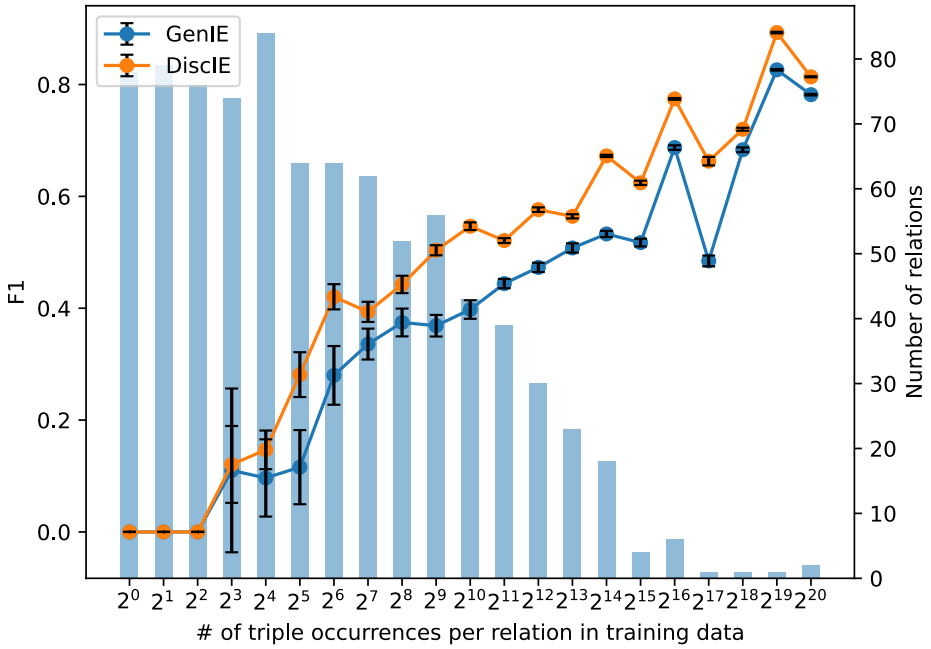


Figure 6.2: F1 for GenIE and DISCIE over REBEL plotted for buckets of relations; each bucket contains all relations occurring a specific number of times in the training data. Each blue bar shows the number of relations occurring in the # of triples as given by the x-axis (see right vertical axis).

6.3.3 Efficiency

We evaluate the efficiency via the GeoNRE dataset by running GenIE and DISCIE three times on its evaluation dataset.[39] Due to the length differences of the examples, the average number of seconds per 1000 examples can vary between datasets. In the GeoNRE dataset, DISCIE is approximately 27 times as fast as GenIE while outperforming it or matching it on several benchmarks (see Table 6.7).

Table 6.7: Efficiency on GeoNRE dataset run on a single NVIDIA A6000 GPU

Model	Seconds/1000 Examples
DISCIE	$21.17_{\pm 0.62}$
GenIE	$571.95_{\pm 7.08}$

39. GenIE takes a long time to evaluate on the other datasets on a single GPU. Therefore we opted for only running the efficiency tests on the smallest dataset. While the average speed differs between the datasets, DISCIE was considerably faster for all of them.

6.3.4 Error Analysis

Figure 6.3 shows what components amount to what percentage of error on REBEL. As can be seen, is the candidate generation the least prone to errors. Usually, the correct candidate is in the generated candidate set. Candidate ranking is more prone to errors than the candidate generation. Sometimes, the wrong candidate is ranked the highest. The components that contribute the most to the errors are the relation extraction and mention recognition.

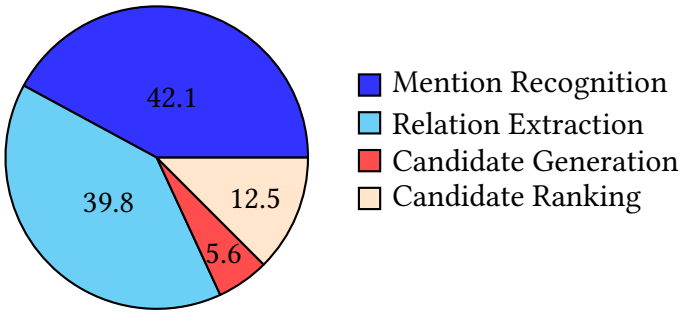


Figure 6.3: Error distribution over all components on REBEL

Additionally, we compare the results for three different examples for both GenIE and DISCIE in Table 6.8. Example 1 shows that DISCIE often performs better when focusing on long-tail relations. Here, DISCIE predicts both the relations `employer` and `musical conductor` while GenIE only predicts `member of`. While `member of` is close to `employer`, `employer` is more specific. On the other hand, `musical conductor`, a long-tail relation, is not predicted by GenIE while DISCIE can predict these. In Example 2 GenIE predicted the correct relation `employer` while DISCIE mistakenly predicted `educated at`. Here, `educated at` can also be seen as a fitting relation but it was just not labeled. Note, that this a common occurrence with GenIE, namely triples are predicted that are reasonable but not labeled. Lastly, Example 3 shows a case where both GenIE and DISCIE fail. Here, both methods generate more triples than necessary. Most of them describe implicit relations. A notable exception is the triple `(Spain, capital, Madrid)` that is predicted by both models. This relation is not stated in the input text but both models likely just predict it due to it often being seen during training.

Table 6.8: Comparison of the performance of DISCIE and GenIE on three different examples. Ground-truth triples are shown in bold

Method	DISCIE	GenIE
Ex. 1	In 2009, Vásquez was named a Gustavo Dudamel conducting fellow with the Los Angeles Philharmonic.	
Result 1	**(Gustavo Dudamel, employer, Los Angeles Philharmonic), (Los Angeles Philharmonic, musical conductor, Gustavo Dudamel)**	(Gustavo Dudamel, member of, Los Angeles Philharmonic)
Ex. 2	She earned her Ph.D in mathematics from the University of Illinois at Urbana–Champaign in 1919 under the supervision of Arthur Byron Coble.	
Result 2	(Arthur Byron Coble, educated at, University of Illinois at Urbana–Champaign)	**(Arthur Byron Coble, employer, University of Illinois at Urbana–Champaign)**
Ex. 3	The Santiago Bernabéu Stadium (,) is a football stadium in Madrid, Spain.	
Result 3	**(Santiago Bernabéu Stadium, sport, Association football), (Santiago Bernabéu Stadium, located in the administrative territorial entity, Madrid)**, (Santiago Bernabéu Stadium, country, Spain), (Madrid, country, Spain),(Spain, capital, Madrid), (Santiago Bernabéu Stadium, instance of, stadium)	**(Santiago Bernabéu Stadium, sport, Association football), (Santiago Bernabéu Stadium, located in the administrative territorial entity, Madrid)**, (Santiago Bernabéu Stadium, country, Spain), (Madrid, country, Spain),(Spain, capital, Madrid)

6.4 Related Work

Entity Linking has a long history of research (Möller et al., 2022). Recent methods can be categorized into two types. First, discriminative methods that are based on the bi-encoder / cross-encoder pairing (Ayoola et al., 2022; Logeswaran et al., 2019; Wu, Petroni, et al., 2020). Both encoders are commonly BERT-like models. The bi-encoder encodes the description of each entity and matches it to the text by using an approximate nearest neighbor search. This is important as the next step, the cross-encoding, is expensive. Here, those neighbors are reranked by applying a cross-encoder to the concatenation of both, the input text and the entity description. The highest-ranked entity is then the final linked one. In the past, type information was used in several works in the entity linking domain. Incorporating it lead to a large increase in performance (Ayoola et al., 2022; Raiman, 2022; Raiman and Raiman, 2018). In contrast to that, we do not employ type information during entity linking but during relation extraction. Another type of entity linker is based on generative models (Cao et al., 2021; De Cao et al., 2022). Here, instead of using some external description of an entity, the whole model memorizes the knowledge

graph (KG) during training. The linked entity is then directly generated by the model. Such methods skip the problem of mining negatives which are crucial for a good performance of bi-encoder-based methods (Cao et al., 2021).

Relation extraction methods usually assume that the entities in the input text are already identified. The task is then to classify whether a relation between two entities is expressed in the text and if it is, what relation holds. Recent methods rely either on CNN (dos Santos et al., 2015; Nguyen and Grishman, 2015; Zeng et al., 2014), RNN (Miwa and Bansal, 2016; Ni and Florian, 2019) or transformer networks (Baldini Soares et al., 2019; Zhong and Chen, 2021). Also, generative models are applied, usually by extracting entities and relations jointly (Paolini et al., 2021; Zhang et al., 2020) but also methods solely focusing on relation extraction (RE) exist (Huguet Cabot and Navigli, 2021; Ni et al., 2022). In contrast to DISCIE, these methods generally focus on a small number of relations and do not consider the entity linking task. Zhang et al. (Zhang et al., 2022) include fine-grained information into a generative joint entity and relation extraction method. But in contrast to us they only focus on entity extraction and not entity linking. Furthermore, they only incorporate a single type per entity.

There exist two directions of research related to closed information extraction. First, pipeline-based approaches. For that, initially, the entities in the text were recognized, then the relations between the entities were identified and finally, relations and entities are linked to the KG (Angeli et al., 2015; Chaganty et al., 2017; Galárraga et al., 2014). While the modularity of pipeline-based approaches makes it possible to simply exchange some modules with a better one, they suffer from error propagation. To combat that, recent methods focus on tackling the problem end-to-end (Liu et al., 2018; Sui et al., 2021; Trisedya et al., 2019). Here, each step of the pipeline is jointly executed at once. This enables the models to have interaction between the entity recognition, relation extraction and entity linking process. Lately, generative models like BART, T5, GPT-4 became more popular (Lewis et al., 2020; OpenAI, 2023; Raffel et al., 2020). Usually, the tasks are here simply modeled as the translation of text to text. In 2022, Josifoski et al. (Josifoski et al., 2022; Josifoski et al., 2023) applied such a generative model to the CIE task reaching SOTA. Furthermore, they are the first two evaluate the CIE task on a large dataset with hundreds of relations and millions of entities. Our method is the first discriminative approach focusing on the large-scale closed-information extraction task. In contrast to GenIE by Josifoski et al., we do not rely on a generative model, but a discriminative one. Furthermore, instead of performing relation extraction solely on the textual data, we incorporate the entity candidate information in form of their descriptions and types. Both features prove to be especially valuable when doing the relation extraction task on datasets with a large number of relations.

6.5 Conclusion and Future Work

In this work, we showed that including fine-grained type information into a discriminative closed information extraction method leads to a large improvement. By using the type information, the model can learn the implicit ontological

information contained in the underlying KG. It especially leads to an **increased performance on long-tail relations**. Furthermore, due to the reliance of DISCIE on only smaller language models, it can deliver great performance while being much **more efficient**. This allows our model to match or even surpass the performance of larger end-to-end CIE information models while being much faster.

A generative model such as GenIE can be trained on the closed information extraction task without having access to the entity mention positions. In contrast to that, our training setup relies on them. In future work, we want to investigate whether the model can be modified to skip the mention recognition. Furthermore, the inference procedure is currently performed in a greedy way. We suspect that globally optimizing the disambiguation graph can lead to an increase in performance, which we also want to pursue further in the future. Also, incorporating the type information into the entity linking module might lead to improvement. Finally, analysing which types have a bigger impact on performance is worth exploring.

Limitations

SynthIE (Josifoski et al., 2023) showed that the REBEL dataset suffers from some qualitative problems such as false negatives. We did not compare our method against a larger generative model, such as LLama (Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023b). Although an adapter-fine-tuned variant of such a large language model might potentially outperform our method, it would require a significantly larger parameter count and be less efficient. Our objective was to demonstrate that substantial performance improvements can be achieved even with a smaller parameter count and some external data.

Supplemental Material Statement: Source code for our System is available from: https://github.com/semantic-systems/discie

7

Analyzing the Influence of Knowledge Graph Information on Relation Extraction

Bibliographic Information

Cedric Möller and Ricardo Usbeck. 2025. Analyzing the Influence of Knowledge Graph Information on Relation Extraction. In *The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I,* edited by Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena, 15718:460–480. Lecture Notes in Computer Science. **Best Student Paper Award**. Springer

Abstract

We examine the impact of incorporating knowledge graph information on the performance of relation extraction models across a range of datasets. Our hypothesis is that the positions of entities within a knowledge graph provide important insights for relation extraction tasks. We conduct experiments on multiple datasets, each varying in the number of relations, training examples, and underlying knowledge graphs. Our results demonstrate that integrating knowledge graph information significantly enhances performance, especially when dealing with an imbalance in the number of training examples for each relation. We evaluate the contribution of knowledge graph-based features by combining established relation extraction methods with graph-aware Neural Bellman-Ford networks. These features are tested in both supervised and zero-shot settings, demonstrating consistent performance improvements across various datasets.

7.1 Introduction

Populating an existing knowledge graph (KG) with new information is an essential challenge. An integral subtask for this is relation extraction (RE). RE is the task of identifying the expressed relation between two entities. The focus of this subtask usually resides on a relation expressed in a sentence, document, or between multiple documents. In contrast to that, link prediction (Wang, Qiu, et al., 2021) infers potential relations based on the structure of a knowledge graph (Ji et al., 2022).

In this paper, our goal is to combine both ways of tackling the task, based on the text and based on the graph, in a single framework. For that, we incorporate a Neural Bellman Ford (NBF) network (Zhu et al., 2021) into the RE process, allowing us to include graph-based information while being generalizable to new entities in the knowledge graph. This leads to a model that jointly considers information in the KG and the document.

We study the impact of the KG data on several established datasets, focusing on supervised and zero-shot scenarios. To tackle the supervised and zero-shot scenarios, we use two different versions of the NBF network, one that assumes that all encountered relations in the graph are known before, and another that creates the relation representations on the fly. As we focus solely on RE, we assume that the actual entity mentions in the text are pre-annotated. Therefore, the task is to determine whether a relationship exists between a pair of entities and, if so, identify which specific relationship holds.

Unlike existing methods (Bastos et al., 2021; Jain et al., 2024), our approach does not rely on learned entity embeddings, allowing it to generalize to new entities. Furthermore, our developed post-prediction mechanism shows further improvements in document-level RE. Additionally, we introduce a method that is also suitable for zero-shot settings.

Our contributions are an analysis of the impact of knowledge graph information in the following relation extraction settings:

1. **Supervised Setting**: We investigate how incorporating knowledge graph information enhances model performance. For that, we modify the NBF network to be usable in the relation extraction problem. This improves accuracy by providing richer contextual information and better resolving ambiguities.

2. **Zero-shot Setting**: We explore the role of knowledge graphs in zero-shot relation extraction, demonstrating their potential to enable models to generalize to unseen categories by providing semantic background and auxiliary data for informed predictions.

Code: Our code is available at
https://github.com/semantic-systems/kg-based-re.

7.2 Method

7.2.1 Problem Definition

We target the RE problem. Given a text and a pair of annotated entities within the text, the goal is to identify the relation expressed between the two entities out of the set of all relations R. Depending on the input text, multiple entities might be mentioned in the text. For each pair of entities multiple relations are potentially expressed. We refer to this as document-level RE. If there are only two entities between which the relation needs to be extracted, we refer to it to sentence-level RE.

In a **supervised setting**, the set of potential relations is known during training. It is possible that no relation is expressed between two entities, which can also be interpreted as an additional `no_relation` relation. When one considers the **zero-shot setting**, the assumption is that the set of relations seen during training differs from the set of relations encountered during evaluation. To solve this task, the method has to generalize to unseen relations.

