RENÉ SPECK

KNOWLEDGE EXTRACTION FOR THE DATA WEB

Data Science Group, Department of Computer Science

Doctoral Dissertation

KNOWLEDGE EXTRACTION FOR THE DATA WEB

A dissertation presented by

RENÉ SPECK

to the

Faculty for Computer Science, Electrical Engineering and Mathematics

of

Paderborn University

in partial fulfillment of the requirements for the degree of

Dr. rer. nat.

July 12, 2024

REVIEWERS

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

Prof. Dr. Thomas Riechert
Faculty of Computer Science and Media
Leipzig University of Applied Sciences

SUPERVISOR

Prof. Dr. Axel-Cyrille Ngonga Ngomo

BIBLIOGRAPHIC DATA

5 parts, 12 chapters, 32 figures, 38 tables, 17 listings, 2 appendix chapters, 205 pages

ABSTRACT

Knowledge extraction is an essential component in numerous applications across various fields of research, such as fact-checking, knowledge graph creation and population, and knowledge graph question answering. In its application areas, knowledge extraction often plays a crucial role by permitting the extraction of meaningful insights from large volumes of unstructured data. The obtained knowledge allows reasonable and intelligent decision-making. Our contributions enhance the core tasks of knowledge extraction, thereby improving the overall quality in various research areas build on these core tasks. Our first contribution is based on ensemble learning for named entity recognition and achieves a 40% error rate reduction on this specific core task. We provide our results with Fox, a framework that serializes a knowledge graph with well-defined machine-processable standard as its output. We propose Cetus, a pattern-based entity type extraction approach to populate knowledge graphs. Our approach achieves a higher F-measure than other approaches on the entity type extraction and linking task. We present our survey on holistic entity linking and our reference framework for this core task. We introduce Ocelot, a distant supervised closed relation extraction approach based on distributional semantics and a tree generalization. Our approach harvests generalized dependency tree patterns of high quality and extracts relations from unstructured text with its generalized trees of higher precision than two state-of-the-art systems for this core task. With our Scms approach, we facilitate semantic data integration with our Fox framework in content management systems to semantify their content and to support efficient decision-making. We report on our participation in the Semantic Web challenge with our approach, Leopard, which was designed for the prediction and validation of attributes, as well as for the population of knowledge graphs. Thus, our approach improves accuracy and completeness of knowledge graphs. Leopard combines a variety of diverse knowledge and text extraction methods, leveraging sources from both the multilingual Document Web and the multilingual Data Web, while incorporating precision ranking techniques. Lastly, we outline our participation in the Open Knowledge Extraction Challenge, elaborating on how we utilized our proposed approaches during the challenge. Overall, we believe that our contributions constitute a significant step forward in the field of knowledge extraction and knowledge graph creation. Our approaches not only enhance the quality and efficiency in this field but also pave the way for new possibilities in data understanding and integration.

ZUSAMMENFASSUNG

Die Wissensextraktion ist ein wesentlicher Bestandteil in zahlreichen Anwendungen verschiedenster Forschungsfelder, wie beispielsweise die Prüfung von Fakten, das Erstellen und Befüllen von Wissensgraphen sowie Frage-Antwort-Systeme basierend auf Wissensgraphen. Die Wissensextraktion spielt in ihren Anwendungsgebieten oft eine entscheidende Rolle, indem sie die Extraktion bedeutungsvoller Erkenntnisse aus großen Mengen unstrukturierter Daten ermöglicht. Das gewonnene Wissen unterstützt bei fundierten und intelligenten Entscheidungsfindungen. Unsere Beiträge verbessern die Qualität der Wissensextraktion in ihren Kernaufgaben, dadurch erhöhen wir die Gesamtqualität in verschiedenen Forschungsbereichen. Unser erster Beitrag basiert auf Ensemblemethoden für die Erkennung von Entitäten in Text und erreicht eine Reduzierung der Fehlerrate um 40% für diese spezifische Kernaufgabe. Wir stellen unsere Ergebnisse mit Fox zur Verfügung, ein Framework, das einen Wissensgraphen mit maschinenverarbeitbaren Standards als Ausgabe serialisiert. Wir schlagen Cetus vor, ein auf Muster basierender Ansatz für die Extraktion von Entitätstypen und für das Befüllen von Wissensgraphen. Unser Ansatz erreicht eine höhere F-measure als andere Ansätze für die Aufgabe Entitätstypen zu extrahieren und mit Wissensgraphen zu verknüpfen. Wir präsentieren unsere Untersuchungen zu holistischen Ansätzen für das verlinkten von Entitäten und schlagen unser Referenzframework für diese Kernaufgaben vor. Wir stellen Ocelot vor, ein Ansatz für die Extraktion von vordefinierten Relationen basierend auf distributionaler Semantik und einer Baumgeneralisierung. Wir kombinieren dabei überwachtes und unüberwachtes Lernen für diese Kernaufgabe. Unser Ansatz verallgemeinert Muster in Dependenzbäumen von hoher Qualität und extrahiert Relationen aus unstrukturiertem Text mit diesen Baumgeneralisierungen präziser als zwei andere Systeme, die dem Stand der Technik entsprechen. Mit unserem Scms Ansatz erleichtern wir die semantische Datenintegration in Content-Management-Systemen mit unserem Fox-Framework. Unser Ansatz unterstützt Inhalte zu semantisieren und unterstützt damit eine effizientere Entscheidungsfindung. Wir berichten über unsere Teilnahme an der Semantic Web Challenge mit unserem Ansatz Leopard, der für die Vorhersage und Validierung von Attributen sowie für die Befüllung von Wissensgraphen entwickelt wurde. Unser Ansatz unterstützt dabei die Genauigkeit und Vollständigkeit von Wissensgraphen zu verbessern. Leopard kombiniert eine Vielzahl von vielfältigen Wissens- und Textextraktionsmethoden und nutzt Quellen sowohl aus dem mehrsprachigen Web der Dokumente, als auch dem mehrsprachigen

Web der Daten, unter Einbeziehung von Rankingverfahren auf Präzisionsbasis. Abschließend skizzieren wir unsere Teilnahme an der Open Knowledge Extraction Challenge und erläutern, wie wir unsere vorgeschlagenen Ansätze während der Challenge genutzt haben. Wir sind insgesamt davon Überzeugung, dass unsere Beiträge einen bedeutenden Fortschritt im Bereich der Wissensextraktion und der Erstellung von Wissensgraphen darstellen. Unsere Ansätze verbessern nicht nur die Qualität und die Effizienz, sondern eröffnen auch neue Möglichkeiten für das Verständnis und die Integration von Daten.

PUBLICATIONS

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

AWARDS AND NOTABLE MENTIONS

- **Finalist of the Rich Context Competition 2019** for Rricha Jalota, Nikit Srivastava, Daniel Vollmers, **René Speck**, Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "Finding datasets in publications: The University of Paderborn approach." In: *Rich Search and Discovery for Research Datasets*. Ed. by Julia I. Lane, Ian Mulvany, and Paco Nathan. SAGE Publications Ltd, 2020, pp. 129–141
- **Second-best score in both ISWC Semantic Web Challenge 2017 tasks behind the IBMs system Socrates** for **René Speck** and Axel-Cyrille Ngonga Ngomo. "Leopard — A baseline approach to attribute prediction and validation for knowledge graph population." In: *Journal of Web Semantics* (2019)
- **Best Performing System at the Open Knowledge Extraction Challenge 2017 for Tasks 1 and 2** for **René Speck**, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. "Open Knowledge Extraction Challenge 2017." In: *Semantic Web Challenges: Fourth SemWebEval Challenge at ESWC 2017*. Communications in Computer and Information Science. Springer International Publishing, 2017
- **First prize at Task 2 of the Open Knowledge Extraction Challenge at ISWC 2015** for Michael Röder, Ricardo Usbeck, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "CETUS – A Baseline Approach to Type Extraction." In: *1st Open Knowledge Extraction Challenge at International Semantic Web Conference*. 2015

JOURNAL ARTICLES

1. ★ Italo L. Oliveira, Renato Fileto, **René Speck**, Luís P.F. Garcia, Diego Moussallem, and Jens Lehmann. "Towards holistic Entity Linking: Survey and directions." In: *Information Systems* 95 (2021), p. 101624

2. ★ **René Speck** and Axel-Cyrille Ngonga Ngomo. "Leopard — A baseline approach to attribute prediction and validation for knowledge graph population." In: *Journal of Web Semantics* (2019)

3. Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and **René Speck**. "DeFacto - Temporal and Multilingual Deep Fact Validation." In: *Web Semantics: Science, Services and Agents on the World Wide Web* (2015)

CONFERENCE ARTICLES

4. Jiayi Li, Sheetal Satheesh, Stefan Heindorf, Diego Moussallem, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "AutoCL: AutoML for Concept Learning." In: *The 2nd World Conference on eXplainable Artificial Intelligence (xAI-2024)*. 2024

5. **René Speck**, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. "Twitter Network Mimicking for Data Storage Benchmarking." In: *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. 2021, pp. 298–305

6. ★ **René Speck** and Axel-Cyrille Ngonga Ngomo. "On Extracting Relations using Distributional Semantics and a Tree Generalization." In: *Proceedings of The 21th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2018)*. 2018

7. Axel-Cyrille Ngonga Ngomo, Michael Röder, Diego Moussallem, Ricardo Usbeck, and **René Speck**. "BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking." In: *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*. Ed. by Emiel Krahmer, Albert Gatt, and Martijn Goudbeek. Association for Computational Linguistics, 2018, pp. 339–349

8. ★ **René Speck** and Axel-Cyrille Ngonga Ngomo. "Ensemble Learning of Named Entity Recognition Algorithms using Multilayer Perceptron for the Multilingual Web of Data." In: *K-CAP 2017: Knowledge Capture Conference*. ACM. 2017, p. 4

9. Vladimir Salin, Maria Slastihina, Ivan Ermilov, **René Speck**, Sören Auer, and Sergey Papshev. "Semantic Clustering of Website Based on Its Hypertext Structure." In: *Knowledge Engineering and Semantic Web*. Ed. by Pavel Klinov and Dmitry Mouromtsev. Cham: Springer International Publishing, 2015, pp. 182–194