In addition to textual information, we assume the availability of a knowledge graph. A knowledge graph is a graph consisting of nodes and edges. Each node corresponds to a specific entity, either depicting some individual or concept. The edges between nodes are equipped with a relation that denotes some relationship between two nodes. Two nodes linked by a relation are commonly also denoted as a triple. Formally, we define the graph as $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V \times R_G$ is the set of edges with R_G being the edges existing in the graph. R_G does not need to overlap with R. For example, an edge equipped with the `married_to` relation between the subject `Barack Obama` and the object `Michelle Obama` expresses that both those persons are married to each other. Multiple edges may hold between two nodes. In this work, we assume that the corresponding nodes of the entities marked in the text are known and can be used to incorporate KG-internal information.

Our method consists of two main components (see Figure 7.1): the textual module and the graph module. The textual module purely works on the textual input, while the graph module considers the potential relation between two mentioned entities as expressed in the graph.

7.2.2 Textual Module

Supervised. The input text consists of the marked entities where each entity mention is surrounded by $-signs. The textual encoder follows a well-established architecture as proposed by Zhou et al. (Zhou et al., 2021). Initially, the textual input is encoded using an encoder-only model

$$H = \text{Encoder}(X), \quad H = [h_1, h_2, \dots, h_N], \quad h_i \in \mathbb{R}^d.$$

Then, for each of the marked entity mentions m_j, the encoded token of its left-side $-sign is taken, denoted as $h_{m_j} = h_{p_j}$ where p_j is the position of the sign.

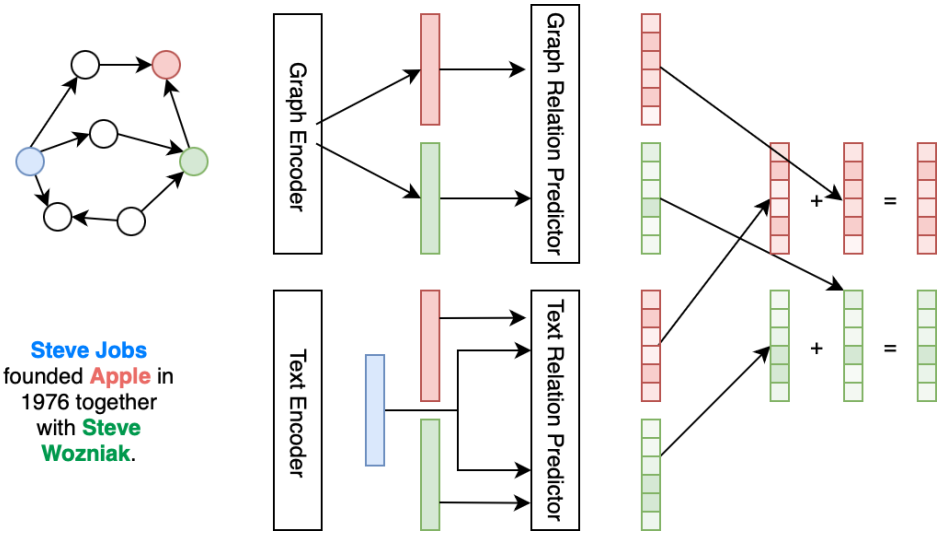


Figure 7.1: Model architecture: The figure illustrates relation prediction between a subject (blue) and two objects (red and green). Text and graph are encoded to predict relations involving Steve Jobs. The graph predictor identifies likely graph-based relations, while the text predictor identifies text-expressed relations, leading to two vectors per entity. Both vectors are combined, giving the final predictions. Identically colored mentions and nodes represent the same entity, and the predictors output a relation distribution, operating in either supervised or zero-shot modes.

Additionally, the attention scores for the \$ to all other tokens throughout all layers are averaged over all layers and normalized

$$\bar{a}_{p_j} = \frac{1}{L} \sum_{\ell=1}^{L} a_{p_j}^{(\ell)}$$

with $a_{p_j}^{(\ell)} = A^{(\ell)}[p_j, :]$. As multiple entity mentions M_{entity} might exist in a document for a single *entity*, the attention scores are averaged over all entity-specific mentions

$$\tilde{a}_{\text{entity}} = \frac{1}{|M_{\text{entity}}|} \sum_{j=1}^{|M|} \bar{a}_{p_j}$$

and the \$-sign encodings are pooled by applying the `logsumexp` operation to get the entity-encoding

$$\tilde{h}_{\text{entity}} = \texttt{logsumexp}(\{h_{m_1}, h_{m_2}, \ldots, h_{m_{|M_{\text{entity}}|}}\}).$$

This gives us two key representations per entity: the attention $\tilde{a}_{\text{entity}}$ to all other tokens and the entity's \$-encoding $\tilde{h}_{\text{entity}}$, which are used in the next steps. For each pair of entities, we compute a combined representation by first point-wise multiplying the attention scores of both entities $\tilde{a}_k \odot \tilde{a}_l$ and normalizing them, giving $\tilde{a}_{k,l}$. Then, we compute an attention-based pair representation by performing a weighted sum over all encoded tokens, resulting in

$$c = \sum_{i=1}^{N} \tilde{a}_{k,l}[i] \cdot h_i.$$

Finally, the subject encoding is calculated by

$$s_k = W_s \text{concat}\left(\tilde{h}_k, c\right) + b_s$$

where W_s is a weight matrix and b_s a bias. The same is done to get an object encoding o_l. Both are used to predict the scores for all relations by applying a bilinear mapping

$$p_t = s_k^\top W o_l \in \mathbb{R}^R$$

where $W \in \mathbb{R}^{d \times R \times d}$ and R is the number of relations.

Zero-shot. For zero-shot learning, we employ an RE approach via multiple-choice classification (Lan et al., 2023). For each document s, we concatenate it with the label l_r and description d_r of a relation r

$$x_r = \text{concat}(s, l_r, d_r).$$

The concatenated input x_r is then fed into an encoder-only model to obtain the token encoding

$$H_r = \text{Encoder}(x_r), \quad H_r = [h_1^{(r)}, h_2^{(r)}, \dots, h_N^{(r)}].$$

We extract the encoding of the first token $h_1^{(r)}$ and project it using a two-layer mapping to a score

$$\text{score}(r) = W_2\, \sigma(W_1 h_1^{(r)} + b_1) + b_2$$

where W_1 and W_2 are weight matrices, b_1 and b_2 are biases, and σ is the tanh function.

This process is repeated for all relations $r \in R$ to compute their respective predictions, giving $p_t \in \mathbb{R}^R$.

7.2.3 Graph Module

The graph module uses the background KG to predict likely relations. The exception is when the Post-Prediction step is also applied (see Section 7.2.5), where both the background KG and text-based triples are used.

Supervised. For the graph-encoder, we follow a modified version of the Neural Bellmann-Ford (NBF) graph neural network (Zhu et al., 2021). Originally designed for link prediction, this method takes a subject entity and a relation as input. Given our dataset's large number of relations, running this model for each relation introduces significant computational overhead, which was not feasible with our available resources.

To address this, we eliminate the need to specify an input relation. Instead, we predict all possible output relations based on the final representation of each node.

Initially, each node in the graph is initialized with a zero vector, except for a designated start node, which is assigned a specific vector g_{start}:

$$g_v^{(0)} = \begin{cases} g_{\text{start}}, & \text{if } v = v_{\text{start}}, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

The graph undergoes T iterations of message passing, where each edge (u, v) is represented by its direction and associated relation r. The message passing consists of three main steps for each node v:

1. Propagation: Information is propagated along the edges using the DistMult operation (Yang et al., 2015)

$$m_{u \to v}^{(t)} = g_u^{(t-1)} \odot r_{(u,v)}$$

where $m_{u \to v}^{(t)}$ is the message from node u to node v, $r_{(u,v)}$ is a relation-specific representation and $\odot$ denotes element-wise multiplication.

2. Aggregation: Incoming messages are aggregated for each node v using Principal Neighbourhood Aggregation (PNA) (Corso et al., 2020)

$$m_v^{(t)} = \text{Aggregate}\left(\{m_{u \to v}^{(t)} \mid u \in \mathcal{N}(v)\}\right)$$

where $\mathcal{N}(v)$ denotes the neighbors of v.

3. Update: The node's representation is updated via a linear projection and a non-linear activation with

$$g_v^{(t)} = \mu(W^{(t)} m_v^{(t)} + b^{(t)})$$

where $W^{(t)}$ and $b^{(t)}$ are the weights and biases of the linear transformation at iteration t and μ is the ReLU activation function.

After T iterations, the message passing is stopped, and the representation of each node is retrieved. This representation captures the relation of any node with respect to the start node.

We initialize the start node as the node corresponding to the subject entity and retrieve the representations of each object entity node g_l. In contrast to the original method, this representation g_l is then passed through a two layer network to predict the scores for all relations instead of a single relation

$$p_g = W_4 \, \mu(W_3 g_l + b_3) + b_4 \in \mathbb{R}^R$$

where W_3, W_4, b_3, and b_4 are weights and biases.

Zero-shot. For zero-shot learning, we rely on the zero-shot variant of the NBF network, denoted as ULTRA (Galkin et al., 2024). ULTRA consists of two components: a relation graph encoder and an entity graph encoder, both implemented as NBF networks.

The relation graph is constructed with nodes representing relations and edges connecting them if the subject/object of one relation is the subject/object of another.

A designated relation is set as the start, and the relation graph encoder produces a representation for each relation node h_r

$$h_r = \text{RelationGraphEncoder}(r_{\text{start}})$$

where $r \in R_G$.

These relation representations h_r are used in the entity graph encoder, which outputs entity representations h_l conditioned on a start node and the relation. These are then fed into a multi-layer projection to compute a single score

$$\text{score}(r) = W_6\, \mu(W_5 h_{\text{entity}} + b_5) + b_6$$

where W_5, W_6, b_5, and b_6 are weights and biases, and μ is the ReLU activation function.

To classify all relations, ULTRA is run for each relation in the set R, and the predictions are concatenated:

$$p_g = [\text{score}(r_1), \text{score}(r_2), \ldots, \text{score}(r_R)] \in \mathbb{R}^R$$

7.2.4 Final Prediction

To compute the final prediction, we integrate the logits from both sources by accumulating them. Let α and β represent the respective weights for each source. The final logits are computed as

$$p = p_t + p_g \in \mathbb{R}^R$$

where $p_t \in \mathbb{R}^R$ and $p_g \in \mathbb{R}^R$ are the logits from the two predictive models (either for the supervised or zero-shot setting), and R represents the number of relations.

7.2.5 Post-Prediction

In document-level RE, rule-learning techniques are frequently employed to enhance the predictive capability of models (Fan et al., 2022; Qi et al., 2024; Ru et al., 2021). These approaches leverage an initial set of relations generated by a text-only model and perform reasoning using learned rules to infer additional relations.

We adopt a similar methodology for document-level relation extraction tasks. Specifically, starting with the initial predictions produced by the textual component, we enrich the underlying graph. This enrichment involves adding new edges to the graph between each pair of nodes (v_i, v_j) whenever a relation is identified between the corresponding entities in the input text.

R_G denotes the set of predefined relation types in the initial graph. To distinguish between a priori known relations and those inferred through reasoning, we define an additional set of relation types R' corresponding exclusively to the predicted relations. The graph $G = (V, E)$ is extended by introducing new edges:

$$E' = \{(v_i, v_j, r') \mid r' \in R', v_i, v_j \in V, \text{ and } r' \text{ is predicted for } (v_i, v_j)\}.$$

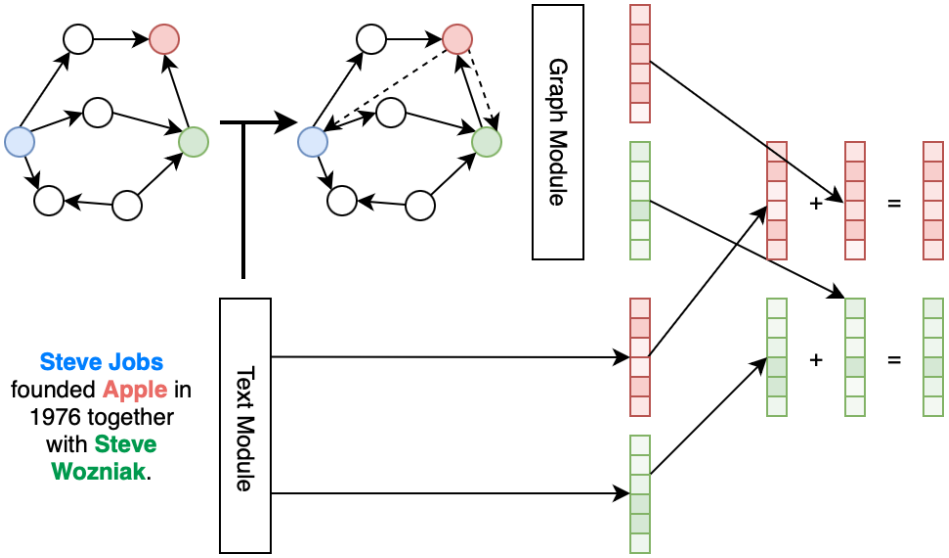


Figure 7.2: The figure illustrates the post-prediction mechanism. The input text is initially processed through the textual module to identify relations. These identified relations are then incorporated into the input graph for the graph module. Using the updated graph and the initial textual predictions, the final predictions are generated.

The extended graph is then represented as $G' = (V, E \cup E')$.

By explicitly encoding these predicted relations as a separate set R', the model can effectively distinguish between the original relations R and the newly inferred ones, thus enabling more nuanced reasoning and relation classification. Post-Prediction is only used for document-level RE. See Figure 7.2 for an overview.

7.2.6 Losses

For datasets with more than two entities per example, we use the HingeABL loss (Wang et al., 2023), which is designed to more-gracefully handle the problem of imbalance in multi-class classification optimization in contrast to a regular binary cross-entropy loss. For sentence-level RE datasets, we use the cross-entropy loss to optimize the models.

7.3 Evaluation

7.3.1 Setup

We evaluate the influence of the graph information on several datasets. As the encoder models, we used RoBERTa-large for Re-DocRED, BERT-base for DWIE, and BioBERT for BioRel. For Wiki-ZSL and FewRel, we relied on BERT-base. We chose those to be comparable to other existing methods.

Supervised datasets. As supervised datasets, we use the Re-DocRED (Tan, Xu, et al., 2022), DWIE (Zaporojets et al., 2021) and BioRel (Xing et al., 2020) dataset. We chose those datasets as they are available with annotated entity mentions and

have been evaluated by past methods that utilised KG information. An overview of them can be found in Table 7.1.

For Re-DocRED and DWIE, we rely on Wikidata as the corresponding knowledge graph. For BioRel, we rely on two available ontologies: Medline (Yang, 2003) and NCIt (Kumar and Smith, 2005).

We gathered each ontology and linked the entity mentions to the corresponding nodes. On Re-DocRED, 83% of mentions matched a Wikidata node, while on DWIE, only 53% did. The rest were literals or unlinked during dataset creation. On BioRel, 82% of mentions were covered; the rest were unidentifiable in the available ontologies. DWIE and Re-DocRED are document-level RE datasets, with DWIE containing longer documents. BioRel is a sentence-level RE dataset.

Our main metric is F1. For Re-DocRED and DWIE, we use micro F1, where true positives are correctly classified relations, false negatives are relations missed between entities, and false positives are incorrect relation predictions. For Re-DocRED, we also report Ign-F1, which excludes triples seen during training.

For BioRel, we use Macro-F1, averaging F1 over all relation classes.

Each method is trained three times, and we report the averaged metrics.

Table 7.1: Supervised RE datasets.

Dataset Name	# Documents	# Relations	# Mentions	# Triples
DWIE	802	65	43,373	317,204
Re-DocRED	4053	97	132,375	120,539
BioRel	533,560	125	1,067,120	533,560

Zero-shot datasets. For the zero-shot evaluations we rely on the popular FewRel (Chia et al., 2022; Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021) datasets (see Table 7.2 for an overview). These are sentence-level RE datasets. Both contain annotated entity mentions linked to Wikidata (Vrandečić and Krötzsch, 2014). The datasets are prepared in three versions, with 5, 10, or 15 test relations. Furthermore, for each of the versions, the full dataset is resampled five times. That means, the model is evaluated on five different runs for each version to compensate for the high variance due to the small number of test relations. All mentioned entities are linked to Wikidata. Macro F1-measure is calculated to evaluate the performance of methods.

Table 7.2: Zero-shot RE datasets.

Dataset Name	# Documents	# Relations	# Mentions	# Triples
FewRel	70,000	100	140,000	70,000
Wiki-ZSL	94,383	113	132,375	188,766

Graph. For each entity in a document, we gather its two-hop neighborhood in the knowledge graph, limiting edges to 100 per hop, resulting in up to 10,000 entities per neighborhood. This enables exploration of four-hop connections. We remove nodes appearing in only one triple unless they belong to the original entity set, helping manage datasets with up to 200,000 nodes per subgraph.

For sentence-level RE, we remove direct triples between subjects and objects to prevent trivial relations, particularly in distantly-supervised datasets.[40] There, 80–90% of triples overlap between graph and text, whereas in document-level RE, less than 45% do. In the zero-shot setting, we sample 1000 triples per relation, extract two-hop neighborhoods, and construct the relation graph, filtering noise by keeping only relation-relation edges found in at least 10% of sampled neighborhoods.

The GNN runs four layers of message passing to utilize the max-hop distance between entities. More hops did not improve performance.

Methods. In our analysis on DWIE, we evaluate a range of document-level RE methods, referencing works such as (Vashishth et al., 2018; Verlinden et al., 2021; Xu et al., 2021). We specifically focus on comparing the performance of the top models utilizing rule-learning techniques, as highlighted in (Fan et al., 2022; Qi et al., 2024; Ru et al., 2021). Additionally, we incorporate two methods that integrate knowledge graph information, as explored by (Bastos et al., 2021; Jain et al., 2024).

On ReDocRED, we compare against several BERT-based and RoBERTa-based methods (Ma et al., 2023; Tan, He, et al., 2022; Xu et al., 2021; Zhang et al., 2021; Zhou et al., 2021) again including two that incorporate knowledge graph information (Bastos et al., 2021; Jain et al., 2024) as well.

On BioREL we compare against the best-performing non-biomedical (Bastos et al., 2021; Raffel et al., 2020; Sorokin and Gurevych, 2017; Zhu et al., 2019) and biomedical RE methods (Jain et al., 2023) as reported by Jain et al. (Jain et al., 2023).

On Wiki-ZSL and Fewrel, we compare against several zero-shot methods, ranging from entailment-based methods (Rocktäschel et al., 2016), encoding-based methods (Chen and Li, 2021; Tran et al., 2023; Tran et al., 2022; Zhao, Zhan, et al., 2023), generative methods (Chia et al., 2022), and discriminative prompting methods (Lan et al., 2023; Lv et al., 2023)

In our supervised experiments, we refer to our method as ATLOP-KG when incorporating the knowledge graph component into the ATLOP architecture, and as ATLOP-KG-PP when including both the KG component and the post-prediction module. Additionally, we present our baseline results using the ATLOP architecture with the HingeABL loss, labeled as ATLOP-Hinge. Furthermore, we denote a method solely using the graph component as NBF.

40. Distant supervision annotates documents using the KG, so keeping these triples could let the model simply verify existing relations.

For the zero-shot experiments, we designate our approach as MC-BERT-KG, and we also present results for MC-BERT with added descriptions, referred to as MC-BERT w/ descriptions.

Hyperparameters. All training and inference runs were performed on a single NVIDIA A6000 GPU. We used a batch size of 8 for each supervised training run and a batch size of 16 for the unsupervised runs. We use learning rates of $3e-5$ for the text encoders and of $1e-4$ for the graph encoders. We trained each method by relying on early stopping based on the validation F1-measure.

7.3.2 Results

Table 7.3: F1 Scores on DWIE.

Model	F1
DRN*GloVe (Xu et al., 2021)	56.04
RESIDE (Vashishth et al., 2018)	66.78
RECON (Bastos et al., 2021)	66.94
KB-Graph (Verlinden et al., 2021)	66.89
ATLOP (Zhou et al., 2021)	75.13
DocRE-CLiP (Jain et al., 2024)	67.10
LogicRE-ATLOP (Ru et al., 2021)	75.67
MILR-ATLOP (Fan et al., 2022)	76.51
JMLR-ATLOP (Qi et al., 2024)	77.85
ATLOP-Hinge (Ours)	77.35
NBF (Ours)	44.27
ATLOP-KG (Ours)	<u>78.50</u>
ATLOP-KG-PP (Ours)	**79.46**

Table 7.4: F1 Scores on ReDocRED.

Model	Ign-F1	F1
ATLOP (Zhou et al., 2021)	76.82	77.56
DRN* (Xu et al., 2021)	74.3	75.6
KG-DocRE (Tan, He, et al., 2022)	80.32	81.04
DocuNet (Zhang et al., 2021)	78.52	79.64
DREEAM (Ma et al., 2023)	<u>80.39</u>	<u>81.44</u>
DocRE-CLiP (Jain et al., 2024)	**80.57**	**81.55**
ATLOP-Hinge (Ours)	76.97	78.07
NBF (Ours)	46.94	49.58
ATLOP-KG (Ours)	77.70	78.70
ATLOP-KG-PP (Ours)	77.80	78.83

Supervised. Our method outperforms recent state-of-the-art methods on the DWIE dataset by 1.5 F1-measure points, as illustrated in Table 7.3. Notably, even without graph information, our text-based RE architecture performs comparably

to existing state-of-the-art models. The inclusion of graph information further enhances performance, which is remarkable given our reliance on the same architectural framework as the state-of-the-art. While including KG information already leads to a larger improvement, using the post-prediction step also improves it even more. We hypothesize that our adaptations for handling long documents, typical in DWIE, contribute significantly to this improvement. This adaptation involves splitting input documents based on the encoder's maximum token length while introducing a stride to maintain contextual coherence. This approach seems to compensate for adjustments potentially overlooked in previous state-of-the-art methods. Without that, our performance diminishes to the one as reported in the paper by Qi et al.(Qi et al., 2024).

Table 7.5: Accuracy and F1 Scores on BioREL.

Model	**Accuracy**	**Macro F1 Score**
GPGNN (Zhu et al., 2021)	85	84.00
ContextAware (Sorokin and Gurevych, 2017)	89	87.00
T5 (Raffel et al., 2020)	88	86.00
RECON (Bastos et al., 2021)	89.6	86.00
ReOnto (Jain et al., 2023)	92	90.00
NBF	75.35	74.74
ATLOP-Hinge (Ours)	<u>96.69</u>	<u>95.71</u>
ATLOP-KG (Ours)	**97.93**	**96.90**

Conversely, our approach does not achieve state-of-the-art performance on the Re-DocRED dataset (Table 7.4). Although incorporating KG information enhances performance over the text-only model, the post-prediction step contributes minimally. This outcome may stem from the typically smaller document sizes in Re-DocRED, reducing the advantage offered by the KG enhancements. Additionally, our lack of pre-training and evidence fusion, which distinguish the leading approaches (Ma et al., 2023), might explain this discrepancy. Nevertheless, we chose to forego these complex stages to minimize computational overhead, focusing instead on exploring the fundamental effects of graph integration. The unexpectedly high performance of DocRE-CLiP (Jain et al., 2024) is notable, particularly given their claims regarding the significant impact of integrating graph information. However, upon reviewing their paper and accompanying code, we were unable to ascertain the specific methods they employed to incorporate this information. Additionally, there is an indication that their training process encompasses entities from not only the training set but also the test and validation sets.

While further inspecting the performance of our method on Re-DocRED, we also saw that the biggest difficulty on the document-level RE datasets was to identify whether any relation is expressed between two entities, not which. Only six percent of errors on Re-DocRED stem from the problem of disambiguating between different relations. We assume that graph information is less helpful to decide whether any relation is expressed as this is more a problem related to the textual RE component. An additional cause might be that the number of relations

Table 7.6: Results on FewRel and Wiki-ZSL.

		Wiki-ZSL			FewRel		
m	Model	P	R	F1	P	R	F1
5	CIM (Rocktäschel et al., 2016)	49.63	48.81	49.22	58.05	61.92	59.92
	ZS-BERT (Chen and Li, 2021)	71.54	72.39	71.96	76.96	78.86	77.90
	Tran et al. (2022)	87.48	77.50	82.19	87.11	86.29	86.69
	RelationPrompt NG (Chia et al., 2022)	51.78	46.76	48.93	72.36	58.61	64.57
	RelationPrompt (Chia et al., 2022)	70.66	83.75	76.63	90.15	88.50	89.30
	RE-Matching (Zhao, Zhan, et al., 2023)	78.19	78.41	78.30	92.82	92.34	92.58
	DSP-ZRSC (Lv et al., 2023)	94.1	77.1	84.8	93.4	92.5	92.9
	Tran et al. (2023)	**94.50**	**96.48**	**95.46**	**96.36**	**96.68**	**96.51**
	MC-BERT (Lan et al., 2023)	80.28	84.03	82.11	90.82	90.13	90.47
	MC-BERT w/ descriptions (Ours)	85.00	84.41	84.68	93.33	92.50	92.91
	MC-BERT-KG (Ours)	88.89	89.46	88.92	88.39	91.37	92.02
10	CIM	46.54	47.90	45.57	47.39	49.11	48.23
	ZS-BERT	60.51	60.98	60.74	56.92	57.59	57.25
	Tran et al. (2022)	71.59	64.69	67.94	64.41	62.61	63.50
	RelationPrompt NG	54.87	36.52	43.80	66.47	48.28	55.61
	RelationPrompt	68.51	74.76	71.50	80.33	79.62	79.96
	RE-Matching	74.39	73.54	73.96	83.21	82.64	82.93
	DSP-ZRSC	80.0	74.0	76.9	80.7	**88.0**	84.2
	Tran et al. (2023)	**85.43**	**88.14**	**86.74**	81.13	82.24	81.68
	MC-BERT	72.81	73.96	73.38	86.57	85.27	85.92
	MC-BERT w/ descriptions (Ours)	74.89	76.05	75.46	85.16	83.36	84.24
	MC-BERT-KG (Ours)	81.72	80.52	81.10	**88.63**	84.80	**86.63**
15	CIM	29.17	30.58	29.86	31.83	33.06	32.43
	ZS-BERT	34.12	34.38	34.25	35.54	38.19	36.82
	Tran et al. (2022)	38.37	36.05	37.17	43.96	39.11	41.36
	RelationPrompt NG	54.45	29.43	37.45	66.49	40.05	49.38
	RelationPrompt	63.69	67.93	65.74	74.33	72.51	73.40
	RE-Matching	67.31	67.33	67.32	73.80	73.52	73.66
	DSP-ZRSC	77.5	64.4	70.4	82.9	78.1	80.4
	Tran et al. (2023)	64.68	65.01	65.30	66.44	69.29	67.82
	MC-BERT	65.71	67.11	66.40	80.71	79.84	80.27
	MC-BERT w/ desc (Ours)	68.53	69.81	69.16	79.22	78.19	78.69
	MC-BERT-KG (Ours)	**79.28**	**76.95**	**78.07**	**84.46**	**80.90**	**82.64**

for both datasets is rather small being fewer than 100. Therefore, the ambiguity between different relations might be rather low.

On BioRel (see Table 7.5), our method performs the best. The textual RE component has again a large impact; however, the inclusion of graph information leads to a

Table 7.7: Ablation on Wiki-ZSL $m = 15$.

Model	P	R	F1
MC-BERT-KG - graph only	69.63	66.16	67.78
MC-BERT-KG - 1 - hop	70.00	71.06	70.05
MC-BERT-KG - 2 - hop	76.74	75.95	76.25
MC-BERT-KG - 3 - hop	75.44	75.46	75.44
MC-BERT-KG - with direct triples	76.95	75.57	76.25
MC-BERT-KG	**79.28**	**76.95**	**78.07**

performance far beyond the previous SOTA. We suspect that the impact is higher on this dataset as there are more relations to disambiguate in comparison to DWIE and Re-DocRED.

Zero-shot. In zero-shot RE, the influence of graph information exhibits variation across tasks. As shown in Table 7.6, graph information significantly enhances performance in the most challenging scenario ($m = 15$) for the Wiki-ZSL dataset. While its impact on FewRel is comparatively modest, our method still surpasses the previous state-of-the-art by two F1-measure points. This indicates that graph information is particularly beneficial when data presents higher complexity or ambiguity, whereas its advantage diminishes in simpler scenarios such as $m = 5$ or $m = 10$. In almost every setting, except for FewRel with $m = 5$, incorporating graph information results in a substantial performance enhancement compared to not utilizing it, as evident when comparing to MC-BERT w/ descriptions. It is important to note that integrating graph information is independent of other model improvements, meaning that the current state-of-the-art methods across several datasets could potentially be further enhanced by incorporating this additional information as well.

7.3.3 Ablation studies

In addition to the ablation studies already included in the previous section, we further investigate our method on the $m = 15$ split of the Wiki-ZSL dataset. Specifically, we analyse the influence of the number of hops and keeping the links between subject and object entities.

Analyzing the ablation results for using only the graph model (Table 7.7), we observe competitive performance with other state-of-the-art approaches, indicating that the KG context provides substantial cues to discern the correct relation, especially in scenarios with a clear single correct relation. However, augmenting this with textual information results in a substantial performance increase of approximately 12 F1-measure points. This underscores the complementary nature of text and graph information, highlighting the importance of leveraging both to maximize RE efficacy. One advantage of the Wiki-ZSL dataset is that a relation is consistently expressed between any two entities. If this were not the case, the model would have to rely more heavily on textual context, making the graph information less dependable. This is evident in the significantly poorer

performance of KG-only methods on document-level RE datasets, as reflected in Tables 7.3 and 7.4.

If we use fewer than 4 hops, we see a diminishing performance. Surprisingly, only using 2-hops outperforms using at maximum 3-hops of path lengths. As the 1-hop distance is effectively not using any knowledge graph information due to us filtering the single hops out, its performance reduces to the text-only model. Interestingly, including direct triples during training actually reduces performance. We attribute this to the model's tendency to rely on straightforward single-hop information rather than considering the broader context of the surrounding neighborhood, which is not always the optimal approach.

7.4 Related Work

Regular RE is usually tackled as a classification problem. The input text is encoded, and a classification head is attached. To encode text, CNNs (Zeng et al., 2014), RNNs (Miwa and Bansal, 2016) or transformers (Zhong and Chen, 2021) are usually employed. Recently, pre-trained models have been extensively used which are fine-tuned on the RE task (Zhang et al., 2021; Zhou et al., 2021).