10. Mofeed M. Hassan, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "Using Caching for Local Link Discovery on Large

Data Sets." English. In: *Engineering the Web in the Big Data Era*. Vol. 9114. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 344–354

11. Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, **René Speck**, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. "GERBIL – General Entity Annotation Benchmark Framework." In: *24th International Conference on World Wide Web*. 2015

12. ★ **René Speck** and Axel-Cyrille Ngonga Ngomo. "Ensemble Learning for Named Entity Recognition." In: *The Semantic Web – ISWC 2014*. Vol. 8796. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 519–534

13. Axel-Cyrille Ngonga Ngomo, Norman Heino, **René Speck**, and Prodromos Malakasiotis. "A tool suite for creating Question Answering benchmarks." In: *Proceedings of LREC*. 2014

14. **René Speck** and Axel-Cyrille Ngonga Ngomo. "On Caching for Local Graph Clustering Algorithms." In: *Australasian Conference on Artificial Intelligence*. 2013, pp. 56–67

15. ★ Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, **René Speck**, and Martin Kaltenböck. "SCMS - Semantifying Content Management Systems." In: *ISWC 2011*. 2011

BOOK CHAPTERS

16. Diego Moussallem, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "Generating Explanations in Natural Language from Knowledge Graphs." In: *Knowledge Graphs for eXplainable Artificial Intelligence*. Vol. 47. Studies on the Semantic Web. 2020, pp. 213–241

17. Rricha Jalota, Nikit Srivastava, Daniel Vollmers, **René Speck**, Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "Finding datasets in publications: The University of Paderborn approach." In: *Rich Search and Discovery for Research Datasets*. Ed. by Julia I. Lane, Ian Mulvany, and Paco Nathan. SAGE Publications Ltd, 2020, pp. 129–141

WORKSHOP ARTICLES, POSTERS & OTHERS

18. Diego Moussallem, Paramjot Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, Michael Röder,

René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. "A General Benchmarking Framework for Text Generation." In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Dublin, Ireland (Virtual): Association for Computational Linguistics, Dec. 2020, pp. 27–33

19. ★ **René Speck**, Michael Röder, Felix Conrads, Hyndavi Rebba, Catherine Camilla Romiyo, Gurudevi Salakki, Rutuja Suryawanshi, Danish Ahmed, Nikit Srivastava, Mohit Mahajan, and Axel-Cyrille Ngonga Ngomo. "Open Knowledge Extraction Challenge 2018." In: *Semantic Web Evaluation Challenge*. Springer International Publishing, 2018, pp. 39–51

20. ★ **René Speck**, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. "Open Knowledge Extraction Challenge 2017." In: *Semantic Web Challenges: Fourth SemWebEval Challenge at ESWC 2017*. Communications in Computer and Information Science. Springer International Publishing, 2017

21. **René Speck**, Diego Esteves, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. "DeFacto - A Multilingual Fact Validation Interface." In: *14th International Semantic Web Conference (ISWC 2015), 11-15 October 2015, Bethlehem, Pennsylvania, USA (Semantic Web Challenge Proceedings)*. Ed. by Sean Bechhofer and Kostis Kyzirakos. Semantic Web Challenge, International Semantic Web Conference 2015. 2015

22. ★ Michael Röder, Ricardo Usbeck, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "CETUS – A Baseline Approach to Type Extraction." In: *1st Open Knowledge Extraction Challenge at International Semantic Web Conference*. 2015

23. ★ **René Speck** and Axel-Cyrille Ngonga Ngomo. "Named Entity Recognition using FOX." In: *International Semantic Web Conference 2014 (ISWC2014), Demos & Posters*. 2014

24. Klaus Lyko, Konrad Höffner, **René Speck**, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. "SAIM—One Step Closer to Zero-Configuration Link Discovery." In: *Proc. of the Extended Semantic Web Conference Posters & Demos*. 2013

ACKNOWLEDGMENTS

I would like to express my profound appreciation to my supervisor and mentor Prof. Dr. Axel-Cyrille Ngonga Ngomo for his constant encouragement, support, and guidance throughout my research. His constructive comments and feedback to me and other colleagues within discussions have shaped my work invaluable.

My gratitude extends to all colleagues with whom I jointly collaborated with on papers and articles that led to this work, especially Dr. Michael Röder, Dr. Diego Moussallem and Prof. Dr. Ricardo Usbeck. Their conversations and assistance helped me stay motivated and focused on my work.

I thank my family for their deep love. Their understanding has been a source of strength for me during the challenging times of my research.

I am grateful to my love of life and my friends for their support and patience.

I am also grateful to the faculty members of the Department of Computer Science - Data Science Group (DICE) at Paderborn University, who have contributed their expertise and knowledge to my academic growth.

Finally, I would like to acknowledge the participants of my study, without whom my research would not have been possible. Their willingness to share their experiences and insights has enriched my work and contributed to its quality.

Thank you all for your contributions and support.

Parts of this work has been supported by 1. the H2020 project HOBBIT (GA no. 688227), 2. the EuroStars projects DIESEL (no. 01QE1512C) and QAMEL (project no. 01QE1549C), and 3. the BMWI projects GEISER (no. 01MD16014E) and OPAL (no. 19F2028A)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

Part I

INTRODUCTION

1 INTRODUCTION

1.1 MOTIVATION

The tremendous and almost incomprehensible amount of data produced by humans and provided on the Web each day continues to grow extensively at rapid rates [6]. Estimates suggest that this data will more than double[1,2] from 2023 to 2027. This astounding ever-growing amount of new daily data consists of apparently at least 80%[3] [28] that is not easily processable or quantifiable by machines [165, 181], i. e., unstructured data. This data includes diverse types of *text sources* [113], such as news, comments, and social media posts [207]. Currently, these text sources lack of well-defined machine-processable semantics that hinders their efficient use by applications [18, 20, 122, 142, 181, 195, 209, 240]. Manual annotation and linking with semantic meaning [195] of such vast amounts of text to circumvent this lack is expensive due to the prohibitive workforce needed [101] and the difficulty of obtaining high-quality and standardized results [20]. Consequently, desirable characteristics to transform text sources to well-defined machine-processable semantics are, for instance, minimal human effort (i. e., (semi-) automatically), scalability, and accuracy [198, 236, 240, 262].

In contrast, the objective of knowledge extraction is to transform a corpus of text documents into a structured output described using knowledge representation formalisms [165]. These formalisms can provide model-theoretic semantics[4] by relying on well-defined machine-processable standards [115] such as those recommended by the World Wide Web Consortium (W3C), for instance, the Web Ontology Language [170, 217] (OWL) and the RDF Schema [23] (RDFS). Thus, knowledge extraction plays a crucial role by enabling agents to extract insights from large volumes of data. Hence, allowing the extracted well-defined machine-processable semantics, for instance, integrated into explainable machine learning approaches to provide more meaningful explanations and trustworthy decisions [14, 148, 210, 211, 268, 285].

Commonly, knowledge extraction models its output in a graph-based data model, i. e., a Knowledge Graph (KG). KGs are often ap-

1 https://www.idc.com/getdoc.jsp?containerId=US50554523

2 https://aws.amazon.com/blogs/publicsector/aws-partners-help-public-sector-organizations-harness-the-power-of-data/

3 https://hbr.org/2022/02/why-becoming-a-data-driven-organization-is-so-hard

4 https://www.w3.org/TR/owl-semantics/direct.html

plied in scenarios that include integrating, managing, and extracting value from diverse data sources at large scales [113]. KGs constitute not only graph-based knowledge representations [62] but also provide an ever-evolving [113, 114] shared substrate of knowledge [204] in practice. Thus, a KG is a collection of interlinked descriptions of entities and provides knowledge about these entities and their relations usually in a standardized and machine-processable format. Today, using KGs, and thus semantic processing of graph data, is the norm rather than the exception [174]. Companies, such as Facebook [204], Google [255], and Microsoft [252] have created their own. KGs on the Web have been growing in number and size [10, 64, 73, 109, 114, 235, 236]. Publicly accessible interlinked KGs that rely on standards, such as OWL, that facilitate machine interpretability of Web content forming the Linked Open Data (LOD)-Cloud[5] [195]. In 2007, the LOD-Cloud with its linked data consisted of 12 datasets such as DBpedia and MusicBrainz (see Figure A.1). The growth of the datasets that need to be interlinked fosters the growth of the LOD-Cloud. Thus, the LOD-Cloud consisted of 570 datasets in 2014 and consisted of 1,255 datasets later in 2020 (see Figure A.2). Still, KGs are not exhaustive and ongoing effort is needed to enrich KGs with newly emerging information in unstructured data [122].

Thus, knowledge extraction from text sources is an essential component in many applications of various research fields, including, for instance, KG creation and population [248, 263], fact-checking [88, 256], and KG question answering [54, 275, 306]. Opting to create and utilize a KG unlocks a variety of methods that can be employed to extract valuable insights from diverse data sources [113]. An example of one of these methods is semantic data integration [39, 198] where the constructed KG models the structured semantics of the data and thus fosters knowledge integration and alignment across heterogeneous sources.

However, the principal techniques by which KGs can be created or enriched are: human collaboration, text sources, markup sources, structured sources and schema/ontology creation [113, 114]. Human collaboration is expensive, a more efficient way of KG creation is the use of structured or semi-structured sources [107]. Thus, creating these sources (semi-) automatically with high precision and recall by extracting information from text sources supports an efficient way of KG creation but is a non-trivial challenge [113].

The four core extraction tasks [113, 122] for KG creation and population are: (i) *Preprocessing*, (ii) *Named Entity Recognition (NER)*, (iii) *Entity Linking (EL)*, and (iv) *Relation Extraction (RE)*. The NER task serves as a preprocessing step [207] of EL to identify entities in text sources and link these entities to a KG [182, 185]. RE is another core extraction task [165, 193] to model the interlinking of entities in a KG. Appli-

5 https://lod-cloud.net/#about

cations that rely on components for knowledge extraction demand highly accurate results from the extraction process [240]. To meet this demand, precise extraction processes for the aforementioned core extraction tasks are necessary.