Document-level RE is tackled mostly in two different ways: either by improving the capabilities of pre-trained language models (PLMs) to identify expressed long-range relations (Zhang et al., 2021; Zhou et al., 2021) or by representing the text information in a more structured way by either modeling the document as a graph (Xu et al., 2021; Zeng et al., 2020) or by learning additional reasoning rules (Fan et al., 2022; Qi et al., 2024; Ru et al., 2021). Our method is connected to both as we combine the SOTA-performing models relying on PLMs with the graph representation using graph neural networks.

Regarding zero-shot RE, representation-learning-based methods (Chen and Li, 2021; Tran et al., 2023; Tran et al., 2022; Zhao, Zhan, et al., 2023) usually try to embed textual and relational information in the same vector space. The relational information, such as labels or descriptions, is transformed into a representation of the relation. Representations are learned such that the representation of the true relation resides close to a representation of the text in the vector space, while the false relation representations are pushed further away. Recently, generative language models have been increasingly utilized for the task (Chen et al., 2022; Chia et al., 2022; Lv et al., 2023; Ni et al., 2022). Here, the model is prompted with the input text and information on the potential relations. The model is then fine-tuned to either generate the relation as expressed in the input text or a full triple consisting of the two entities and the relation. Some methods model the problem as a textual entailment problem (Obamuyide and Vlachos, 2018; Rahimi and Surdeanu, 2023; Sainz et al., 2021). The method by Lan et al. (Lan et al., 2023) models relation classification as a multiple-choice problem where the text is encoded with relation information and a score is calculated. This is done for all relations, and the relation with the highest score is taken. As this method can be equipped with knowledge graph information, we extend this method in our zero-shot RE experiments.

RE under the use of knowledge graph information is an underexplored area. Recently, there have been only a few methods investigating this problem. Most methods either incorporate only one-hop information of entities or rely on trained static representations of entities making the generalization to unseen entities difficult (Bastos et al., 2021; Jain et al., 2024; Möller and Usbeck, 2024c; Vashishth et al., 2018; Verlinden et al., 2021). Others linearize the underlying graph information while not considering the structural information (Jain et al., 2023). In contrast, our method can generalize to new entities and even relations while considering multi-hop information.

7.5 Conclusion and Future Work

We showed that incorporating graph information consistently improves RE, particularly when textual components are undertrained due to data scarcity. However, its benefit diminishes in datasets where the main challenge is determining whether a relation exists, such as document-level RE datasets. Thus, this inclusion is especially valuable in zero-shot settings.

A post-prediction step that integrates relations predicted via text also enhances document-level RE, particularly for very long documents.

Future work will focus on improving efficiency. The current subgraph sampling introduces many irrelevant nodes; a more refined strategy would mitigate this. Scalability is another challenge in zero-shot settings, as GNNs must run separately for each relation, causing high computational overhead. Addressing this is crucial for both RE and link prediction. Lastly, the method assumes an existing path between entities, which is reasonable for link prediction. However, leveraging graph information beyond direct paths could be valuable when textual evidence is available.

We omitted LLMs for comparability with state-of-the-art methods. Our approach remains orthogonal to improvements from fine-tuned LLMs, which could also integrate graph components.

8 Conclusion

8.1 Summary and Research Question Answers

In this thesis, we studied the impact of incorporating knowledge graph (KG) information into the task of information extraction, specifically examining two main tasks: entity linking (EL) and relation extraction (RE).

8.1.1 Entity Linking

In Chapter 3, we investigated how recent EL methods utilize KG information. We compiled all EL methods, focusing on Wikidata, up to 2021 and analyzed which elements of the Wikidata KG each method utilizes. This enabled us to answer the first research question:

Research Question 1

How do recent Wikidata-based entity linking methods leverage knowledge graph information?

We showed that recent EL methods only marginally consider the Wikidata KG. While many methods rely on entity label information, entity descriptions are often ignored and replaced with information from Wikipedia. Few methods consider type information within the KG. The actual structural graph information, such as the connections between entities, is utilized in a varying degree by either utilizing knowledge graph embeddings or linearized triples, which are simplified textual representations of graph structures. Furthermore, a unique quality of Wikidata, qualifier information, which provides additional details on facts, is ignored by most methods. Furthermore, we observed that the evolving nature of Wikidata is not accounted for by any of the identified methods.

Building on these findings, we investigated whether KG information has a positive impact on the task of EL in Chapter 4, particularly for out-of-KG entities. We incorporated KG embeddings into the EL approach in a two-stage process. In the first stage, we link entities and determine whether they point to an out-of-KG entity. During this stage, we link iteratively while considering the KG embeddings of all previously linked entities as valuable context information. In the second stage, we cluster the out-of-KG entities while considering the KG embeddings of all surrounding in-KG entities in the same text. We evaluated this approach using two datasets: one newly created and one synthetically modified EL dataset. This setup allows us to answer the second research question:

Research Question 2

How does incorporating knowledge graph information affect the performance of entity linking methods on out-of-KG entities?

While we observed a positive impact of the KG information on entities already in the KG, the benefit for out-of-KG entities was limited. One possible explanation is the small number of in-KG entities surrounding out-of-KG entities, combined with the low expressivity of the KG embeddings employed in this study. KG information can only have a positive impact on the linking of out-of-KG entities if there are enough in-KG entities surrounding it in the text. In the extreme case, when only a single entity, namely the out-of-KG entity, is part of the text, information from the KG is unattainable and hence cannot be incorporated at all.

8.1.2 Relation Extraction

Having investigated the impact of KG information on EL, we next shifted our focus to RE. Here, we examined the impact of fine-grained type information as well as the structural information in the KG in both fully-supervised and zero-shot settings.

We incorporated fine-grained type information in two forms: as textual information and as vector representations.

In Chapter 5, we first incorporated fine-grained type information in textual form into zero-shot RE. In this case, fine-grained types refer to more detailed ones as available in Wikidata, such as *shipyard*, *movie* or similar and are not limited to the usual entity types as encountered in named entity recognition (NER), like *person*, *location*, and *organisation*. We achieved this by adding the text labels of types of the subject and object entity by. The input to a cross-encoder-based method is here the concatenation of the input text, the entity types as well as the label and description of the potential relation. The rationale is that the description of the relation provides cues about the type of entities that usually interact within the scope of the relation.

Given this method, we can answer the third research question:

Research Question 3

What effect does the inclusion of fine-grained entity types have on the task of zero-shot relation extraction?

We demonstrated that the inclusion of type information was effective, as it resulted in a significant boost in performance in the zero-shot setting. This performance increase becomes even more noticeable when faced with a larger number of potential relations to extract. When more relations are available to match against, there is also a greater likelihood that another similar relation might co-occur, where the type information is crucial.

Next, in Chapter 6, we investigated the impact of fine-grained entity-type information in a closed information extraction. This refers to extracting complete linked triples from text. Our main focus was on RE, aiming to improve the extraction of complete linked triples from text by integrating type information into the RE component.

Rather than using the types in their textual form, we represented each type as a vector and included these vector representations of the subject and object entity types within the RE method. For EL, we used token-pairwise span detection for identifying entity mentions, a standard bi-encoder to generate candidates and a cross-encoder for ranking the entities. We investigated the impact on datasets with a varying degree of relations by conducting experiments on a dataset with nearly a thousand relations and others with fewer than a hundred. This analysis addresses the fourth research question:

Research Question 4

How does the utilization of fine-grained entity-type information affect the performance of relation extraction methods with numerous relations?

We observed that EL performs very well on the datasets, but RE remains a challenge. When dealing with hundreds of relations, the inclusion of fine-grained entity-type information produced a noticeable positive impact compared to models without type information and compared to an existing generative approach that extracts triples directly from text. For datasets with fewer relations, the overall performance improvement was less significant. Nonetheless, we still observed a positive effect on long-tail relations, indicating that additional type information benefits relations that are rarely seen during training.

Lastly, in Chapter 7, we explored the impact of structural information on RE by incorporating conditional message passing on the KG into RE methods. We sampled a two-hop neighborhood around each entity in the KG that appears in the text, which resulted in a subgraph with 4-hop paths connecting all entities. We then applied a conditional message passing method to propagate information between entity pairs, thereby generating a representation of their relationship in the KG. This KG-derived representation was then integrated with a RE method that also uses the textual input. We applied two versions of the RE method: one for the

fully-supervised setting and another for the zero-shot setting. We evaluated the performance of these methods on various datasets, ranging from sentence-level to document-level RE datasets. Additionally, we compared the zero-shot approach with the fully-supervised one.

Based on this, we can answer the last research question:

Research Question 5

What impact does the inclusion of structural information from the knowledge graph have in zero-shot and fully-supervised relation extraction settings?

We found that incorporating structural KG information through path information between entities positively impacts RE performance. This effect is especially pronounced under limited supervision and ambiguous relationships, as seen in the comparison between a fully-supervised and zero-shot settings, where the impact in the zero-shot setting is significantly larger. Additionally, we demonstrated that considering shorter paths yields decreased performance.

In summary, we have shown that, across many scenarios, incorporating KG information can significantly aid in the task of information extraction. As the data to be extracted and the underlying background KG can differ considerably, utilizing more information that connects the two is highly beneficial. With the ever-increasing amount of data created every day and vastly differing needs in regard to what to do with the data, being able to generalize to new entities or even relations is vital. It can certainly be said that a solution to this problem has not yet been found, as full generalizability to new texts and KGs has not yet been achieved. However, we were able to contribute to certain aspects of the hopefully not far-distant solution to this.

8.2 Limitations and Future Work

In this section, we discuss the limitations of our current work and outline directions for future research.

8.2.1 Dataset and Knowledge Graph Limitations

We conducted a survey on the use of KG information in EL where we focused only on the Wikidata KG in Chapter 3. While our work focuses on general-purpose, open-domain KGs, there are domain-specific surveys in other areas, such as the biomedical field (Shi et al., 2023). Conducting comparative analyses across different domains and KGs would be a valuable direction for future work.

The same applies to the chapters introducing different approaches. Most of them were evaluated using the Wikidata KG. Exploring the impact of knowledge graph information in other domains and with different KGs would provide valuable insights.

The Wikievents dataset introduced in Chapter 4, which is based on the "Current events" section of Wikipedia, lacks rich context in the form of co-occurring entities. This was the case as the used text examples consisted of short summaries of news articles. Investigating the impact on longer, more realistic documents could be very beneficial.

Finally, there is a lack of datasets focusing on knowledge graph population under realistic settings. Most existing works and their respective dataset address EL and RE problems separately, including their respective subtasks. Zero-shot approaches usually concentrate only on new entities or relations, without considering that existing elements also require linking. Assessing methods that perform both EL and RE using an incomplete background KG is a logical next step, as it accounts for the complexities involved in building a real-world KG. Entities and relations are often missing, making it crucial to study how these methods fare in more realistic scenarios. To achieve this, developing an appropriate dataset is essential as there is currently a lack of one, and we believe this will be a vital step forward.

8.2.2 Modeling and Methodological Limitations

Due to availability reasons, we relied on static KG embeddings in Chapter 4. While this does not limit out-of-KG EL performance, it does limit the zero-shot EL performance on in-KG entities. More recent and powerful GNNs (Busbridge et al., 2019; Galkin et al., 2024; Zhu et al., 2021) outperform static KG embeddings. Investigating such methods is a promising and necessary step for future work. Furthermore, while entity linking for in-KG entities has reached acceptable performance levels, the identification and clustering of out-of-KG entities remain unsatisfactory. As a result, this problem is still far from being solved.

The triple extraction method in Chapter 6 was not trained in an end-to-end setting. Employing end-to-end training could potentially lead to further improvements. Additionally, we incorporated type information primarily in the RE. It has been shown in the literature that integrating it into EL leads to performance increases as well (Ayoola et al., 2022; Raiman, 2022; Raiman and Raiman, 2018). Integrating and training both methods, RE and EL, jointly promises further performance improvements. In regard to the entity type representation we relied on assigning an independent vector representation to each entity type. Exchanging this with other embeddings which consider the hierarchical nature of type information might be beneficial as well (Dai and Zeng, 2023; López et al., 2019).

8.2.3 Efficiency and Scalability Challenges

The way the zero-shot RE methods in Chapters 5 and 7 include KG information is inefficient because the text encoder-only model must process each relation separately. When the number of relations is large, this causes a significant computational overhead. Incorporating KG information in a prototypical learning framework (Snell et al., 2017), which is often used in zero-shot RE, can greatly improve efficiency. In this context, prototypical learning involves creating a latent representation, or prototype, for each relation. This prototype serves as a reference to match actual relation instances. The benefit is that each relation's

representation needs to be computed only once, and then each input example can be compared simply using a similarity measure, such as the dot product.

8.2.4 Integration with Large Language Models (LLMs)

We did not involve large language models (LLMs) in our experiments for several reasons. First, this thesis began in 2021, at a time when open-source LLMs were not yet available. Such models only became widely accessible in 2023, with the release of open-source models like LLaMA (Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al., 2023b). Second, our methodology often relied on fine-tuning, which remained feasible with smaller pretrained models and the hardware available at the time. Third, our work builds on prior approaches that predominantly employed encoder-only architectures, to which our methods are most directly applicable. Nevertheless, many of the techniques developed here are compatible with LLMs, and we expect that the integration of KG information would remain beneficial in that context. In related areas such as question answering, the value of KG integration has already been demonstrated through Graph-RAG and similar approaches (Peng et al., 2024). While there are few generative methods already considering a background KG (Mai et al., 2025; Zhang and Soh, 2024) many only focus on performing open information extraction from text (Das et al., 2024; Prabhong et al., 2024; Zhang, Cao, Wang, et al., 2024). Extending our work to LLMs and decoder-based architectures is therefore a promising direction for future research.

8.3 Final Remarks

This thesis highlights the ongoing importance of structured knowledge in modern natural language processing systems. By systematically studying how different types of KG information affect KG population tasks, we provide insights that can inform future model design, particularly in low-resource and zero-shot settings. As research increasingly moves toward integrating symbolic and neural representations, we hope that this work contributes a meaningful step toward more interpretable, robust, and adaptable information extraction systems.

References

David Abián, Albert Meroño-Peñuela, and Elena Simperl. 2022. An Analysis of Content Gaps Versus User Needs in the Wikidata Knowledge Graph. In *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings,* edited by Ulrike Sattler, Aidan Hogan, C. Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato, 13489:354–374. Lecture Notes in Computer Science. Springer. (Cited on page 18).

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity Linking Via Explicit Mention-Mention Coreference Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, 4644–4658. Seattle, United States: Association for Computational Linguistics. (Cited on pages 4, 8, 35, 87, 94 sq.).

Prabal Agarwal, Jannik Strötgen, Luciano del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. DiaNED: Time-Aware Named Entity Disambiguation for Diachronic Corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* edited by Iryna Gurevych and Yusuke Miyao, 686–693. Melbourne, Australia: Association for Computational Linguistics. (Cited on page 79).

Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. 2020. Named Entity Extraction for Knowledge Graphs: A Literature Overview. *IEEE Access* 8:32862–32881. (Cited on pages 77 sq.).

Marjan Albooyeh, Rishab Goel, and Seyed Mehran Kazemi. 2020. Out-Of-Sample Representation Learning for Knowledge Graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020,* edited by Trevor Cohn, Yulan He, and Yang Liu, 2657–2666. Online: Association for Computational Linguistics. (Cited on page 70).

Paulo Dias Almeida, Jorge Gustavo Rocha, Andrea Ballatore, and Alexander Zipf. 2016. Where the Streets Have Known Names. In *Computational Science and Its Applications - ICCSA 2016 - 16th International Conference, Beijing, China, July 4-7, 2016, Proceedings, Part IV,* edited by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, Carmelo Maria Torre, David Taniar, Bernady O. Apduhan, Elena N. Stankova, and Shangguang Wang, 9789:1–12. Lecture Notes in Computer Science. Springer. (Cited on pages 47, 56).

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Tejasvi Chaganty, Jason Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D. Manning. 2015. Bootstrapped Self Training for Knowledge Base Population. In *Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015.* NIST. (Cited on page 120).

Renzo Angles. 2018. The Property Graph Database Model. In *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, May 21-25, 2018,* edited by Dan Olteanu and Barbara Poblete, vol. 2100. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 14).

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An Efficient Zero-Shot-Capable Approach to End-To-End Entity Linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track,* edited by Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min, 209–220. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics. (Cited on pages 8, 35, 93, 103, 113, 119, 143).

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450. arXiv: 1607.06450. (Cited on pages 23 sq.).

Jinheon Baek, Dong Bok Lee, and Sung Ju Hwang. 2020. Learning to Extrapolate Knowledge: Transductive Few-Shot Out-Of-Graph Link Prediction. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual,* edited by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. (Cited on page 70).

Amit Bagga and Breck Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics.* (Cited on page 94).

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* edited by Anna Korhonen, David Traum, and Lluís Màrquez, 2895–2905. Florence, Italy: Association for Computational Linguistics. (Cited on page 120).

Debayan Banerjee, Debanjan Chaudhuri, Mohnish Dubey, and Jens Lehmann. 2020. PNEL: Pointer Network Based End-To-End Entity Linking Over Knowledge Graphs. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I,* edited by Jeff Z. Pan, Valentina A. M. Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, 12506:21–38. Lecture Notes in Computer Science. Springer. (Cited on pages 56, 63, 69 sqq.).

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. RECON: Relation Extraction Using Knowledge Graph Context in a Graph Neural Network. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021,* edited by Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, 1673–1685. ACM / IW3C2. (Cited on pages 9, 50, 124, 132 sqq., 138).

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA,* edited by Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, 932–938. MIT Press. (Cited on page 22).

Tim Berners-Lee. 2009. Semantic Web and Linked Data. Accessed: 2025-07-20. (Cited on page 15).

G. P. Shrivatsa Bhargav, Dinesh Khandelwal, Saswati Dana, Dinesh Garg, Pavan Kapanipathi, Salim Roukos, Alexander G. Gray, and L. Venkata Subramaniam. 2022. Zero-shot Entity Linking with Less Data. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022,* edited by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, 1681–1697. Association for Computational Linguistics. (Cited on page 8).

Christopher M Bishop and Nasser M Nasrabadi. 2006. Pattern recognition and machine learning. Vol. 4. 4. Springer. (Cited on page 22).

Kevin Blissett and Heng Ji. 2019. Cross-Lingual NIL Entity Clustering for Low-Resource Languages. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference,* edited by Maciej Ogrodniczuk, Sameer Pradhan, Yulia Grishina, and Vincent Ng, 20–25. Minneapolis, USA: Association for Computational Linguistics. (Cited on pages 82, 93).

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic Acids Res.* 32 (Database-Issue): 267–270. (Cited on page 17).

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008,* edited by Jason Tsong-Li Wang, 1247–1250. ACM. (Cited on pages 42, 52).

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. Edited by Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger. *Advances in neural information processing systems* 26:2787–2795. (Cited on pages 26, 59, 85).

Emanuela Boros, Elvys Linhares Pontes, Luis Adrián Cabrera-Diego, Ahmed Hamdi, Nicolas Sidére, and Antoine Doucet. 2020. Robust Named Entity Recognition and Linking on Historical Multilingual Documents, (cited on pages 56, 65, 70 sq.).

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 7833–7845. Online: Association for Computational Linguistics. (Cited on pages 56, 60, 67, 69 sqq., 74, 76).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual,* edited by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. (Cited on page 25).

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-Source Library for Dialogue Systems. In *Proceedings of ACL 2018, System Demonstrations,* edited by Fei Liu and Thamar Solorio, 122–127. Melbourne, Australia: Association for Computational Linguistics. (Cited on page 56).

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational Graph Attention Networks. *CoRR* abs/1904.05811. (Cited on pages 28, 143).

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. 2024. Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces. *ACM Comput. Surv.* 56 (6): 159:1–159:42. (Cited on page 26).

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* OpenReview.net. (Cited on pages 8, 30, 34, 93, 119 sq.).

Taylor Cassidy, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jiawei Han, Dan Roth, and Jing Zheng. 2011. CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011.* NIST. (Cited on pages 82, 93).

Arie Cattan, Sophie Johnson, Daniel S. Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021. Scico: Hierarchical Cross-Document Coreference for Scientific Concepts. In *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021,* edited by Danqi Chen, Jonathan Berant, Andrew McCallum, and Sameer Singh. (Cited on page 94).

Alberto Cetoli, Mohammad Akbari, Stefano Bragaglia, Andrew D. O'Harney, and Marc Sloan. 2018. Named Entity Disambiguation Using Deep Learning on Graphs. *CoRR* abs/1810.09164. (Cited on pages 56 sq., 70 sq.).

Arun Chaganty, Ashwin Paranjape, Percy Liang, and Christopher D. Manning. 2017. Importance Sampling for Unbiased On-Demand Evaluation of Knowledge Base Population. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* edited by Martha Palmer, Rebecca Hwa, and Sebastian Riedel, 1038–1048. Copenhagen, Denmark: Association for Computational Linguistics. (Cited on page 120).

Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a Knowledge Graph to Enable Precision Medicine. *Nature Scientific Data* 10 (1): 67. (Cited on pages 1, 17).

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards Zero-Shot Relation Extraction with Attribute Representation Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, 3470–3479. Online: Association for Computational Linguistics. (Cited on pages 36, 100 sq., 105, 131 sq., 135, 137).

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-Aware Prompt-Tuning with Synergistic Optimization for Relation Extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022,* edited by Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, 2778–2788. ACM. (Cited on pages 105, 137).

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. In *Findings of the Association for Computational Linguistics: ACL 2022,* edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 45–57. Dublin, Ireland: Association for Computational Linguistics. (Cited on pages 100 sq., 105 sq., 131 sq., 135, 137).

Davide Chicco. 2021. Siamese Neural Networks: An Overview. *Artificial neural networks,* 73–94. (Cited on page 110).

Imma Subirats Coll, Kristin Kolshus, Andrea Turbati, Armando Stellato, Esther Mietzsch, Daniel Martini, and Marcia Zeng. 2022. AGROVOC: the Linked Data Concept Hub for Food and Agriculture. *Comput. Electron. Agric.* 196:105965. (Cited on page 17).

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A Framework for Benchmarking Entity-Annotation Systems. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013,* edited by Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon, 249–260. International World Wide Web Conferences Steering Committee / ACM. (Cited on page 67).

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Velickovic. 2020. Principal Neighbourhood Aggregation for Graph Nets. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual,* edited by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. (Cited on page 128).

Hongliang Dai and Ziqian Zeng. 2023. From Ultra-Fine to Fine: Fine-tuning Ultra-Fine Entity Typing Models to Fine-grained. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023,* edited by Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, 2259–2270. Association for Computational Linguistics. (Cited on page 143).

Arunav Das, Nadeen Fathallah, and Nicole Obretincheva. 2024. Navigating Nulls, Numbers and Numerous Entities: Robust Knowledge Base Construction from Large Language Models. In *Joint proceedings of the 2nd workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM 2024) and the 3rd challenge on Language Models for Knowledge Base Construction (LM-KBC 2024) co-located with the 23nd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 12, 2024,* edited by Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jeff Z. Pan, Tuan-Phong Nguyen, and Bohui Zhang, vol. 3853. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 2, 144).

DBpedia. Dbpedia Live. (Cited on page 42).

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual Autoregressive Entity Linking. Edited by Brian Roark and Ani Nenkova. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 10:274–290. (Cited on pages 93, 119).

Antonin Delpeuch. 2020. OpenTapioca: Lightweight Entity Linking for Wikidata. In *Proceedings of the 1st Wikidata Workshop (Wikidata 2020) co-located with 19th International Semantic Web Conference(OPub 2020), Virtual Conference, November 2-6, 2020,* edited by Lucie-Aimée Kaffee, Oana Tifrea-Marciuska, Elena Simperl, and Denny Vrandecic, vol. 2773. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 56, 61, 70 sq., 73).

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2022. Scicero: A Deep Learning and Nlp Approach for Generating Scientific Knowledge Graphs in the Computer Science Domain. *Knowl. Based Syst.* 258:109945. (Cited on page 114).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* edited by Jill Burstein, Christy Doran, and Thamar Solorio, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. (Cited on pages 1, 8, 23 sq., 64).

Reinhard Diestel. 2000. Graph Theory. Electronic Edition 2000. Originally published 1997. New York: Springer Verlag. (Cited on page 27).

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec,* 2:837–840. 1. Citeseer. (Cited on page 9).

Hang Dong, Jiaoyan Chen, Yuan He, Yinan Liu, and Ian Horrocks. 2023. Reveal the Unknown: Out-of-Knowledge-Base Mention Discovery with Entity Linking. Edited by Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos, 452–462. (Cited on page 94).

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014,* edited by Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, 601–610. ACM. (Cited on page 17).

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. 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),* edited by Chengqing Zong and Michael Strube, 626–634. Beijing, China: Association for Computational Linguistics. (Cited on page 120).

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-Quad 2.0: A Large Dataset for Complex Question Answering Over Wikidata and Dbpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II,* edited by Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, 11779:69–78. Lecture Notes in Computer Science. Springer. (Cited on page 71).

Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. 2018. EARL: Joint Entity and Relation Linking for Question Answering over Knowledge Graphs. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I,* edited by Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, 11136:108–126. Lecture Notes in Computer Science. Springer. (Cited on page 9).

Sourav Dutta and Gerhard Weikum. 2015. C3EL: A Joint Model for Cross-Document Co-Reference Resolution and Entity Linking. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* edited by Lluís Màrquez, Chris Callison-Burch, and Jian Su, 846–856. Lisbon, Portugal: Association for Computational Linguistics. (Cited on pages 82, 93).

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020,* edited by Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, vol. 2696. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 57, 64, 71, 73, 76).

Robin Ellgren. 2020. Exploring Emerging Entities and Named Entity Disambiguation in News Articles. Master's thesis, Linköping University. (Cited on page 57).

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018),* edited by Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA). (Cited on page 71).

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. 2022. Self-Supervised Representation Learning: Introduction, advances, and challenges. *IEEE Signal Process. Mag.* 39 (3): 42–62. (Cited on page 23).

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA,* edited by Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, 226–231. AAAI Press. (Cited on page 35).

Angela Fahrni, Benjamin Heinzerling, Thierry Göckel, and Michael Strube. 2013. Hits' Monolingual and Cross-Lingual Entity Linking System at TAC 2013. In *Proceedings of the Sixth Text Analysis Conference, TAC 2013, Gaithersburg, Maryland, USA, November 18-19, 2013.* NIST. (Cited on pages 82, 93).

Shengda Fan, Shasha Mo, and Jianwei Niu. 2022. Boosting Document-Level Relation Extraction by Mining and Injecting Logical Rules. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* edited by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, 10311–10323. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. (Cited on pages 129, 132 sq., 137).

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: On-The-Fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010,* edited by Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, 1625–1628. ACM. (Cited on page 72).

Paolo Ferragina and Ugo Scaiella. 2012. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Softw.* 29 (1): 70–75. (Cited on page 8).

Wikimedia Foundation. 2025. Data Dumps: Dumps Sizes and Growth. Accessed: 2025-02-14. (Cited on page 34).

Evan French and Bridget T. McInnes. 2023. An Overview of Biomedical Entity Linking Throughout the Years. *J. Biomed. Informatics* 137:104252. (Cited on page 8).

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing Open Knowledge Bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014,* edited by Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, 1679–1688. ACM. (Cited on page 120).

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message Passing for Hyper-Relational Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 7346–7359. Online: Association for Computational Linguistics. (Cited on pages 52, 54).

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards Foundation Models for Knowledge Graph Reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024.* OpenReview.net. (Cited on pages 29, 128, 143).

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* edited by Martha Palmer, Rebecca Hwa, and Sebastian Riedel, 2619–2629. Copenhagen, Denmark: Association for Computational Linguistics. (Cited on pages 8, 65).

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards More Challenging Few-Shot Relation Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019,* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 6249–6254. Association for Computational Linguistics. (Cited on pages 9, 36).

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS),* edited by Eunjeong L. Park, Masato Hagiwara, Dmitrijs Milajevs, and Liling Tan, 1–6. Melbourne, Australia: Association for Computational Linguistics. (Cited on page 66).

Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. 2021. Recent Advances in Open Set Recognition: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (10): 3614–3631. (Cited on page 95).

GeoNames Team. 2025. Geonames Geographical Database. Accessed: 2025-05-18. (Cited on page 17).

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017,* edited by Doina Precup and Yee Whye Teh, 70:1263–1272. Proceedings of Machine Learning Research. PMLR. (Cited on pages 26, 28).

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. Http://www.deeplearningbook.org. MIT Press. (Cited on pages 5, 18 sqq., 22).

David Graus, Tom Kenter, Marc Bron, Edgar Meij, and Maarten de Rijke. 2012. Context-Based Entity Linking - University of Amsterdam at TAC 2012. In *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012.* NIST. (Cited on pages 82, 93).

Kara Greenfield, Rajmonda S. Caceres, Michael Coury, Kelly Geyer, Youngjune Gwon, Jason Matterer, Alyssa C. Mensch, Cem Safak Sahin, and Olga Simek. 2016. A Reverse Approach to Named Entity Extraction and Linking in Microposts. In *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016), Montréal, Canada, April 11, 2016,* edited by Aba-Sah Dadzie, Daniel Preotiuc-Pietro, Danica Radovanovic, Amparo Elizabeth Cano Basave, and Katrin Weller, 1691:67–69. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 82, 93).

Desta Haileselassie Hagos, Rick Battle, and Danda B. Rawat. 2024. Recent Advances in Generative AI and Large Language Models: Current Status, Challenges, and Perspectives. *IEEE Trans. Artif. Intell.* 5 (12): 5873–5893. (Cited on page 25).

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge Transfer for Out-Of-Knowledge-Base Entities : A Graph Neural Network Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017,* edited by Carles Sierra, 1802–1808. ijcai.org. (Cited on page 70).

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-Of-The-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* edited by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, 4803–4809. Brussels, Belgium: Association for Computational Linguistics. (Cited on pages 100, 113, 131).

Bahareh Harandizadeh and Sameer Singh. 2020. Tweeki: Linking Named Entities on Twitter to a Knowledge Graph. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020),* edited by Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, 222–231. Online: Association for Computational Linguistics. (Cited on pages 66, 70 sq., 74).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016,* 770–778. IEEE Computer Society. (Cited on pages 23 sq.).

Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism Detection in Wikidata. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016,* edited by Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi, 327–336. ACM. (Cited on page 55).

Nicolas Heist. 2024. Exploiting Semi-Structured Information in Wikipedia for Knowledge Graph Construction. PhD thesis, University of Mannheim. (Cited on pages 2, 30, 35).

Nicolas Heist, Sven Hertling, Daniel Ringler, and Heiko Paulheim. 2020. Knowledge Graphs on the Web - an Overview. In *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges,* edited by Ilaria Tiddi, Freddy Lécué, and Pascal Hitzler, 47:3–22. Studies on the Semantic Web. IOS Press. (Cited on page 78).