For example, one application area of importance is Knowledge Graph Question Answering (KGQA) that aims at answering questions by accessing KGs using a query language. This is generally accomplished by converting user questions to corresponding SPARQL Protocol And RDF Query Language [37] (SPARQL) queries, whose result sets are the answers to the questions [54]. KGQA systems are typically composed of two stages: (i) the query analyzer, and (ii) the retrieval stage [112, 254]. Common techniques within the query analyzer are NER together with EL and RE [300]. From the *QALD* dataset [276], we consider the following example question: "*Is Michelle Obama the wife of Barack Obama?*". A common way to answer this question with a KGQA system is to produce a semantic representation of this question within the query analyzer. Listing A.1 exemplifies a possible semantic representation in an Resource Description Framework [45, 136] (RDF) graph serialization of the aforementioned example question. In the retrieval stage, a semantic representation is commonly converted into a standard language for querying RDF graphs, i. e., SPARQL query, (e. g., Listing A.2) to request another source, for instance, a KG, for further analytics and to create an answer to the question.

Hence, the key steps to structuring knowledge from text sources for further analysis [140, 181, 231] include the spotting of token spans that constitute entity mentions (e. g., "*Michelle Obama*" and "*Barack Obama*") along with assigning types (e. g., `PERSON`) to these mentions with the NER task. The steps include the linking of identified mentions to entities in KGs (e. g., `dbr:Michelle_Obama`[6]) by applying EL. Furthermore, extracting relations (e. g., `dbo:spouse`) between entities with RE.

However, collections of semantically-typed relational patterns as provided by RE approaches, such as Patty [192] and Boa [89], are frequently applied in the query analyzer of KGQA systems to match word patterns and link those to a KG for modeling the semantics of a question [58, 60, 275]. In these collections, one pattern matches several relations in many cases. For instance, in the collections of Patty and Boa, the pattern "*was born*" corresponds, respectively, to six and to three relations of the KG DBpedia. These mismatches can result in semantic drift, i. e., a significant number of false positives (*FP*s) [254]. Another limitation of these tools, e. g., Patty and Boa, is their inability to extract relations that are not enclosed by entity mentions, e. g., in the sentence "*Michelle and Barack are married.*" the verb expresses a relation, i. e., `dbo:spouse`, but is not surrounded by the entity mentions "*Michelle*" and "*Barack*" in the sentence.

6 `https://dbpedia.org/page/Michelle_Obama`

Standard processing pipelines that consume text sources for knowledge extraction miss the opportunity to gain semantic insights from emerging entities that are novel to the underlying KG. That is, these pipelines recognize entities based on linguistic models and link them to a KG or ignore them in case they are emerging entities. At the time of writing the thesis linking novel entities has only been the concern of a few approaches [111, 277].

Surveys about EL present a good overview of existing approaches, datasets, benchmarks [248, 291], and EL approaches for a specific type of text, such as microblog posts [51]. However, at the time of writing the thesis, these surveys do not consider in detail emerging strategies that influence new EL approaches and can be regarded as ways to consider several facets of the EL task concomitantly in a more holistic EL processes.

The NER tools that have resulted from decades of research implement a diversity of algorithms relying on a large number of heterogeneous formalisms. Consequently, these algorithms have diverse strengths and weaknesses. For instance, a NER tool may perform significantly better for a specific entity type, e. g., persons, whereas this tool may perform poorer for another entity type. Several services and frameworks that consume text sources into generate semi-structured or even structured data at the time of writing the thesis rely on solely one of the formalisms developed for NER or simply merging the results of several tools, for instance, by using simple voting. By doing so, these approaches fail to make use of the diversity of NER algorithms. On the other hand, it is a well-known fact that algorithms with diverse strengths and weaknesses can be aggregated in various ways to create a system that outperforms the best individual algorithms within the system [290]. While previous works have already suggested that ensemble learning can be used to improve NER [198] no comparison of the performance of existing supervised machine-learning approaches for ensemble learning on the NER task have been presented at the time of writing the thesis.

In this thesis, we propose strategies for improving the quality of knowledge extraction on three core extraction tasks that are important for various research fields, such as KG creation and KGQA.

1.2 RESEARCH QUESTIONS AND CONTRIBUTIONS

In this section, we present a set of research questions derived from the Section 1.1, along with our contributions.

***RQ1**. (A) Does any ensemble learning-based NER algorithm outperform the best basic NER algorithm in terms of higher F-measure; (B) Which ensemble learning algorithm yields the highest F-measure for the NER task, especially compared to simple voting on the tool's outcome without considering entity type levels?*

We address the problem that no comparison of the performance of existing ensemble learning-based approaches on the NER task has been presented so far.

Therefore, we bridge this research gap by presenting a thorough evaluation of ensemble learning-based approaches on NER in *Chapter 3*. To this end, we combine four different state-of-the-art approaches by using 15 different algorithms for ensemble learning and evaluate their performance on five different datasets. Our results suggest that ensemble learning can reduce the error rate of state-of-the-art NER systems by 40%, thereby leading to over 95% *F-score* in our best run.

***RQ2**. Can we identify the types of emerging and existing entities in unstructured text and link these types to a KG?*

Our focus is on the problem that there is no method available, at the time of writing the thesis, for extracting evidences of entity types, i. e., a sub-sequence of words in a sentence containing the entity type and link these types to a subset of given ontology classes.

We bridge this research gap by presenting a baseline approach to entity type extraction in *Chapter 4*. This approach is based on a three-step pipeline comprising: (i) offline, knowledge-driven type pattern extraction from natural language text based on grammar rules, (ii) an analysis of input text to extract types, and (iii) the mapping of the extracted type evidence to a subset of the DOLCE+DnS Ultra Lite ontology classes. We implement and compare two approaches for the third step.

***RQ3**. Can we identify key aspects of holistic EL approaches and adequately describe these aspects?*

We address the problem that no comparison of emerging strategies that influence holistic EL approaches have been presented so far.

To fill this research gap, we conducted a comparative analysis of a variety of holistic EL approaches in *Chapter 5*. We aim to provide a comprehensive review of the topic and show aspects of holistic EL approaches including the exploitation of distinct inputs, data features, and the use of diverse Natural Language Processing (NLP) tasks. However, these aspects of holistic EL approaches have not been adequately described in the literature yet. They are usually implicit in the EL proposals. We define these key aspects of holistic EL approaches

and believe that they can be useful to better understand, classify, and compare these approaches besides giving insights about promising research directions for novel approaches.

***RQ4**. (A) Can we improve the Precision and Recall achieved by state-of-the-art RE while keeping the Distant Supervision (DS) and scalability it abides by; (B) Can distributional semantics decrease the sensitivity to semantic drift of RE approaches based on DS?*

One limitation of distant supervised RE systems is that a single extraction expressing a predefined relation can be matched to several relations, resulting in a significant number of false positives. This leads to noisy behavior, particularly in applications such as question answering. Another limitation of classical RE systems is their inability to extract relations that are not surrounded by mentions of named entities.

In *Chapter 6* we introduce a distant supervised RE approach based on distributional semantics and a tree generalization. Our approach uses training data obtained from a reference KG to derive dependency parse trees that might express a relation. It then uses a novel generalization algorithm to construct dependency tree patterns for the relation. Distributional semantics are used to eliminate false candidate patterns. We evaluate the performance in experiments on a large corpus using ninety target relations. Our evaluation results suggest that our approach achieves a higher precision and recall than two state-of-the-art systems. Furthermore, our results also underpin the scalability of our approach and the decreased sensitivity to semantic drift.

1.3 THESIS STRUCTURE

In this section, we outline the structure of the thesis with an overview of each part. The thesis is divided into five parts with a total number of 12 chapters including two appendix chapters.

The first part consists of two chapters. *Chapter 1* provides an introduction that outlines the author's motivation, research questions, hypotheses and contributions. *Chapter 2* introduces notations and formalizations used in the ensuing chapters of the thesis.

The second part consists of four chapters. *Chapter 3* presents our ensemble learning for NER. Thereafter, we present entity type extraction and linking from unstructured text in *Chapter 4*. Followed by *Chapter 5*, providing our survey and directions towards holistic EL. In the last chapter of the second part, *Chapter 6*, we present our RE using distributional semantics and a tree generalization.

The third part consists of three chapters, Chapters 7 to 9, proposing applications and challenges related to our contributions in the afore-

mentioned second part of the thesis. *Chapter 7* presents our approach for semantifying Content Management Systems (CMSs). In *Chapter 8*, we present a baseline approach to attribute prediction and validation for KG population. In the last chapter of the third part, *Chapter 9*, we report on the organization and participation of the Open Knowledge Extraction (OKE) Challenge.

The fourth part summarizes our insight on strategies for improving knowledge extraction in *Chapter 10*. Furthermore, we present possible extensions to our algorithms and discuss possible applications of our algorithms to different areas of computer science. We conclude the thesis by presenting ideas for future research directions.

In the last part provides the appendix with additional images and tables, acronyms, RDF namespaces, symbols, bibliography, and our declaration.

Parts of the thesis have been published as peer-reviewed articles at research conferences and in scientific journals. Hence, at the beginning of the Chapters 2 to 10 a footnote with the ¶ symbol lists published articles overlapping with the chapter's content of the thesis and the role of the thesis' author within the creation of these articles.

2

PRELIMINARIES

2.1 BASIC FUNDAMENTALS OF TEXT SOURCES

First, we define elements to which we will refer across various chapters within the thesis and can be regarded as basic fundamental building blocks of text sources, specifically, unstructured natural language text.