Nicolas Heist and Heiko Paulheim. 2023. Nastylinker: Nil-Aware Scalable Transformer-Based Entity Linker. In *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings,* edited by Catia Pesquita, Ernesto Jiménez-Ruiz, Jamie P. McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphaël Troncy, and Sven Hertling, 13870:174–191. Lecture Notes in Computer Science. Springer. (Cited on pages 4, 8, 35, 87, 94).

Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. 2010. Foundations of Semantic Web Technologies. Chapman / Hall/CRC Press. (Cited on page 15).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9 (8): 1735–1780. (Cited on page 57).

Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. 2014. Discovering Emerging Entities with Ambiguous Names. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014,* edited by Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, 385–396. ACM. (Cited on pages 3, 7 sq., 32, 49, 82, 88, 93).

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing,* edited by Regina Barzilay and Mark Johnson, 782–792. Edinburgh, Scotland, UK.: Association for Computational Linguistics. (Cited on pages 8, 63).

Aidan Hogan, Marcelo Arenas, Alejandro Mallea, and Axel Polleres. 2014. Everything You Always Wanted to Know about Blank Nodes. *Journal of Web Semantics* 27:42–69. (Cited on page 15).

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers. (Cited on pages 1, 13 sq., 16).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA,* edited by Kamalika Chaudhuri and Ruslan Salakhutdinov, 97:2790–2799. Proceedings of Machine Learning Research. PMLR. (Cited on page 84).

Binxuan Huang, Han Wang, Tong Wang, Yue Liu, and Yang Liu. 2020. Entity Linking for Short Text Using Structured Knowledge Graph Via Multi-Grained Text Matching. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020,* edited by Helen Meng, Bo Xu, and Thomas Fang Zheng, 4178–4182. ISCA. (Cited on pages 56, 64, 69 sqq.).

Junbo Huang, Longquan Jiang, Cedric Möller, and Ricardo Usbeck. 2024. Event Extraction Alone Is Not Enough. In *Proceedings of Text2Story - Seventh Workshop on Narrative Extraction From Texts held in conjunction with the 46th European Conference on Information Retrieval (ECIR 2024), Glasgow, Scotland, UK, March 24, 2024,* edited by Ricardo Campos, Alípio Mário Jorge, Adam Jatowt, Sumit Bhatia, and Marina Litvak, 3671:105–114. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 12).

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction by End-To-End Language Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021,* edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 2370–2381. Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on pages 113, 120).

Huy M. Huynh, Trong T. Nguyen, and Tru Hoang Cao. 2013. Using Coreference and Surrounding Contexts for Entity Linking. In *2013 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, RIVF 2013, Hanoi, Vietnam, November 10-13, 2013,* 1–5. IEEE. (Cited on pages 82, 93).

Filip Ilievski. 2019. Identity of Long-tail Entities in Text. Vol. 43. Studies on the Semantic Web. IOS Press. (Cited on page 3).

Filip Ilievski, Pedro A. Szekely, and Bin Zhang. 2021. CSKG: the Commonsense Knowledge Graph. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings,* edited by Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, and Mehwish Alam, 12731:680–696. Lecture Notes in Computer Science. Springer. (Cited on pages 1, 17).

Anastasiia Iurshina, Jiaxin Pan, Rafika Boutalbi, and Steffen Staab. 2022. NILK: Entity Linking Dataset Targeting Nil-Linking Cases. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022,* edited by Mohammad Al Hasan and Li Xiong, 4069–4073. ACM. (Cited on page 94).

Monika Jain, Raghava Mutharaju, Ramakanth Kavuluru, and Kuldeep Singh. 2024. Revisiting Document-Level Relation Extraction with Context-Guided Link Prediction. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada,* edited by Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, 18327–18335. AAAI Press. (Cited on pages 9, 124, 132 sqq., 138).

Monika Jain, Kuldeep Singh, and Raghava Mutharaju. 2023. Reonto: A Neuro-Symbolic Approach for Biomedical Relation Extraction. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part IV,* edited by Danai Koutra, Claudia Plant, Manuel Gomez Rodriguez, Elena Baralis, and Francesco Bonchi, 14172:230–247. Lecture Notes in Computer Science. Springer. (Cited on pages 9, 132, 134, 138).

Ganesh Jawahar, Benoıt Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers,* edited by Anna Korhonen, David R. Traum, and Lluís Màrquez, 3651–3657. Association for Computational Linguistics. (Cited on page 23).

Heng Ji and Ralph Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA,* edited by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, 1148–1158. The Association for Computer Linguistics. (Cited on pages 30, 82, 93).

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Networks Learn. Syst.* 33 (2): 494–514. (Cited on pages 82, 108, 124).

Longquan Jiang, Junbo Huang, Cedric Möller, and Ricardo Usbeck. 2025. Ontology-Guided, Hybrid Prompt Learning for Generalization in Knowledge Graph Question Answering. **Best Student Paper Award**, *2025 19th International Conference on Semantic Computing (ICSC),* 28–35. (Cited on page 12).

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative Information Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, 4626–4643. Seattle, United States: Association for Computational Linguistics. (Cited on pages 38, 108, 113 sq., 120).

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing,* edited by Houda Bouamor, Juan Pino, and Kalika Bali, 1555–1574. Singapore: Association for Computational Linguistics. (Cited on pages 120 sq.).

Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd. Online manuscript released January 12, 2025. (Cited on page 22).

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR* abs/2001.08361. (Cited on page 5).

Nora Kassner, Fabio Petroni, Mikhail Plekhanov, Sebastian Riedel, and Nicola Cancedda. 2022. EDIN: An End-To-End Benchmark and Pipeline for Unknown Entity Discovery and Indexing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* edited by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, 8659–8673. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. (Cited on pages 8, 35, 87, 90, 93).

Kensho R&D group. 2020. Kensho Derived Wikimedia Dataset. kaggle. (Cited on pages 67, 71, 73).

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. A survey on recent advances in named entity recognition. *arXiv preprint arXiv:2401.10825,* (cited on page 31).

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. (Cited on page 26).

Barbara Kitchenham. 2004. Procedures for Performing Systematic Reviews. *Keele, UK, Keele University* 33 (2004): 1–26. (Cited on page 46).

Marcus Klang, Firas Dib, and Pierre Nugues. 2019. Overview of the Ugglan Entity Discovery and Linking System. *CoRR* abs/1903.05498. (Cited on page 62).

Marcus Klang and Pierre Nugues. 2014. Named Entity Disambiguation in a Question Answering System. In *TheFifth Swedish Language Technology Conference (SLTC2014).* (Cited on page 57).

Marcus Klang and Pierre Nugues. 2020. Hedwig: A Named Entity Linker [in English]. In *Proceedings of the Twelfth Language Resources and Evaluation Conference,* edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, 4501–4508. Marseille, France: European Language Resources Association. (Cited on pages 56, 62, 69 sqq.).

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 6982–6993. Online: Association for Computational Linguistics. (Cited on page 57).

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-Aware Distantly Supervised Relation Extraction with Linked Arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1891–1901. Doha, Qatar: Association for Computational Linguistics. (Cited on page 106).

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-To-End Neural Entity Linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning,* edited by Anna Korhonen and Ivan Titov, 519–529. Brussels, Belgium: Association for Computational Linguistics. (Cited on page 65).

Angelie Kraft and Ricardo Usbeck. 2022. The Lifecycle of "facts": A Survey of Social Bias in Knowledge Graphs. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers),* edited by Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, 639–652. Online only: Association for Computational Linguistics. (Cited on page 18).

Tanti Kristanti and Laurent Romary. 2020. DeLFT and Entity-fishing: Tools for CLEF HIPE 2020 Shared Task. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020,* edited by Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, vol. 2696. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 57).

Anand Kumar and Barry Smith. 2005. Oncology Ontology in the NCI Thesaurus. In *Artificial Intelligence in Medicine, 10th Conference on Artificial Intelligence in Medicine, AIME 2005, Aberdeen, UK, July 23-27, 2005, Proceedings,* edited by Silvia Miksch, Jim Hunter, and Elpida T. Keravnou, 3581:213–220. Lecture Notes in Computer Science. Springer. (Cited on page 131).

Kai Labusch and Clemens Neudecker. 2020. Named Entity Disambiguation and Linking on Historic Newspaper Ocr with Bert. *Working Notes of CLEF 2020 - Confer- ence and Labs of the Evaluation Forum,* (cited on pages 56, 65, 70 sq.).

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021,* edited by Zhi-Hua Zhou, 4483–4491. ijcai.org. (Cited on page 82).

Yuquan Lan, Dongxu Li, Yunqi Zhang, Hui Zhao, and Gang Zhao. 2023. Modeling Zero-Shot Relation Classification As a Multiple-Choice Problem. In *International Joint Conference on Neural Networks, IJCNN 2023, Gold Coast, Australia, June 18-23, 2023,* 1–8. IEEE. (Cited on pages 99, 101, 106, 127, 132, 135, 137).

Ora Lassila, J Hendler, and T Berners-Lee. 2001. The Semantic Web. *Scientific American* 284 (5): 34–43. (Cited on pages 14, 17).

Phong Le and Ivan Titov. 2018. Improving Entity Linking by Modeling Latent Relations between Mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* edited by Iryna Gurevych and Yusuke Miyao, 1595–1604. Melbourne, Australia: Association for Computational Linguistics. (Cited on pages 8, 83).

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web* 6 (2): 167–195. (Cited on pages 14, 42).

Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-Biggraph: A Large Scale Graph Embedding System. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019,* edited by Ameet Talwalkar, Virginia Smith, and Matei Zaharia. mlsys.org. (Cited on page 87).

Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. 2014. Mining of Massive Datasets, 2nd Ed. Cambridge University Press. (Cited on page 32).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-To-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 7871–7880. Online: Association for Computational Linguistics. (Cited on pages 24 sq., 114, 120).

Xueling Lin and Lei Chen. 2019. Canonicalization of Open Knowledge Bases with Side Information from the Source Text. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019,* 950–961. IEEE. (Cited on page 73).

Xueling Lin, Lei Chen, and Chaorui Zhang. 2021. TENET: Joint Entity and Relation Linking with Coherence Relaxation. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021,* edited by Guoliang Li, Zhanhuai Li, Stratos Idreos, and Divesh Srivastava, 1142–1155. ACM. (Cited on page 9).

Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. 2020. Kbpearl: A Knowledge Base Population System Supported by Joint Entity and Relation Linking. *Proc. VLDB Endow.* 13 (7): 1035–1049. (Cited on pages 9, 56, 63, 70 sq., 73).

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural Relation Extraction with Multi-Lingual Attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* edited by Regina Barzilay and Min-Yen Kan, 34–43. Vancouver, Canada: Association for Computational Linguistics. (Cited on page 50).

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design Challenges for Entity Linking. Edited by Michael Collins and Lillian Lee. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 3:315–328. (Cited on page 78).

Yang Liu, Kang Liu, Liheng Xu, and Jun Zhao. 2014. Exploring Fine-Grained Entity Type Constraints for Distantly Supervised Relation Extraction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,* edited by Junichi Tsujii and Jan Hajic, 2107–2116. Dublin, Ireland: Dublin City University / Association for Computational Linguistics. (Cited on page 106).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692. (Cited on pages 24, 59, 84).

Yue Liu, Tongtao Zhang, Zhicheng Liang, Heng Ji, and Deborah L. McGuinness. 2018. Seq2rdf: An End-To-End Application for Deriving Triples from Natural Language Text. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018,* edited by Marieke van Erp, Medha Atre, Vanessa López, Kavitha Srinivas, and Carolina Fortuna, vol. 2180. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 120).

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* edited by Anna Korhonen, David Traum, and Lluís Màrquez, 3449–3460. Florence, Italy: Association for Computational Linguistics. (Cited on pages 8, 32, 93, 119).

Federico López, Benjamin Heinzerling, and Michael Strube. 2019. Fine-Grained Entity Typing in Hyperbolic Space. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019,* edited by Isabelle Augenstein, Spandana Gella, Sebastian Ruder, Katharina Kann, Burcu Can, Johannes Welbl, Alexis Conneau, Xiang Ren, and Marek Rei, 169–180. Association for Computational Linguistics. (Cited on page 143).

Bo Lv, Xin Liu, Shaojie Dai, Nayu Liu, Fan Yang, Ping Luo, and Yue Yu. 2023. DSP: Discriminative Soft Prompts for Zero-Shot Entity and Relation Extraction. In *Findings of the Association for Computational Linguistics: ACL 2023,* edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 5491–5505. Toronto, Canada: Association for Computational Linguistics. (Cited on pages 101, 105, 132, 135, 137).

Youmi Ma, An Wang, and Naoaki Okazaki. 2023. DREEAM: Guiding Attention with Evidence for Improving Document-Level Relation Extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics,* edited by Andreas Vlachos and Isabelle Augenstein, 1971–1983. Dubrovnik, Croatia: Association for Computational Linguistics. (Cited on pages 108, 132 sqq.).

Chengcheng Mai, Yuxiang Wang, Ziyu Gong, Hanxiang Wang, and Yihua Huang. 2025. KnowRA: Knowledge Retrieval Augmented Method for Document-level Relation Extraction with Comprehensive Reasoning Abilities. *CoRR* abs/2501.00571. arXiv: 2501.00571. (Cited on pages 9, 144).

Magnus Manske. 2020. Wikidata Stats. (Cited on page 53).

Sina Menzel, Hannes Schnaitter, Josefine Zinck, Vivien Petras, Clemens Neudecker, Kai Labusch, Elena Leitner, and Georg Rehm. 2021. Named Entity Linking mit Wikidata und GND – Das Potenzial handkuratierter und strukturierter Datenquellen für die semantische Anreicherung von Volltexten. In *Qualität in der Inhaltserschließung,* edited by Michael Franke-Maier, Anna Kasprzik, Andreas Ledl, and Hans Schürmann, 229–258. De Gruyter. (Cited on page 95).

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. 2019. KnowledgeNet: A Benchmark Dataset for Knowledge Base Population. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 749–758. Hong Kong, China: Association for Computational Linguistics. (Cited on pages 71, 73).

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings,* edited by Yoshua Bengio and Yann LeCun. (Cited on page 8).

Makoto Miwa and Mohit Bansal. 2016. End-To-End Relation Extraction Using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* edited by Katrin Erk and Noah A. Smith, 1105–1116. Berlin, Germany: Association for Computational Linguistics. (Cited on pages 105, 120, 137).

Cedric Möller. 2022. Knowledge Graph Population with Out-of-KG Entities. In *The Semantic Web: ESWC 2022 Satellite Events - Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings,* edited by Paul Groth, Anisa Rula, Jodi Schneider, Ilaria Tiddi, Elena Simperl, Panos Alexopoulos, Rinke Hoekstra, Mehwish Alam, Anastasia Dimou, and Minna Tamper, 13384:199–214. Lecture Notes in Computer Science. **Runner-up**, **Best Doctoral Consortium Paper Award**. Springer. (Cited on page 11).

Cedric Möller, Jens Lehmann, and Ricardo Usbeck. 2022. Survey on English Entity Linking on Wikidata: Datasets and Approaches. *Semantic Web* 13 (6): 925–966. (Cited on pages 10, 41, 119).

Cedric Möller and Ricardo Usbeck. 2024a. DISCIE-Discriminative Closed Information Extraction. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II,* edited by Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan, 15232:23–40. Lecture Notes in Computer Science. Springer. (Cited on pages 10, 107).