Definition 2.1 (Sequence) *We follow Zhang et al. (2015) [301] to define a sequence. Let I be a set of distinct items. A sequence $S = \langle a_1, a_2, \ldots, a_n \rangle$ is an ordered list, where $a_i \in I$ is an item for $1 \leq i \leq n$. The length of a sequence, denoted as $|S|$, is the total number of items including repeated items, i. e., $|S| = n$.*

Definition 2.2 (Word) *Let $\omega \in \Sigma^*$ be a word that is a finite sequence of characters over an alphabet Σ, e. g., "Paderborn".*

Definition 2.3 (Sentence) *Let $\mathfrak{s} = \langle \omega_i \rangle_{i=1,2,\ldots}$ be a sentence that is a finite sequence of words. We follow Zhan et al. (2020) [299] and define the set of all subsequences of a sentence with $2^{\mathfrak{s}} = \{\langle \omega_i, \omega_{i+1}, \ldots, \omega_j \rangle : 1 \leq i \leq j \leq |\mathfrak{s}|\}$, where (i, j) is a span of a sentence beginning with the word i and ending with the word j.*

Definition 2.4 (Corpus) *Let $\mathscr{C} = \langle \mathfrak{s}_i \rangle_{i=1,2,\ldots}$ be a corpus that is a finite sequence of sentences.*

2.2 KNOWLEDGE & KNOWLEDGE GRAPH

In the thesis, we follow Hogan et al. (2020) [113] and view a Knowledge Graph (KG) as a graph of data that conforms to a graph-based data model whose nodes represent entities of interest and whose edges represent relations between these entities. The intention is to accumulate (e. g., from external sources, or extracted from the KG itself) and convey knowledge (i. e., something that is known) of the real world in KGs that may be composed of simple or quantified statements. These statements are represented as triples (*head entity, relation, tail entity*) [203, 284] and usually encoded in Resource Description Framework [45, 136] (RDF) [9, 67, 150] triples. Therefore, KGs provide knowledge about entities and their relations in a standardized and machine-processable data model.

Let $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$ be a KG, $\mathcal{E}$ denotes a set of entities (i. e., things of the real world), $\mathcal{R}$ denotes a set of relations (i. e., relationships between things), a set of triples with statements $\mathcal{S} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$,

where each triple $(h, r, t) \in \mathcal{S}$ contains two entities $h, t \in \mathcal{E}$ and a relation $r \in \mathcal{R}$ [48, 49, 139]. We further assume the existence of a labeling function $\gamma : \mathcal{E} \cup \mathcal{R} \to \Sigma^*$ that maps each entity and relation to a word.

2.3 EVALUATION MEASURES

In this section of the thesis, we present common Key Performance Indicators (KPIs) utilized as evaluation measures in multiple chapters.

2.3.1 *Precision, Recall & F-measure*

The thesis evaluation model can be formulated as follows [197]: Let the universe $\mathcal{U}$ be the set of entities a system $\mathcal{A}$ can classify, let $\mathcal{U}_{\mathcal{A}} \subseteq \mathcal{U}$ be the set of classified entities by $\mathcal{A}$, and let $\mathcal{U}_{\mathcal{T}^+} \subseteq \mathcal{U}$ be the set of relevant entities for a given task $\mathcal{T}$, i. e., $\mathcal{U}_{\mathcal{T}^-} = \mathcal{U} \setminus \mathcal{U}_{\mathcal{T}^+}$ are irrelevant.

The Precision of $\mathcal{A}$ computes the ratio between the number of relevant entities classified correctly by $\mathcal{A}$ (i. e., $TP = |\mathcal{U}_{\mathcal{T}^+} \cap \mathcal{U}_{\mathcal{A}}|$) and the total number of entities $\mathcal{A}$ classified (i. e., $TP + FP = |\mathcal{U}_{\mathcal{A}}|$), whereas the Recall computes the ratio between TP and the total number of relevant entities (i. e., $TP + FN = |\mathcal{U}_{\mathcal{T}^+}|$). The number of irrelevant entities classified correctly by $\mathcal{A}$ is given as $TN = |\mathcal{U}_{\mathcal{T}^-} \cap \mathcal{U}_{\mathcal{A}}|$.

Definition 2.5 (Precision) *The Precision [13, 72] is defined as the number of correct classified relevant results at a ratio of the number of all returned results by the evaluated system:*

$$Pr = \frac{TP}{TP + FP} \quad . \tag{2.1}$$

Definition 2.6 (Recall) *The Recall [13, 72] is defined as the number of correct classified relevant results returned by the evaluated system at a ratio of the number of all relevant result:*

$$Re = \frac{TP}{TP + FN} \quad . \tag{2.2}$$

Definition 2.7 (F-measure) *The F-measure [13, 72, 76] can be used to summarize the Precision and the Recall of a system. It is defined as the harmonic mean of Precision and Recall:*

$$F\text{-}score = \frac{2PrRe}{Pr + Re} = \frac{2TP}{2TP + FP + FN} \quad . \tag{2.3}$$

2.3.2 *Matthews Correlation Coefficient & Error Rate*

Definition 2.8 (Matthews Correlation Coefficient) *The Matthews Correlation Coefficient [34, 168] considers both the TP and the TN as correct*

classification and is rather unaffected by sampling biases. Higher values indicating better classifications:

$$Mcc = \frac{TPTN - FPFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad . \tag{2.4}$$

Definition 2.9 (Error Rate) *The Error Rate [155] is the ratio of the number of misclassifications for the classifier to the actual total number. Thus, it monitors the fraction of positive and negative classifications for that the classifier failed:*

$$Er = \frac{FP+FN}{TP+TN+FP+FN} \quad . \tag{2.5}$$

2.3.3 *Micro and Macro Averages*

Microaveraging [126, 235] gathers the decisions for every dataset i globally and consolidates them into a single confusion matrix for which it computes the following averages, hence treating the contributions of all datasets and can be influenced by the dataset sizes:

$$Pr_{\text{mic}} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}, \tag{2.6}$$

$$Re_{\text{mic}} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i}, \tag{2.7}$$

$$\begin{aligned} F\text{-}score_{\text{mic}} &= \frac{2Pr_{\text{mic}}Re_{\text{mic}}}{Pr_{\text{mic}} + Re_{\text{mic}}} \\ &= \frac{2\sum_{i=1}^{n} TP_i}{2\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i + \sum_{i=1}^{n} FN_i} \end{aligned}, \tag{2.8}$$

$$Mcc_{\text{mic}} = \frac{\sum_{i=1}^{n} TP_i \sum_{i=1}^{n} TN_i - \sum_{i=1}^{n} FP_i \sum_{i=1}^{n} FN_i}{\sqrt{(\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i)\ldots(\sum_{i=1}^{n} TN_i + \sum_{i=1}^{n} FN_i)}}, \tag{2.9}$$

$$Er_{\text{mic}} = \frac{\sum_{i=1}^{n} FP_i + \sum_{i=1}^{n} FN_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} TN_i + \sum_{i=1}^{n} FP_i + \sum_{i=1}^{n} FN_i} \quad . \tag{2.10}$$

Macroaveraging [126, 235] on the other hand follows the idea of the *one-against-all* approach [7], thus it computes the performance for each dataset i individually and then computes across all datasets the

following averages, hence treating all datasets equally independent of the dataset sizes:

$$Pr_{\text{mac}} = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} Pr_i = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \frac{TP_i}{TP_i + FP_i}\,, \tag{2.11}$$

$$Re_{\text{mac}} = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} Re_i = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \frac{TP_i}{TP_i + FN_i}\,, \tag{2.12}$$

$$\textit{F-score}_{\text{mac}} = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \textit{F-score}_i = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \frac{2TP_i}{2TP_i + FP_i + FN_i}\,, \tag{2.13}$$

$$Mcc_{\text{mac}} = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} Mcc_i = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \frac{TP_i TN_i - FP_i FN_i}{\sqrt{(TP_i + FP_i) \dots (TN_i + FN_i)}}\,, \tag{2.14}$$

$$Er_{\text{mac}} = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} Er_i = \frac{1}{\mathfrak{n}} \sum_{i=1}^{\mathfrak{n}} \frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i}\,. \tag{2.15}$$

2.4 TOKEN-WISE & ENTITY-WISE EVALUATION

The thesis evaluation model considers two evaluation cases, since it can happen that only word segments may be classified as being correct by a system. Hence, we regard partial matches of a word as partially correct in the *token-wise evaluation* (i. e., following the idea of weak entity matching [279]). Whereas in the *entity-wise evaluation* (i. e., following the idea of strong entity matching [279]), we only regard exact matches as correct.

For example, a gold standard dataset, expressing the ground truth, considers the country "*Federal Republic of Germany*" as being of the entity type `LOCATION`. If a system classifies "*Germany*" as being of the correct type and omits "*Federal Republic of*", it is assigned 1 *TP* and 3 *FN* in the token-wise evaluation. In the entity-wise evaluation, this classification is simply considered to be incorrect.

Part II

KNOWLEDGE EXTRACTION

3

ENSEMBLE LEARNING FOR NAMED ENTITY RECOGNITION

In this chapter, we present our thorough evaluation of ensemble learning for Named Entity Recognition (NER). Later in Chapter 7, we introduce in an early prototype of our approach and describe, on a practical use case example, semantic data integration with linked data of our approach.

3.1 RELATED WORK

NER tools and frameworks implement a broad spectrum of approaches. Nadeau et al. (2007) [188] and Nasar et al. (2021) [193] give an exhaustive overview for the NER task. Li et al. (2020) [154] surveys on deep learning for NER, which is not covered in this work, as deep learning had not been developed at the time this article was written.

The first systems for NER implemented dictionary-based approaches, which relied on a list of Named Entities (NEs) and tried to identify these in text [8, 281]. Following work then showed that these approaches did not perform well for NER tasks such as recognizing proper names [245]. Thus, rule-based approaches were introduced. These approaches rely on hand-crafted rules [38, 267] to recognize NEs. Most rule-based approaches combine dictionary and rule-based algorithms to extend the list of known entities. Nowadays, hand-crafted rules for recognizing NEs are usually implemented when no training examples are available for the domain or language to process [189]. When training examples are available, the methods of choice are borrowed from supervised machine learning. Approaches such as Hidden Markov Models [305], Maximum Entropy Models [43] and Conditional Random Fields [75] have been applied to the NER task. Due to scarcity of large training corpora as necessitated by supervised machine learning approaches, the semi-supervised [188, 216] and unsupervised machine learning paradigms [66, 190] have also been used for extracting NER from text. In [290], a system was presented that combines with stacking and voting classifiers which were trained with several languages, for language-independent NER.

Several benchmarks for NER have been proposed at the time of writing the thesis. For example, [41] presents a benchmark for NER

¶ Parts of this chapter have been published as conference articles [259–261]. The author of this thesis is also the main author of these articles. For these three publications, the author developed the main ideas, designed, and implemented major parts of the solution, and wrote the majority of the publication.

and Entity Linking (EL) approaches. Especially, the authors define the NE annotation task. Other benchmark datasets include the manually annotated datasets presented in [237]. Here, the authors present annotated datasets extracted from RSS feeds as well as datasets retrieved from news platforms. Other authors designed datasets to evaluate their own systems. For example, the *Web* dataset (which we use in our evaluation) is a particularly noisy dataset designed to evaluate the system presented in [230]. The dataset *Reuters*, which we also use, consists annotated documents chosen out of the *Reuters-215788* corpus and was used in [16].

3.2 OVERVIEW

In this section, we give a short introduction in NER, ensemble learning and our framework for ensemble learning-based NER.

3.2.1 *Named Entity Recognition*

NER encompasses two main tasks: (i) The identification of mentions of NEs such as "*Germany*", "*Paderborn University*" and "*G. W. Leibniz*" in a given unstructured text [189, 230], and (ii) the classification of these mentions into predefined entity types, such as `LOCATION`, `ORGANIZATION` and `PERSON` [159, 191].

In general[1], this task can be viewed as the sequential prediction problem of estimating the probabilities $P(\tau_i|\omega_{i-j} \dots \omega_{i+k}, \tau_{i-l} \dots \tau_{i-1})$, where $\langle \omega_i \rangle_{i=1,2,\dots}$ is an input sequence of words, (i. e., the preprocessed sentence) and $\langle \tau_i \rangle_{i=1,2,\dots}$ the output sequence (i. e., the entity types). Furthermore, the indices j, k and l are relative small numbers to allow tractable inference and avoid overfitting of this conditional probability distribution. Several features for estimating this distribution exist, for instance regarding context aggregation, the tokens in the window $\langle \omega_{i-2}, \omega_{i-1}, \omega_i, \omega_{i+1}, \omega_{i+2} \rangle$ or the conjunction of such a window and τ_{i-1}. Most classical NER systems use additional features, e. g., Part of Speech (POS) tags and additional external knowledge such as gazetteers or word class models, i. e., Brown clusters [25].

3.2.2 *Ensemble Learning*

The goal of an ensemble learning algorithm $\mathfrak{E}$ is to generate a classifier $\mathcal{C}$ with a high predictive performance by combining the predictions of a set of m basic classifiers $\mathcal{B}_1, \dots, \mathcal{B}_m$ [57]. One central observation in this respect, combining $\mathcal{B}_1, \dots, \mathcal{B}_m$ can only lead to a high predictive performance when these basic classifiers are *accurate* and *diverse* [294]. Several approaches have been developed to allow an efficient com-

1 We following the definition of Ratinov and Roth [230] for the NER task.

bination of basic classifiers. The simplest strategy is voting, where each input token is classified as belonging to the entity type that was predicted by the largest number of basic classifiers [57]. Voting can be extended to weighted voting, where each of the basic classifiers is assigned a weight and $\mathcal{C}$ returns the entity type with the highest total prediction weight. More elaborate methods try to ensure the diversity of the classifiers. Approaches that aim to achieve this goal include drawing random samples (with replacement) from the training data (e. g., bagging [21]) or generating sequences of classifiers of high diversity that are trained to recognized each other's mistakes (e. g., boosting [246]). The results of all classifiers are finally combined via weighted voting.

3.2.3 *Named Entity Recognition based on Ensemble Learning*

Here, we consider ensemble learning for NER. Thanks to the long research tradition on the NER topic, the *diversity* and *accuracy* of the tools is already available and can be regarded as given. However, classical ensemble learning approaches present the disadvantage of relying on some form of weighted vote on the output of the classifiers. Thus, if all classifiers $\mathcal{B}_i$ return wrong results, classical ensemble learning approaches are bound to make the same mistake [57]. In addition, voting does not take the different levels of accuracy of classifiers for different entity types into consideration. Rather, it assigns a global weight to each classifier that describes its overall accuracy. Based on these observations, we decided to apply ensemble learning for NER based at entity type level. The main advantage of this ensemble-learning setting is that we can now assign different weights to each tool-type pair.

Formally, we model the ensemble learning task at hand as follows: Let the matrix $M^{mt \times n}$ (Equation 3.1) illustrate the input data for $\mathcal{C}$, where $\mathcal{P}^m_{n,t}$ are predictions of the m-th NER tool that the n-th token is of the t-th type.

$$\begin{pmatrix} \mathcal{P}^1_{1,1} & \cdots & \mathcal{P}^1_{1,t} & \mathcal{P}^2_{1,1} & \cdots & \mathcal{P}^2_{1,t} & \cdots & \mathcal{P}^m_{1,1} & \cdots & \mathcal{P}^m_{1,t} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \mathcal{P}^1_{n,1} & \cdots & \mathcal{P}^1_{n,t} & \mathcal{P}^2_{n,1} & \cdots & \mathcal{P}^2_{n,t} & \cdots & \mathcal{P}^m_{n,1} & \cdots & \mathcal{P}^m_{n,t} \end{pmatrix} \tag{3.1}$$

The goal of ensemble learning for NER is to detect a classifier that leads to a correct classification of each of the n tokens into one of the types τ.

3.3 EVALUATION

We performed a thorough evaluation of ensemble learning approaches on the NER task by using five different datasets and running a 10-

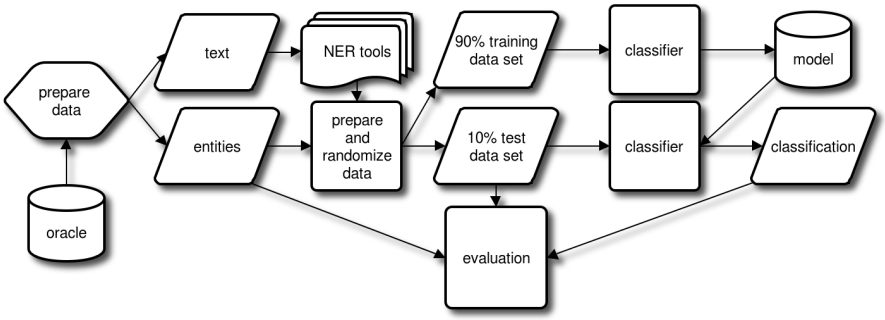


Figure 3.1: Workflow chart of our evaluation pipeline.

fold cross-validation for 15 algorithms. In this section, we present the pipeline, it's setup for our evaluation and our results.

3.3.1 *Pipeline*

Figure 3.1 shows the workflow chart of our evaluation pipeline. In the first step of our evaluation pipeline, we preprocessed our reference dataset to extract the input text (i. e., *text sources*) for the NER tools and to extract the correct NEs, which we used to create training and testing data. In the second step, we made use of all NER tools with this input text to calculate the predictions of all entity types for each token in this input. At this point, we represented the output of the tools as matrix (see Equation 3.1). Thereafter, the matrix was randomly split into 10 disjoint sets as preparation for a 10-fold cross-validation. We trained the different ensemble learning algorithms at hand (i. e., $\mathcal{C}$) with the training dataset (i. e., with 9 of 10 sets) and tested the trained classifier with the testing dataset (i. e., with the leftover set). To use each of the 10 sets as testing set once, we repeated training and testing of the classifiers 10 times and used the disjoint sets accordingly. Furthermore, the pipeline was repeated 10 times to deal with non-deterministic classifiers. In the last step, we compared the classification of the 10 testing datasets with the oracle dataset (i. e., ground truth or gold standard) to calculate Key Performance Indicators (KPIs) for our evaluation.

We ran our pipeline on 15 ensemble learning algorithms. We carried out both a token-wise evaluation and an entity-wise evaluation (Section 2.4). To provide transparent results, we only used open-source libraries in our evaluation. Given that some of these tools at hand do not allow accessing their confidence score without any major alteration of their code, we considered the output of the tools to be binary (i. e., either 1 or 0).

We integrated four NER tools so far: (i) the Stanford Named Entity Recognizer[2] (Stanford) [75], (ii) the Illinois Named Entity Tag-

2 `http://nlp.stanford.edu/software/CRF-NER.shtml` (version 3.2.0)

ger[3] (ILLINOIS) [230], (iii) the Ottawa Baseline Information Extraction[4] (BALIE) [187], and (iv) the Apache OpenNLP Name Finder[5] (OPENNLP) [15]. We only considered the performance of these tools on the entity types set $\mathscr{T} = \{\texttt{LOCATION}, \texttt{ORGANIZATION}, \texttt{PERSON}\}$. To this end, we mapped the entity types of each of the NER tools to these three types.

We utilized the Waikato Environment for Knowledge Analysis (Weka) [99] and the implemented ensemble learning algorithms with default parameters: (i) AdaBoostM1 [79] with J48 as base classifier (ABM1), (ii) Bagging [21] with J48 as base classifier (BG), (iii) Decision Table [137] (DTable), (iv) Functional Trees [84, 144] (FT), (v) A pruned C4.5 DT [228] (J48), (vi) Logistic Model Trees [144, 266] (LMT), (vii) Logistic Regression [29] (LOG), (viii) Additive LOG [80] (LogD), (ix) Multilayer Perceptron [233] (MLP), (x) Naïve Bayes [123] (NB), (xi) Random Forest [22] (RF), (xii) Support Vector Machine [31] (SVM), and (xiii) Sequential Minimal Optimization [104] (SMO).

In addition, we used: (xiv) Voting approach [290] with the majority vote rule [133] (MVote), and (xv) Voting approach at entity type level (TVote). MVote as naive approach combines the results of the NER tools with equal weights and the majority vote rule [133]. It was the baseline ensemble learning technique in our evaluation. In contrast, TVote selects the NER tool with the highest prediction performance for each entity type according to the evaluation and applies that particular tool for the given type.

3.3.2 *Experimental Setup*

The experimental setup for our evaluation includes five datasets, five KPIs and the recommended *Wilcoxon signed-rank test* to measure the statistical significance of our results [50]. For this purpose, we applied each measurement of the ten 10-fold cross-validation runs for the underlying distribution and we set up a 95% confidence interval.