Cedric Möller and Ricardo Usbeck. 2024b. Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering Using Only Knowledge Graphs. In *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands,* edited by Angelo A. Salatino, Mehwish Alam, Femke Ongenae, Sahar Vahdati, Anna Lisa Gentile, Tassilo Pellegrini, and Shufan Jiang, 60:88–105. Studies on the Semantic Web. IOS Press. (Cited on pages 10, 81).

Cedric Möller and Ricardo Usbeck. 2024c. Incorporating Type Information into Zero-Shot Relation Extraction. In *Joint proceedings of the 3rd International workshop on knowledge graph generation from text (TEXT2KG) and Data Quality meets Machine Learning and Knowledge Graphs (DQMLKG) co-located with the Extended Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 26-30, 2024,* edited by Sanju Tiwari, Nandana Mihindukulasooriya, Francesco Osborne, Dimitris Kontokostas, Jennifer D'Souza, Mayank Kejriwal, Maria Angela Pellegrino, Anisa Rula, José Emilio Labra Gayo, Michael Cochez, and Mehwish Alam, 3747:10. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 10, 97, 138).

Cedric Möller and Ricardo Usbeck. 2025. Analyzing the Influence of Knowledge Graph Information on Relation Extraction. In *The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I,* edited by Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena, 15718:460–480. Lecture Notes in Computer Science. **Best Student Paper Award**. Springer. (Cited on pages 10, 123).

Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. 2011. Cross-Lingual Cross-Document Coreference with Entity Linking. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011.* NIST. (Cited on pages 82, 93 sq.).

Nafise Sadat Moosavi and Michael Strube. 2016. Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-Based Entity Aware Metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* edited by Katrin Erk and Noah A. Smith, 632–642. Berlin, Germany: Association for Computational Linguistics. (Cited on page 90).

Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2017. MAG: A Multilingual, Knowledge-Base Agnostic and Deterministic Entity Linking Approach. In *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017,* edited by óscar Corcho, Krzysztof Janowicz, Giuseppe Rizzo, Ilaria Tiddi, and Daniel Garijo, 9:1–9:8. ACM. (Cited on pages 8, 53).

Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2018. Entity Linking in 40 Languages Using MAG. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers,* edited by Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z. Pan, and Mehwish Alam, 11155:176–181. Lecture Notes in Computer Science. Springer. (Cited on page 56).

Isaiah Onando Mulang, Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. 2020. Evaluating the Impact of Knowledge Graph Context on Entity Disambiguation Models. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020,* edited by Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, 2157–2160. ACM. (Cited on pages 8, 56, 59, 64, 67, 69 sqq.).

Isaiah Onando Mulang, Kuldeep Singh, Akhilesh Vyas, Saeedeh Shekarpour, Maria-Esther Vidal, and Sören Auer. 2020. Encoding Knowledge Graph Entity Aliases in Attentive Neural Network for Wikidata Entity Linking. In *Web Information Systems Engineering - WISE 2020 - 21st International Conference, Amsterdam, The Netherlands, October 20-24, 2020, Proceedings, Part I,* edited by Zhisheng Huang, Wouter Beek, Hua Wang, Rui Zhou, and Yanchun Zhang, 12342:328–342. Lecture Notes in Computer Science. Springer. (Cited on pages 53, 55 sq., 62, 69 sqq.).

David Nadeau and Satoshi Sekine. 2007. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes* 30 (1): 3–26. (Cited on page 16).

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2022. Named Entity Recognition and Relation Extraction: State-Of-The-Art. *ACM Comput. Surv.* 54 (1): 20:1–20:39. (Cited on pages 9, 31, 35).

Thien Huu Nguyen and Ralph Grishman. 2015. Relation Extraction: Perspective from Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing,* edited by Phil Blunsom, Shay Cohen, Paramveer Dhillon, and Percy Liang, 39–48. Denver, Colorado: Association for Computational Linguistics. (Cited on page 120).

Jian Ni and Radu Florian. 2019. Neural Cross-Lingual Relation Extraction Based on Bilingual Word Embedding Mapping. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 399–409. Hong Kong, China: Association for Computational Linguistics. (Cited on page 120).

Jian Ni, Gaetano Rossiello, Alfio Gliozzo, and Radu Florian. 2022. A Generative Model for Relation Extraction and Classification. *CoRR* abs/2202.13229. (Cited on pages 105, 120, 137).

Georgiana Nicolaie and Simona-Vasilica Oprea. 2025. Recent Trends and Insights in Semantic Web and Ontology-Driven Knowledge Representation across Disciplines Using Topic Modeling. *Electronics* 14 (7). (Cited on page 14).

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics,* edited by Emily M. Bender, Leon Derczynski, and Pierre Isabelle, 3866–3878. Santa Fe, New Mexico, USA: Association for Computational Linguistics. (Cited on page 38).

Kristian Noullet, Rico Mix, and Michael Färber. 2020. KORE 50^{DYWC}: An Evaluation Data Set for Entity Linking Based on Dbpedia, Yago, Wikidata, and Crunchbase. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020,* edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, 2389–2395. European Language Resources Association. (Cited on pages 71, 73).

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-Shot Relation Classification As Textual Entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER),* edited by James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal, 72–78. Brussels, Belgium: Association for Computational Linguistics. (Cited on pages 106, 137).

Italo L Oliveira, Renato Fileto, René Speck, Luís PF Garcia, Diego Moussallem, and Jens Lehmann. 2020. Towards Holistic Entity Linking: Survey and Directions. *Information Systems,* 101624. (Cited on pages 77 sq.).

OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774. (Cited on page 120).

Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* 22 (10): 1345–1359. (Cited on page 24).

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured Prediction As Translation between Augmented Natural Languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* OpenReview.net. (Cited on page 120).

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph Retrieval-Augmented Generation: A Survey. *CoRR* abs/2408.08921. (Cited on page 144).

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL,* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. ACL. (Cited on pages 8, 62).

Drew Perkins. 2020. Separating the Signal from the Noise: Predicting the Correct Entities in Named-Entity Linking. Master's thesis, Uppsala University. (Cited on pages 56, 59, 67, 70 sq.).

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),* edited by Marilyn Walker, Heng Ji, and Amanda Stent, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics. (Cited on page 59).

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: A Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, 2523–2544. Online: Association for Computational Linguistics. (Cited on page 77).

Riccardo Pozzi, Federico Moiraghi, Fausto Lodi, and Matteo Palmonari. 2022. Evaluation of Incremental Entity Extraction with Background Knowledge and Entity Linking. *The 11th International Joint Conference on Knowledge Graphs, October 27-29, 2022, Hangzhou, China,* (cited on page 94).

Thin Prabhong, Natthawut Kertkeidkachorn, and Areerat Trongratsameethong. 2024. KGC-RAG: Knowledge Graph Construction from Large Language Model Using Retrieval-Augmented Generation. In *Joint proceedings of the 2nd workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM 2024) and the 3rd challenge on Language Models for Knowledge Base Construction (LM-KBC 2024) co-located with the 23nd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 12, 2024,* edited by Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jeff Z. Pan, Tuan-Phong Nguyen, and Bohui Zhang, vol. 3853. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on pages 2, 144).

Vera Provatorova, Svitlana Vakulenko, Evangelos Kanoulas, Koen Dercksen, and Johannes M van Hulst. 2020. Named Entity Recognition and Linking on Historical Newspapers: Uva.ilps & Rel at Clef Hipe 2020. *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum,* (cited on pages 56, 66, 71).

Kunxun Qi, Jianfeng Du, and Hai Wan. 2024. End-To-End Learning of Logical Rules for Enhancing Document-Level Relation Extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024,* edited by Lun-Wei Ku, Andre Martins, and Vivek Srikumar, 7247–7263. Association for Computational Linguistics. (Cited on pages 129, 132 sqq., 137).

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving Language Understanding by Generative Pre-Training, (cited on pages 24 sqq.).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-To-Text Transformer. *J. Mach. Learn. Res.* 21:140:1–140:67. (Cited on pages 25, 120, 132, 134).

Mahdi Rahimi and Mihai Surdeanu. 2023. Improving Zero-Shot Relation Classification Via Automatically-Acquired Entailment Templates. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023),* edited by Burcu Can, Maximilian Mozes, Samuel Cahyawijaya, Naomi Saphra, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Chen Zhao, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefenstette, and Lena Voita, 187–195. Toronto, Canada: Association for Computational Linguistics. (Cited on pages 106, 137).

Jonathan Raiman. 2022. Deeptype 2: Superhuman Entity Linking, All You Need Is Type Interactions. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022,* 8028–8035. AAAI Press. (Cited on pages 8, 119, 143).

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual Entity Linking by Neural Type System Evolution. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018,* edited by Sheila A. McIlraith and Kilian Q. Weinberger, 5406–5413. AAAI Press. (Cited on pages 56, 58, 67, 69 sqq., 119, 143).

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* edited by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, 1375–1384. Portland, Oregon, USA: Association for Computational Linguistics. (Cited on page 49).

Daniel Ringler and Heiko Paulheim. 2017. One Knowledge Graph to Rule Them All? Analyzing the Differences between Dbpedia, Yago, Wikidata & Co. In *KI 2017: Advances in Artificial Intelligence - 40th Annual German Conference on AI, Dortmund, Germany, September 25-29, 2017, Proceedings,* edited by Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm, 10505:366–372. Lecture Notes in Computer Science. Springer. (Cited on page 42).

Petar Ristoski, Zhizhong Lin, and Qunzhi Zhou. 2021. KG-ZESHEL: Knowledge Graph-Enhanced Zero-Shot Entity Linking. In *K-CAP '21: Knowledge Capture Conference, Virtual Event, USA, December 2-3, 2021,* edited by Anna Lisa Gentile and Rafael Gonçalves, 49–56. ACM. (Cited on page 94).

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings,* edited by Yoshua Bengio and Yann LeCun. (Cited on pages 101, 132, 135).

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. GERBIL - Benchmarking Named Entity Recognition and Linking Consistently. *Semantic Web* 9 (5): 605–625. (Cited on page 77).

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2018. Voxel: A Benchmark Dataset for Multilingual Entity Linking. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II,* edited by Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, 11137:170–186. Lecture Notes in Computer Science. Springer. (Cited on page 76).

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2020. Fine-Grained Entity Linking. *Journal of Web Semantics,* 100600. (Cited on page 48).

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020,* edited by Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, 1885–1896. ACM / IW3C2. (Cited on page 54).

Dongyu Ru, Changzhi Sun, Jiangtao Feng, Lin Qiu, Hao Zhou, Weinan Zhang, Yong Yu, and Lei Li. 2021. Learning Logic Rules for Document-Level Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 1239–1250. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on pages 129, 132 sq., 137).

Stuart Russell and Peter Norvig. 2020. Artificial Intelligence: A Modern Approach (4th Edition). Pearson. (Cited on pages 18, 22).

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 1199–1212. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on pages 106, 137).

Ahmad Sakor, Isaiah Onando Mulang', Kuldeep Singh, Saeedeh Shekarpour, Maria-Esther Vidal, Jens Lehmann, and Sören Auer. 2019. Old is Gold: Linguistic Driven Approach for Entity and Relation Linking of Short Text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers),* edited by Jill Burstein, Christy Doran, and Thamar Solorio, 2336–2346. Association for Computational Linguistics. (Cited on pages 9, 62).

Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An Entity and Relation Linking Tool Over Wikidata. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020,* edited by Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, 3141–3148. ACM. (Cited on pages 9, 56, 62, 70 sq.).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108. arXiv: 1910.01108. (Cited on page 24).

Veronica Santos, Daniel Schwabe, and Sérgio Lifschitz. 2024. Can You Trust Wikidata? *Semantic Web* 15 (6): 2271–2292. (Cited on page 18).

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15,* 593–607. Springer. (Cited on page 28).

Edward W Schneider. 1973. Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis. (Cited on page 13).

Özge Sevgili, Alexander Panchenko, and Chris Biemann. 2019. Improving Neural Entity Disambiguation with Graph Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop,* edited by Fernando Alva-Manchego, Eunsol Choi, and Daniel Khashabi, 315–322. Florence, Italy: Association for Computational Linguistics. (Cited on pages 8, 31, 33, 77 sq., 94).

Hassan Shavarani and Anoop Sarkar. 2023. SpEL: Structured Prediction for Entity Linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing,* edited by Houda Bouamor, Juan Pino, and Kalika Bali, 11123–11137. Singapore: Association for Computational Linguistics. (Cited on page 108).

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Trans. Knowl. Data Eng.* 27 (2): 443–460. (Cited on pages 8, 78).

Kartik Shenoy, Filip Ilievski, Daniel Garijo, Daniel Schwabe, and Pedro A. Szekely. 2022. A Study of the Quality of Wikidata. *J. Web Semant.* 72:100679. (Cited on page 18).

Jiyun Shi, Zhimeng Yuan, Wenxuan Guo, Chen Ma, Jiehao Chen, and Meihui Zhang. 2023. Knowledge-graph-enabled biomedical entity linking: a survey. *World Wide Web (WWW)* 26 (5): 2593–2622. (Cited on page 142).

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* edited by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, 793–803. Portland, Oregon, USA: Association for Computational Linguistics. (Cited on page 94).

Amit Singhal. 2012. Introducing the Knowledge Graph: Things, Not Strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed 2022-03-29. (Cited on pages 14, 82).

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA,* edited by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, 4077–4087. (Cited on page 143).

Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* edited by Martha Palmer, Rebecca Hwa, and Sebastian Riedel, 1784–1789. Copenhagen, Denmark: Association for Computational Linguistics. (Cited on pages 50, 100, 132, 134).

Daniil Sorokin and Iryna Gurevych. 2018. Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics,* edited by Malvina Nissim, Jonathan Berant, and Alessandro Lenci, 65–75. New Orleans, Louisiana: Association for Computational Linguistics. (Cited on pages 56, 61, 70 sq.).

Alessandro Sperduti and Antonina Starita. 1997. Supervised Neural Networks for the Classification of Structures. *IEEE Trans. Neural Networks* 8 (3): 714–735. (Cited on page 57).

Andreas Spitz, Johanna Geiß, and Michael Gertz. 2016. So Far Away and yet so Close: Augmenting Toponym Disambiguation and Similarity with Text-Based Networks. In *Proceedings of the Third International ACM SIGMOD Workshop on Managing and Mining Enriched Geo-Spatial Data, GeoRich@SIGMOD 2016, San Francisco, California, USA, June 26 - July 1, 2016,* edited by Andreas Züfle, Benjamin Adams, and Dingming Wu, 2:1–2:6. ACM. (Cited on page 47).

Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. 2024. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024,* edited by Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, 131–140. ACM. (Cited on page 5).

Dianbo Sui, Chenhao Wang, Yubo Chen, Kang Liu, Jun Zhao, and Wei Bi. 2021. Set Generation Networks for End-To-End Knowledge Base Population. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 9650–9660. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on pages 114, 120).

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net. (Cited on page 27).

Suzanne Tamang, Zheng Chen, and Heng Ji. 2012. CUNY BLENDER TAC-KBP2012 Entity Linking System and Slot Filling Validation System. In *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012.* NIST. (Cited on pages 82, 93).

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Document-Level Relation Extraction with Adaptive Focal Loss and Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2022,* edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 1672–1681. Dublin, Ireland: Association for Computational Linguistics. (Cited on pages 132 sq.).