Table 3.1 shows an overview of the datasets in our setup. The *Web* dataset consists of 20 annotated websites as described in [230] and contains the most noise compared to the other datasets. The dataset *Reuters* consists of 50 documents randomly chosen out of the Reuters-215788 corpus[6] [16]. *News** is a small subset of the dataset *News* that consists of text from newspaper articles and was re-annotated manually by the authors to ensure high data quality. Likewise, *Reuters* was extracted and annotated manually by the authors. The last dataset,

3 `http://cogcomp.cs.illinois.edu/page/software_view/NETagger` (version 2.4.0)

4 `http://balie.sourceforge.net` (version 1.8.1)

5 `http://opennlp.apache.org/index.html` (version 1.5.3)

6 The Reuters-215788 corpus is available at: `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

Table 3.1: Number of entities in the datasets separated according three entity types and summarized in total in the last column of the table.

Datasets	*News*	*News**	*Web*	*Reuters*	*All*
LOCATION	5117	341	114	146	5472
ORGANIZATION	6899	434	257	208	7467
PERSON	3899	254	396	91	4549
Total	15915	1029	767	445	17488

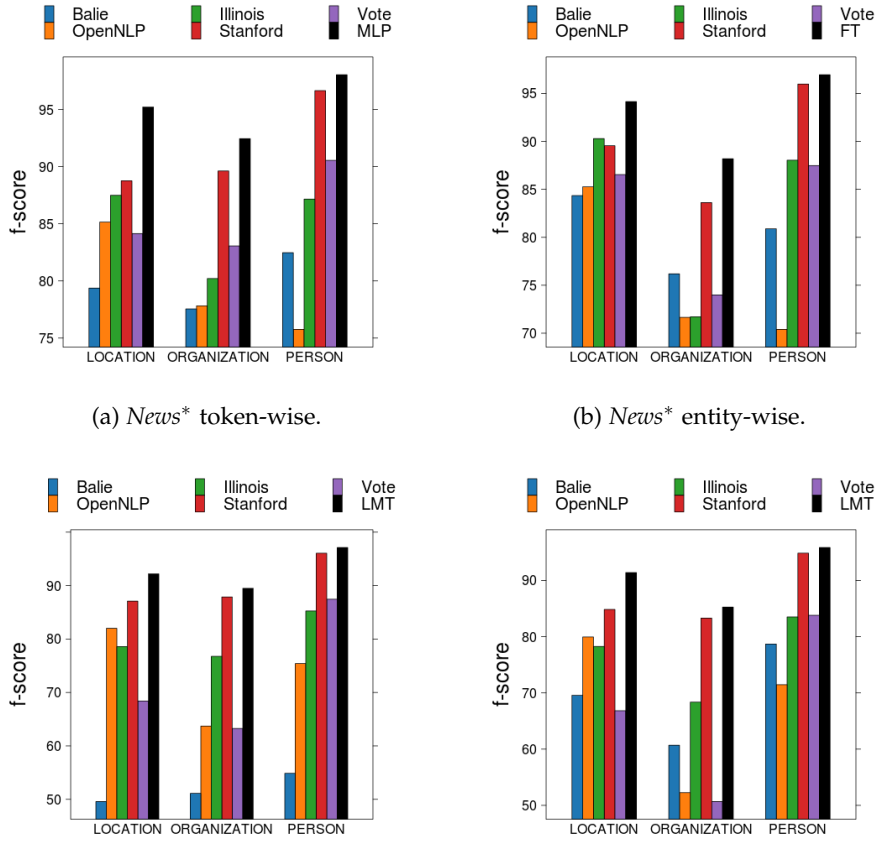
All, consists of the datasets mentioned before merged into one and allows for measuring how well the ensemble learning approaches perform when presented with data from heterogenous sources.

We computed the following values on the test datasets for each entity type $\tau \in \mathscr{T}$: $TP_\tau, TN_\tau, FP_\tau, FN_\tau$ to assess the performance of the different algorithms. These numbers were collected for each entity type τ and averaged over the ten runs of the 10-fold cross-validations. Then, we applied the *one-against-all* approach [7] to convert the multi-class confusion matrix of each dataset into a binary confusion matrix. Subsequently, we determined with macroaveraging the measures given in Equations (2.11) to (2.15). In this case, the number of datasets n corresponds to the number of entity types $|\mathscr{T}|$.

3.4 RESULTS

Tables 3.2 to 3.11 show the results of our evaluation for the 15 classifiers we learned within our pipeline and the four basic NER tools we integrated so far. The best results are marked bold and the basic NER tools are underlined. Figures 3.2 to 3.4 depict the F-measures separated according to the three entity types for the four NER tools, the simple voting approach MVote and the best classifier for the specified dataset.

We reached the highest F-measures on the *News** dataset (Table 3.2 and Table 3.3) for both the token-wise and the entity-wise evaluation. In the token-wise evaluation, the MLP and RF classifiers perform best for Precision ($Pr = 95.28\%$), Error Rate ($Er = 0.32\%$) and Matthews Correlation Coefficient ($Mcc = 0.951$). MLP performs best for F-measure ($F\text{-}score = 95.23\%$) with 0.04% more Recall than RF. The baseline classifier (i. e., MVote) is clearly outperformed by MLP by up to +5.21% Recall, +12.31% Precision, +9.31% F-measure, -0.62% Error Rate and +0.094 Matthews Correlation Coefficient. Furthermore, the best single approach is STANFORD and outperformed by up to +2.83% Recall, +4.27% Precision, +3.55% F-measure, -0.21% Error Rate (that is a reduction by 40%) and +0.037 Matthews Correlation Coefficient. Slightly poorer results are achieved in the entity-wise evaluation, where MLP is second to FT with 0.01% less F-measure.



(a) *News** token-wise.

(b) *News** entity-wise.

(c) *News* token-wise.

(d) *News* entity-wise.

Figure 3.2: F-measures for the *News* and *News** dataset.

Table 3.2: Our results on the *News** dataset with the token-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
MLP	**95.19**	**95.28**	**95.23**	**0.32**	**0.951**
RF	95.15	**95.28**	95.21	**0.32**	**0.951**
ABM1	94.82	95.18	95.00	0.33	0.948
SVM	94.86	95.09	94.97	0.33	0.948
J48	94.78	94.98	94.88	0.34	0.947
BG	94.76	94.93	94.84	0.34	0.947
LMT	94.68	94.95	94.82	0.34	0.946
DTable	94.63	94.95	94.79	0.34	0.946
FT	94.30	95.15	94.72	0.35	0.945
LogD	93.54	95.37	94.44	0.37	0.943
LOG	94.05	94.75	94.40	0.37	0.942
SMO	94.01	94.37	94.19	0.39	0.940
NB	94.61	92.64	93.60	0.42	0.934
STANFORD	92.36	91.01	91.68	0.53	0.914
TVote	92.02	90.84	91.42	0.54	0.911
MVote	89.98	82.97	85.92	0.94	0.857
ILLINOIS	82.79	87.35	84.95	0.92	0.845
BALIE	77.68	82.05	79.80	1.21	0.792
OPENNLP	71.42	90.47	79.57	1.13	0.797

Table 3.3: Our results on the *News** dataset with the entity-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
FT	93.95	**92.27**	**93.10**	**0.30**	**0.930**
MLP	**94.10**	92.13	93.09	**0.30**	0.929
LMT	94.08	91.91	92.97	0.31	0.928
RF	93.76	92.07	92.90	0.31	0.928
BG	93.51	92.18	92.83	0.31	0.927
SVM	93.85	91.46	92.62	0.32	0.925
ABM1	93.30	91.65	92.47	0.33	0.923
J48	93.30	91.65	92.47	0.33	0.923
LOG	93.42	91.39	92.37	0.33	0.922
LogD	92.89	91.68	92.27	0.33	0.921
SMO	92.55	91.26	91.90	0.36	0.917
DTable	92.44	91.29	91.86	0.34	0.917
NB	94.08	88.26	91.01	0.40	0.909
STANFORD	92.00	87.58	89.72	0.45	0.895
TVote	91.43	86.94	89.10	0.47	0.889
ILLINOIS	82.07	84.84	83.34	0.67	0.831
MVote	91.42	76.52	82.67	0.83	0.829
BALIE	81.54	79.66	80.48	0.79	0.801
OPENNLP	69.36	85.02	75.78	0.88	0.760

Table 3.4: Our results on the *News* dataset with the token-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
LMT	**93.73**	92.16	**92.94**	**0.51**	**0.927**
RF	93.56	92.19	92.87	**0.51**	0.926
DTable	93.64	92.10	92.86	**0.51**	0.926
J48	93.50	92.20	92.84	0.52	0.926
ABM1	93.49	92.17	92.83	0.52	0.926
BG	93.11	**92.49**	92.79	0.52	0.925
FT	93.44	92.15	92.79	0.52	0.925
MLP	93.22	92.26	92.73	0.52	0.925
SVM	92.19	**92.49**	92.31	0.54	0.920
SMO	92.15	91.90	92.01	0.57	0.917
LOG	91.38	91.36	91.35	0.63	0.910
LogD	91.42	91.32	91.34	0.62	0.910
Stanford	92.70	88.09	90.34	0.68	0.900
TVote	92.70	88.09	90.34	0.68	0.900
NB	93.36	86.17	89.58	0.77	0.893
Illinois	82.43	78.11	80.20	1.37	0.795
OpenNLP	75.21	74.41	73.71	2.06	0.732
MVote	83.13	69.14	73.03	2.36	0.735
Balie	70.81	72.86	71.54	1.90	0.707

Table 3.5: Our results on the *News* dataset with the entity-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
LMT	**92.95**	88.84	**90.84**	**0.44**	**0.906**
BG	92.82	88.95	90.83	**0.44**	**0.906**
DTable	92.89	88.88	90.83	**0.44**	**0.906**
ABM1	92.87	88.82	90.79	**0.44**	**0.906**
J48	92.87	88.82	90.79	**0.44**	**0.906**
FT	92.90	88.78	90.78	**0.44**	**0.906**
RF	92.84	88.77	90.74	**0.44**	**0.906**
MLP	92.83	88.69	90.70	**0.44**	0.905
SVM	91.56	**89.22**	90.33	0.45	0.901
SMO	91.13	88.36	89.69	0.49	0.895
LOG	90.62	88.09	89.29	0.51	0.891
LogD	90.76	87.83	89.22	0.51	0.890
Stanford	91.78	83.92	87.66	0.58	0.875
TVote	91.78	83.92	87.66	0.58	0.875
NB	92.54	81.16	86.34	0.69	0.863
Illinois	81.66	72.50	76.71	1.11	0.763
Balie	71.58	68.67	69.66	1.42	0.692
OpenNLP	72.71	67.29	67.89	1.80	0.681
MVote	82.71	61.30	67.10	2.19	0.686

On the *News* dataset (Table 3.4-Table 3.5), which was the largest homogenous dataset in our evaluation, we repeatedly achieved high F-measures. The best approach w.r.t. the token-wise evaluation is LMT with an F-measure of 92.94%. RF follows the best approach with respect to F-measure again. Moreover, the best single tool Stanford and the baseline classifier MVote are repeatedly outperformed by up to +2.6% resp. +19.91% F-measure. Once again, the entity-wise results are approximately 2% poorer, with LMT leading the table like in the token-wise evaluation.

On the *Web* dataset, which is the worst-case dataset for NER tools as it contains several incomplete sentences, the different classifiers reached their lowest values (Table 3.6-Table 3.7). For the token-wise evaluation, ABM1 achieves the best *F-score* (69.04%) and *Mcc* (0.675) and is followed by RF again with respect to *F-score*. NB performs best for *Re* (96.64%), LOG for *Pr* (77.89%) and MLP and RF for the *Er* (3.33%). Simple voting is outperformed by ABM1 by up to +3.5% *Re*, +20.08% *Pr*, +10.45% *F-score*, -2.64% *Er* and +0.108 *Mcc*, while Stanford (the best tool for this dataset) is outperformed by up to +3.83% *Re*, +2.64% *Pr*, +3.21% *F-score*, −0.13% *Er* and +0.032 MCC. Similar insights can be won from the entity-wise evaluation, with some classifiers like RF being approximately 10% poorer at token level.

On the *Reuters* dataset (Table 3.8-Table 3.9), which was the smallest dataset in our evaluation, SVM performs best. In the token-wise evaluation, SVM achieves an *F-score* of 87.78%, an *Er* of 0.89% and a Matthews correlation coefficient of 0.875%. They are followed by RF with respect to *F-score* once again. NB performs best for *Re* (86.54%). In comparison, ensemble learning outperforms MVote with SVM by

Table 3.6: Our results on the *Web* dataset with the token-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
ABM1	64.40	74.83	**69.04**	3.38	**0.675**
RF	64.36	74.57	68.93	3.38	0.674
MLP	63.86	75.11	68.81	**3.33**	0.674
FT	62.98	75.47	68.25	**3.33**	0.670
LMT	63.39	74.24	68.04	3.43	0.666
DTable	62.80	74.18	67.85	3.43	0.664
TVote	63.16	73.54	67.66	3.49	0.662
SVM	62.94	73.45	67.60	3.49	0.661
LogD	60.47	77.48	67.57	3.40	0.665
LOG	60.31	**77.89**	67.50	3.39	0.666
SMO	63.47	72.45	67.49	3.57	0.659
BG	61.06	76.19	67.46	3.34	0.663
J48	62.21	73.78	67.21	3.49	0.658
NB	**71.19**	63.42	66.88	4.42	0.647
Stanford	60.57	72.19	65.81	3.51	0.643
Illinois	69.64	60.56	64.44	5.09	0.621
MVote	66.90	54.75	58.59	6.02	0.567
OpenNLP	45.71	58.81	49.18	5.93	0.477
Balie	38.63	43.83	40.15	7.02	0.371

Table 3.7: Our results on the *Web* dataset with the entity-wise evaluation.

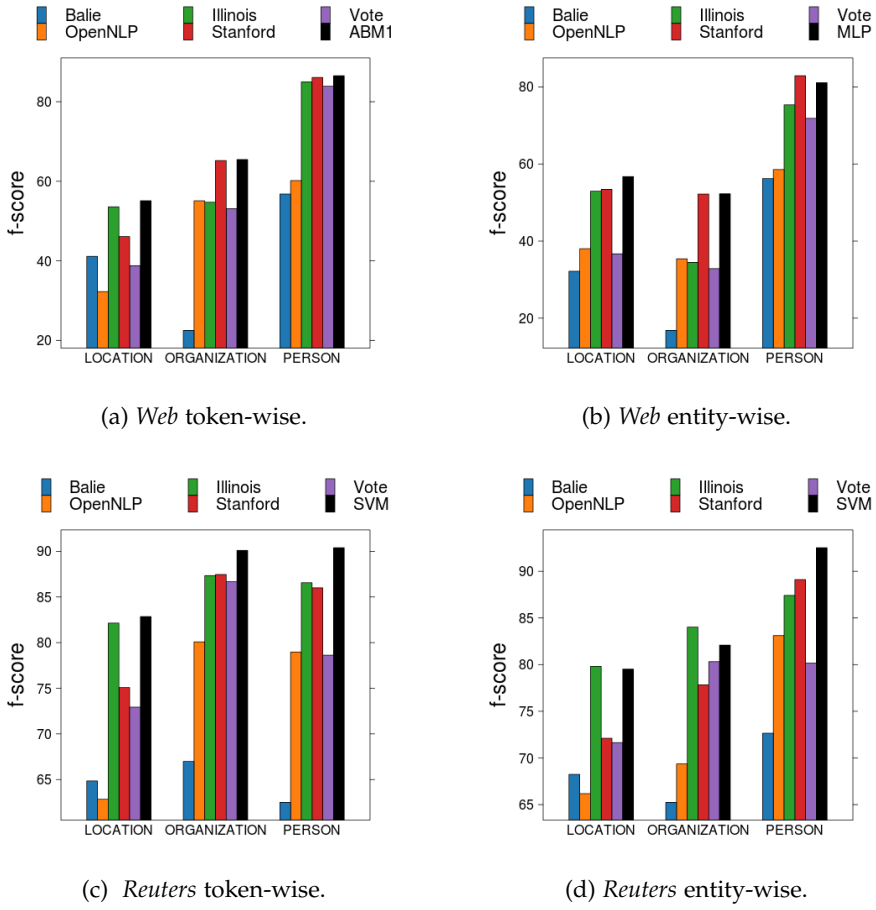
$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
MLP	64.95	61.86	**63.36**	1.99	**0.624**
Stanford	64.80	61.31	62.83	1.95	0.619
LogD	61.25	**64.10**	62.60	**1.94**	0.616
FT	63.67	61.10	62.21	2.09	0.612
ABM1	63.49	61.01	62.17	2.08	0.611
LOG	60.43	63.62	61.95	1.99	0.610
TVote	65.69	59.54	61.82	2.05	0.612
J48	63.21	59.72	61.39	2.12	0.603
BG	64.04	59.10	61.30	2.13	0.603
RF	64.15	55.88	59.69	2.27	0.587
SVM	62.36	57.26	59.57	2.15	0.586
DTable	61.92	57.05	59.34	2.17	0.583
LMT	61.25	56.89	58.96	2.19	0.579
SMO	62.44	56.01	58.83	2.21	0.579
NB	**74.18**	49.20	58.55	3.17	0.586
Illinois	69.31	45.85	54.25	3.82	0.541
MVote	67.42	37.77	47.12	4.84	0.477
OpenNLP	46.94	46.78	43.99	3.71	0.437
Balie	38.07	32.92	35.07	3.63	0.334

Table 3.8: Our results on the *Reuters* dataset with the token-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
SVM	84.57	**91.75**	**87.78**	**0.89**	**0.875**
RF	86.11	89.24	87.58	0.90	0.872
MLP	85.89	89.46	87.55	0.90	0.871
LMT	84.41	91.08	87.43	**0.89**	0.871
J48	84.64	90.70	87.33	0.93	0.870
LOG	84.33	90.85	87.27	**0.89**	0.870
LogD	84.22	91.01	87.22	0.90	0.870
ABM1	84.51	90.47	87.15	0.93	0.868
BG	84.70	90.16	87.14	0.94	0.868
FT	85.25	88.75	86.87	0.95	0.864
DTable	84.41	89.00	86.43	0.99	0.861
SMO	84.45	88.49	86.28	0.98	0.859
Illinois	83.74	88.27	85.35	1.09	0.851
NB	**86.54**	83.18	84.77	1.10	0.842
TVote	81.96	88.66	84.64	1.14	0.844
Stanford	81.57	84.85	82.85	1.20	0.824
MVote	80.11	81.15	79.41	1.43	0.793
OpenNLP	67.94	82.08	73.96	1.76	0.736
Balie	64.92	68.61	64.78	2.62	0.645

Table 3.9: Our results on the *Reuters* dataset with the entity-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
SVM	81.37	**88.85**	**84.71**	**0.69**	**0.846**
ABM1	80.60	88.72	84.15	0.73	0.840
LMT	80.80	87.92	83.96	0.73	0.838
J48	80.41	88.50	83.95	0.73	0.838
BG	80.55	87.70	83.75	0.75	0.836
Illinois	82.77	85.73	83.74	0.72	0.836
LogD	80.70	86.23	83.32	0.75	0.830
DTable	81.11	85.20	82.95	0.79	0.827
RF	80.08	86.11	82.86	0.78	0.826
LOG	80.01	85.51	82.62	0.78	0.823
MLP	80.27	84.09	81.98	0.83	0.817
SMO	79.62	83.21	81.36	0.88	0.809
FT	80.00	82.71	81.32	0.85	0.809
TVote	77.86	85.42	81.00	0.85	0.809
NB	**83.80**	77.68	80.61	0.92	0.802
Stanford	77.56	82.38	79.68	0.90	0.794
MVote	80.35	76.25	77.37	1.03	0.773
OpenNLP	66.85	80.33	72.89	1.18	0.726
Balie	68.90	70.14	68.71	1.39	0.684



(a) *Web* token-wise.

(b) *Web* entity-wise.

(c) *Reuters* token-wise.

(d) *Reuters* entity-wise.