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting DocRED - Addressing the False Negative Problem in Relation Extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* edited by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, 8472–8487. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. (Cited on page 130).

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian M. Suchanek. 2020. YAGO 4: A Reason-Able Knowledge Base. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings,* edited by Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, 12123:583–596. Lecture Notes in Computer Science. Springer. (Cited on pages 42, 50, 55, 59).

Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive Relation Prediction by Subgraph Reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event,* 119:9448–9457. Proceedings of Machine Learning Research. PMLR. (Cited on pages 26, 29, 70).

Avijit Thawani, Minda Hu, Erdong Hu, Husain Zafar, Naren Teja Divvala, Amandeep Singh, Ehsan Qasemi, Pedro A. Szekely, and Jay Pujara. 2019. Entity Linking to Knowledge Graphs to Infer Column Types and Properties. In *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 18th International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019,* edited by Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Kavitha Srinivas, Vasilis Efthymiou, and Jiaoyan Chen, 2553:25–32. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 57).

The Gene Ontology Consortium et al. 2023. The Gene Ontology knowledgebase in 2023. *Genetics* 224, no. 1 (March): iyad031. eprint: https://academic.oup.com/genetics/article-pdf/224/1/iyad031/59147104/iyad031.pdf. (Cited on page 1).

Yuanyuan Tian. 2022. The World of Graph Databases from an Industry Perspective. *SIGMOD Rec.* 51 (4): 60–67. (Cited on page 14).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971,* (cited on page 25).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023b. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971. arXiv: 2302.13971. (Cited on pages 121, 144).

Van-Hien Tran, Hiroki Ouchi, Hiroyuki Shindo, Yuji Matsumoto, and Taro Watanabe. 2023. Enhancing Semantic Correlation between Instances and Relations for Zero-Shot Relation Extraction. *Journal of Natural Language Processing* 30 (2): 304–329. (Cited on pages 101 sqq., 105, 132, 135, 137).

Van-Hien Tran, Hiroki Ouchi, Taro Watanabe, and Yuji Matsumoto. 2022. Improving Discriminative Learning for Zero-Shot Relation Extraction. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge,* edited by Rajarshi Das, Patrick Lewis, Sewon Min, June Thai, and Manzil Zaheer, 1–6. Dublin, Ireland and Online: Association for Computational Linguistics. (Cited on pages 101, 105, 132, 135, 137).

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural Relation Extraction for Knowledge Base Enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* edited by Anna Korhonen, David Traum, and Lluís Màrquez, 229–240. Florence, Italy: Association for Computational Linguistics. (Cited on pages 113, 120).

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016,* edited by Maria-Florina Balcan and Kilian Q. Weinberger, 48:2071–2080. JMLR Workshop and Conference Proceedings. JMLR.org. (Cited on page 27).

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. 2014. AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014),* edited by Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, 263:1113–1114. Frontiers in Artificial Intelligence and Applications. IOS Press. (Cited on page 56).

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Sören Auer, Daniel Gerber, and Andreas Both. 2014. Agdistis - Graph-Based Disambiguation of Named Entities Using Linked Data. In *13th International Semantic Web Conference.* (Cited on page 8).

Cheikh Brahim El Vaigh, Guillaume Le Noé-Bienvenu, Guillaume Gravier, and Pascale Sébillot. 2020. IRISA System for Entity Detection and Linking at CLEF HIPE 2020. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020,* edited by Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, vol. 2696. CEUR Workshop Proceedings. CEUR-WS.org. (Cited on page 57).

Theo van Veen, Juliette Lonij, and Willem Jan Faber. 2016. Linking Named Entities in Dutch Historical Newspapers. In *Metadata and Semantics Research - 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings,* edited by Emmanouel Garoufallou, Imma Subirats Coll, Armando Stellato, and Jane Greenberg, 672:205–210. Communications in Computer and Information Science. (Cited on page 57).

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving Distantly-Supervised Neural Relation Extraction Using Side Information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* edited by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, 1257–1266. Brussels, Belgium: Association for Computational Linguistics. (Cited on pages 132 sq., 138).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA,* edited by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, 5998–6008. (Cited on pages 8, 22 sqq., 59).

Severine Verlinden, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2021. Injecting Knowledge Base Information into End-To-End Joint Entity and Relation Extraction and Coreference Resolution. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021,* edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 1952–1957. Online: Association for Computational Linguistics. (Cited on pages 132 sq., 138).

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada,* edited by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, 2692–2700. (Cited on page 63).

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57 (10): 78–85. (Cited on pages 1, 5, 17, 42, 52 sq., 131).

Jize Wang, Xinyi Le, Xiaodi Peng, and Cailian Chen. 2023. Adaptive Hinge Balance Loss for Document-Level Relation Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023,* edited by Houda Bouamor, Juan Pino, and Kalika Bali, 3872–3878. Singapore: Association for Computational Linguistics. (Cited on pages 20 sq., 130).

Meihong Wang, Linling Qiu, and Xiaoli Wang. 2021. A Survey on Knowledge Graph Embeddings for Link Prediction. *Symmetry* 13 (3): 485. (Cited on page 124).

Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. 2019. Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019,* 7152–7159. AAAI Press. (Cited on page 70).

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-Trained Language Representation. Edited by Brian Roark and Ani Nenkova. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 9:176–194. (Cited on page 70).

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *Found. Trends Databases* 10 (2-4): 108–490. (Cited on page 59).

Wikimedia Foundation. 2020a. Index of /wikidatawiki/entities/. (Cited on pages 54 sq.).

Wikimedia Foundation. 2020b. Wikistats. (Cited on page 43).

Wikimedia Foundation. Wikidata Data Model - Entities. https://grafana.wikimedia.org/d/000000167/wikidata-datamodel. Accessed: 2025-07-15. (Cited on page 18).

Wikimedia Foundation. Wikidata Data Model - Statements. https://grafana.wikimedia.org/d/000000175/wikidata-datamodel-statements. Accessed: 2025-07-15. (Cited on page 18).

David S. Wishart, Yannick D. Feunang, An Chi Guo, Elvis J. Lo, Ana Marcu, Jason R. Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2018. Drugbank 5.0: A Major Update to the Drugbank Database for 2018. *Nucleic Acids Res.* 46 (Database-Issue): D1074–D1082. (Cited on page 17).

World Wide Web Consortium. 2014. RDF 1.1 Concepts and Abstract Syntax. World Wide Web Consortium (W3C) Recommendation. Accessed: 2025-02-01. (Cited on page 14).

World Wide Web Consortium. 2025. Standards. https://www.w3.org/standards/. Accessed: 2025-07-11. (Cited on page 14).

Chengmin Wu and Lei Chen. 2020. Utber: Utilizing Fine-Grained Entity Types to Relation Extraction with Distant Supervision. In *IEEE International Conference on Smart Data Services, SMDS 2020, Beijing, China, October 19-23, 2020,* 63–71. IEEE. (Cited on page 106).

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable Zero-Shot Entity Linking with Dense Entity Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020,* edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 6397–6407. Association for Computational Linguistics. (Cited on pages 8, 33, 56, 60, 70, 93, 111, 119).

Shanchan Wu and Yifan He. 2019. Enriching Pre-Trained Language Model with Entity Information for Relation Classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019,* edited by Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu, 2361–2364. ACM. (Cited on page 105).

Zhaohui Wu, Yang Song, and C. Lee Giles. 2016. Exploring Multiple Feature Spaces for Novel Entity Discovery. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA,* edited by Dale Schuurmans and Michael P. Wellman, 3073–3079. AAAI Press. (Cited on pages 82, 93).

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems,* (cited on page 57).

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (9): 2251–2265. (Cited on page 4).

Rui Xing, Jie Luo, and Tengwei Song. 2020. Biorel: Towards Large-Scale Biomedical Relation Extraction. *BMC Bioinform.* 21-S (16): 543. (Cited on page 130).

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021. Discriminative Reasoning for Document-Level Relation Extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021,* edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 1653–1663. Online: Association for Computational Linguistics. (Cited on pages 132 sq., 137).

Xi Yan, Cedric Möller, and Ricardo Usbeck. 2025. Biomedical Entity Linking with Triple-aware Pre-Training. In *Proceedings of the Third International Workshop on Semantic Technologies and Deep Learning Models for Scientific, Technical and Legal Data (SemTech4STLD 2025), co-located with the Extended Semantic Web Conference (ESWC 2025),* edited by Rima Dessí, Joy Jeenu, Danilo Dessì, Francesco Osborne, and Hidir Aras. Portoroz, Slovenia: CEUR-WS.org, June. (Cited on page 12).

Xi Yan, Aida Usmanova, Cedric Möller, and Patrick Westphal. 2025. Neuro-Symbolic Relation Extraction. In *Handbook on Neurosymbolic AI and Knowledge Graphs.* IOS Press, March. (Cited on page 12).

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings,* edited by Yoshua Bengio and Yann LeCun. (Cited on pages 27, 128).

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized Out-of-Distribution Detection: A Survey. *Int. J. Comput. Vis.* 132 (12): 5635–5662. (Cited on page 95).

Jung-Jin Yang. 2003. An Ontology-Based Intelligent Agent System for Semantic Search in Medicine. In *Intelligent Agents and Multi-Agent Systems, 6th Pacific Rim International Workshop on Multi-Agents, PRIMA 2003, Seoul, Korea, November 7-8, 2003, Proceedings,* edited by Jaeho Lee and Mike Barley, 2891:182–193. Lecture Notes in Computer Science. Springer. (Cited on page 131).

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning Dynamic Context Augmentation for Global Entity Linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 271–281. Hong Kong, China: Association for Computational Linguistics. (Cited on page 59).

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada,* edited by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, 5754–5764. (Cited on page 59).

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. CodRED: A Cross-Document Relation Extraction Dataset for Acquiring Knowledge in the Wild. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* edited by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, 4452–4472. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. (Cited on page 36).

Zi Ye, Yogan Jaya Kumar, Goh Ong Sing, Fengyan Song, and Junsong Wang. 2022. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. *IEEE Access* 10:75729–75741. (Cited on pages 26 sqq.).

Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings,* edited by Yoshua Bengio and Yann LeCun. (Cited on page 61).

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. DWIE: an Entity-Centric Dataset for Multi-Task Document-Level Information Extraction. *Inf. Process. Manag.* 58 (4): 102563. (Cited on page 130).

Klim Zaporojets, Lucie-Aimée Kaffee, Johannes Deleu, Thomas Demeester, Chris Develder, and Isabelle Augenstein. 2022. Tempel: Linking Dynamically Evolving and Newly Emerging Entities. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022,* edited by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. (Cited on page 94).

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification Via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,* edited by Junichi Tsujii and Jan Hajic, 2335–2344. Dublin, Ireland: Dublin City University / Association for Computational Linguistics. (Cited on pages 105, 120, 137).

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double Graph Based Reasoning for Document-Level Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 1630–1640. Online: Association for Computational Linguistics. (Cited on page 137).

Bowen Zhang and Harold Soh. 2024. Extract, Define, Canonicalize: An Llm-Based Framework for Knowledge Graph Construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024,* edited by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, 9820–9836. Association for Computational Linguistics. (Cited on pages 2, 144).

Lei Zhang, Tianxing Wu, Liang Xu, Meng Wang, Guilin Qi, and Harald Sack. 2019. Emerging Entity Discovery Using Web Sources. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding - 4th China Conference, CCKS 2019, Hangzhou, China, August 24-27, 2019, Revised Selected Papers,* edited by Xiaoyan Zhu, Bing Qin, Xiaodan Zhu, Ming Liu, and Longhua Qian, 1134:175–184. Communications in Computer and Information Science. Springer. (Cited on pages 82, 93).

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021. Document-Level Relation Extraction As Semantic Segmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021,* edited by Zhi-Hua Zhou, 3999–4006. ijcai.org. (Cited on pages 132 sq., 137).

Pengyu Zhang, Congfeng Cao, and Paul Groth. 2024. TIGER: Temporally Improved Graph Entity Linker. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024),* edited by Ulle Endriss, Francisco S. Melo, Kerstin Bach, Alberto José Bugarín Diz, Jose Maria Alonso-Moral, Senén Barro, and Fredrik Heintz, 392:3733–3740. Frontiers in Artificial Intelligence and Applications. IOS Press. (Cited on page 8).

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize Exposure Bias of Seq2Seq Models in Joint Entity and Relation Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020,* edited by Trevor Cohn, Yulan He, and Yang Liu, 236–246. Online: Association for Computational Linguistics. (Cited on page 120).

Sheng Zhang, Patrick Ng, Zhiguo Wang, and Bing Xiang. 2022. Reknow: Enhanced Knowledge for Joint Entity and Relation Extraction. *CoRR* abs/2206.05123. (Cited on page 120).

Zhaoyang Zhang, Hongtang Cao, Xiaoyu Wang, Yang Zhang, and Quan Z. Sheng. 2024. Ontology-Driven Archival Knowledge Graph Construction Leveraging Large Language Models. In *Australasian Conference on Information Systems, ACIS 2024, Canberra, Australia, December 4-6, 2024.* (Cited on pages 2, 144).

Jun Zhao, WenYu Zhan, Xin Zhao, Qi Zhang, Tao Gui, Zhongyu Wei, Junzhe Wang, Minlong Peng, and Mingming Sun. 2023. RE-Matching: A Fine-Grained Semantic Matching Method for Zero-Shot Relation Extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 6680–6691. Toronto, Canada: Association for Computational Linguistics. (Cited on pages 25, 101, 105, 132, 135, 137).

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A Survey of Large Language Models. *CoRR* abs/2303.18223. (Cited on page 2).

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A Comprehensive Survey on Relation Extraction: Recent Advances and New Frontiers. *ACM Comput. Surv.* 56 (11): 293:1–293:39. (Cited on pages 9, 36).

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2024. A Comprehensive Survey on Automatic Knowledge Graph Construction. *ACM Comput. Surv.* 56 (4): 94:1–94:62. (Cited on page 1).

Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, 50–61. Online: Association for Computational Linguistics. (Cited on pages 105, 120, 137).

Kang Zhou, Yuepei Li, Qing Wang, Qiao Qiao, and Qi Li. 2024. Gendecider: Integrating "none of the Candidates" Judgments in Zero-Shot Entity Linking Re-Ranking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024,* edited by Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, 239–245. Association for Computational Linguistics. (Cited on pages 8, 33).

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. 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 2022, Vienna, Austria, 23-29 July 2022,* edited by Luc De Raedt, 5694–5701. ijcai.org. (Cited on pages 37 sq.).

Shuyan Zhou, Shruti Rijhwani, John Wieting, Jaime Carbonell, and Graham Neubig. 2020. Improving Candidate Generation for Low-Resource Cross-Lingual Entity Linking. Edited by Mark Johnson, Brian Roark, and Ani Nenkova. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 8:109–124. (Cited on page 56).

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021,* 14612–14620. AAAI Press. (Cited on pages 125, 132 sq., 137).

Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. 2023. Learn to Not Link: Exploring NIL Prediction in Entity Linking. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023,* edited by Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, 10846–10860. Association for Computational Linguistics. (Cited on page 8).

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph Neural Networks with Generated Parameters for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* edited by Anna Korhonen, David Traum, and Lluís Màrquez, 1331–1339. Florence, Italy: Association for Computational Linguistics. (Cited on page 132).

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual,* edited by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, 29476–29490. (Cited on pages 28, 124, 127, 134, 143).

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Doser - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings. In *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings,* edited by Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, 9678:182–198. Lecture Notes in Computer Science. Springer. (Cited on page 56).