Figure 3.3: F-measures for the *Web* and *Reuters* dataset.

up to +4.46% *Re*, +3.48% *Pr*, +2.43% *F-score*, -0.54% *Er* and +0.082 *Mcc*. Moreover, the best NER tool for this dataset, Illinois, is outperformed by up to +0.83% *Re*, +3.48% *Pr*, +2.43% *F-score*, -0.20% *Er* and +0.024 *Mcc*. In Figure 3.3a, we barely see a learning effect as ABM1 is almost equal to one of the integrated NER tools assessed at entity type level especially for the entity type `ORGANIZATION` on the *Web* dataset but in Figure 3.3c on the *Reuters* dataset we clearly see a learning effect for the entity type `ORGANIZATION` and `PERSON` with the SVM approach.

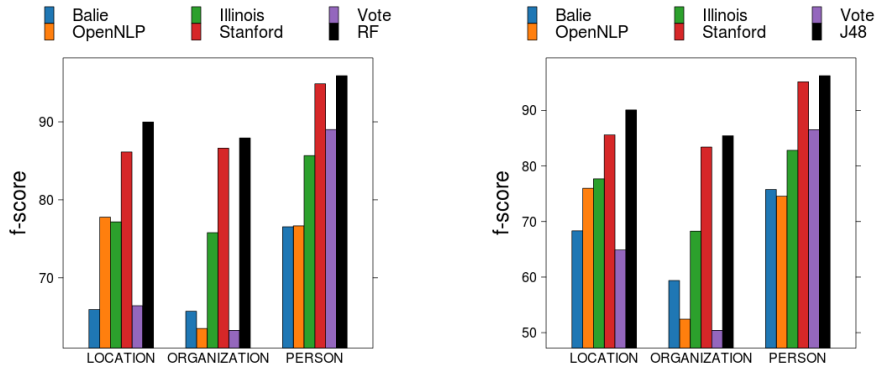
On the *All* dataset for token-wise evaluation (Table 3.10), the RF approach performs best for *F-score* (91.27%), *Er* (0.64%) and *Mcc* (0.909). SVM achieves the best *Pr* (91.24%) and NB the best *Re* (91.00%) again. In comparison, ensemble learning outperformed MVote with RF by up to +9.71% *Re*, +21.01% *Pr*, +18.37% *F-score*, -1.8% *Er* and +0.176% *Mcc* and Stanford, the best tool for this dataset, by up to +0.83% *Re*, +3.24% *Pr*, +2.06% *F-score*, -0.14% *Er* and +0.021% *Mcc*. Again, entity-wise evaluation (Table 3.11) compared to token-wise evaluation, the *F-score* of J48, the best ensemble learning approach here, is approximately 1% poorer with higher *Re* but lower *Pr*. In

Table 3.10: Our results on the *All* dataset with the token-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
RF	91.58	90.97	**91.27**	**0.64**	**0.909**
LMT	91.67	90.86	91.26	**0.64**	**0.909**
ABM1	91.49	90.99	91.24	**0.64**	**0.909**
J48	91.46	90.98	91.22	**0.64**	**0.909**
DTable	91.59	90.84	91.21	**0.64**	**0.909**
FT	91.49	90.82	91.16	0.65	0.908
BG	91.25	91.00	91.12	0.65	0.908
MLP	90.94	91.05	90.99	0.66	0.907
SVM	90.15	**91.24**	90.67	0.67	0.903
SMO	90.13	90.48	90.27	0.71	0.899
LOG	88.69	90.57	89.59	0.76	0.892
LogD	88.92	90.21	89.53	0.76	0.892
STANFORD	90.75	87.73	89.21	0.78	0.888
TVote	90.75	87.73	89.21	0.78	0.888
NB	**92.00**	85.27	88.46	0.89	0.881
ILLINOIS	81.66	77.61	79.54	1.48	0.788
MVote	81.85	69.96	72.90	2.44	0.733
OPENNLP	72.63	75.60	72.65	2.19	0.723
BALIE	67.75	71.65	69.40	2.09	0.685

Table 3.11: Our results on the *All* dataset with the entity-wise evaluation.

$\mathcal{C}$	*Re* [%]	*Pr* [%]	*F-score* [%]	*Er*	*Mcc*
J48	92.68	88.62	**90.59**	**0.44**	**0.904**
ABM1	92.66	88.59	90.56	**0.44**	**0.904**
LMT	92.59	88.50	90.48	0.45	0.903
DTable	92.56	88.44	90.44	0.45	0.902
RF	92.51	88.33	90.35	0.45	0.902
FT	92.47	88.37	90.35	0.45	0.902
BG	92.17	88.55	90.31	0.45	0.901
MLP	92.07	88.60	90.28	0.45	0.901
SVM	90.91	**88.97**	89.88	0.46	0.897
SMO	90.94	87.31	89.00	0.52	0.888
LOG	89.49	88.10	88.70	0.53	0.885
LogD	89.21	87.68	88.36	0.54	0.881
STANFORD	92.00	84.48	88.05	0.56	0.879
TVote	92.00	84.48	88.05	0.56	0.879
NB	**92.69**	80.59	86.04	0.71	0.860
ILLINOIS	81.43	71.82	76.25	1.12	0.759
BALIE	69.27	67.47	67.82	1.48	0.674
OPENNLP	71.29	69.44	67.66	1.80	0.682
MVote	81.97	62.17	67.27	2.17	0.687



(a) *All* token-wise.

(b) *All* entity-wise.

Figure 3.4: F-measures for the *All* dataset.

Figure 3.4, we clearly see a learning effect for RF and J48 at entity type level.

Overall, ensemble learning outperforms all included basic NER tools and the simple voting approach for all datasets with respect to F-measure.

Here, it is worth mentioning that STANFORD and ILLINOIS are the best basic tools in our framework. The three best classifiers with respect to the averaged *F-score*s over our datasets for token-wise evaluation are the RF classifier with the highest value, closely followed by MLP and ABM1 and for entity-wise evaluation ABM1 with the highest value, closely followed by MLP and J48. We cannot observe a significant difference between these.

Table 3.12: *F-score*s [%] of the best 3 classifiers on entity type level token-wise.

$\mathcal{C}$	τ	*News*	*News**	*Web*	*Reuters*	*All*
RF	LOCATION	**92.12**	94.96	**54.58**	**82.25**	**89.98**
RF	ORGANIZATION	**89.45**	**92.44**	**65.60**	**90.53**	**87.93**
RF	PERSON	97.02	**98.25**	**86.61**	89.95	**95.91**
MLP	LOCATION	91.79	**95.22**	53.78	**82.13**	89.62
MLP	ORGANIZATION	89.34	**92.45**	**65.72**	**90.38**	87.63
MLP	PERSON	**97.07**	98.04	**86.94**	90.14	95.73
ABM1	LOCATION	91.75	95.10	55.11	81.19	89.90
ABM1	ORGANIZATION	**89.49**	92.00	65.47	89.91	**87.96**
ABM1	PERSON	**97.12**	97.89	86.53	90.37	95.87

Table 3.13: *F-score*s [%] of the best 3 classifiers on entity type level entity-wise.

$\mathcal{C}$	τ	*News*	*News**	*Web*	*Reuters*	*All*
ABM1	LOCATION	91.26	95.71	**58.21**	**78.99**	90.05
ABM1	ORGANIZATION	**85.19**	85.87	50.66	**80.45**	85.43
ABM1	PERSON	**95.91**	95.81	77.63	**93.02**	96.21
MLP	LOCATION	91.14	95.35	56.72	76.32	89.63
MLP	ORGANIZATION	85.17	**87.30**	**52.29**	78.74	85.38
MLP	PERSON	95.79	96.61	**81.09**	90.88	95.83
J48	LOCATION	**91.27**	95.71	56.53	**78.99**	**90.08**
J48	ORGANIZATION	85.18	85.87	50.56	**80.49**	85.44
J48	PERSON	**95.91**	95.81	77.10	92.36	**96.23**

In Table 3.12 and Table 3.13, we depict the *F-score*s of these three classifiers at entity type level for our datasets. The statistically significant differences are marked in bold. Note that two out of three scores being marked bold for the same setting in a column means that the corresponding approaches are significantly better than the third one yet not significantly better than each other.

In the token-wise evaluation, the MLP and RF classifier surpass the ABM1 on the *News** and *Web* datasets. On the *News** dataset, MLP surpasses RF for LOCATION but RF surpasses MLP for PERSON. On the *Web* dataset, RF is better than MLP for LOCATION but not significantly different from one another for PERSON. Also, for the ORGANIZATION class, no significant difference could be determined on both datasets. On the *Reuters* dataset, MLP and RF are better than ABM1 for LOCATION and ORGANIZATION, but do not differ one another. For the entity type PERSON, no significant difference could be determined for all three classifiers. On the *News* and *All* dataset, RF is significantly best for LOCATION. RF and ABM1 surpass the MLP for ORGANIZATION but are not significantly different. For the entity type PERSON, ABM1 is significantly best on the *News* dataset and RF is best on the *All* dataset.

The entity-level results also suggest shifts amongst the best systems depending on the datasets. Interestingly, MLP and ABM1 are the only two classes of algorithm that appear as top algorithms in both evaluation schemes.

Consequently, our results suggest that while the four approaches RF, MLP, ABM1 and J48 perform best over the datasets at hand, MLP and ABM1 are to be favored. Note that significant differences can be observed across the different datasets and that all four paradigms RF, MLP, ABM1 and J48 should be considered when applying ensemble learning to NER.

Figures 3.5 to 3.6 present the micro F-measures ($F\text{-}score_{\text{mic}}$) of the NER task computed with GERBIL's benchmarks[7] for strong and weak entity matching for several systems. The results of our system, FOX, on several datasets are highlighted in these diagrams, and the results were achieved by using our on the *News** dataset trained MLP version. We picked this setup as a result of our previous experiments. The overall results of our previous experiments suggest MLP and ABM1, and we reached with the MLP approach the highest *F-score* on the *News** dataset. Thus, we chose the MLP approach with the *News** dataset for training our ensemble learning model that we provide in our public demo application.

7 https://gerbil.aksw.org/gerbil/overview (Accessed: 2022-08-13)

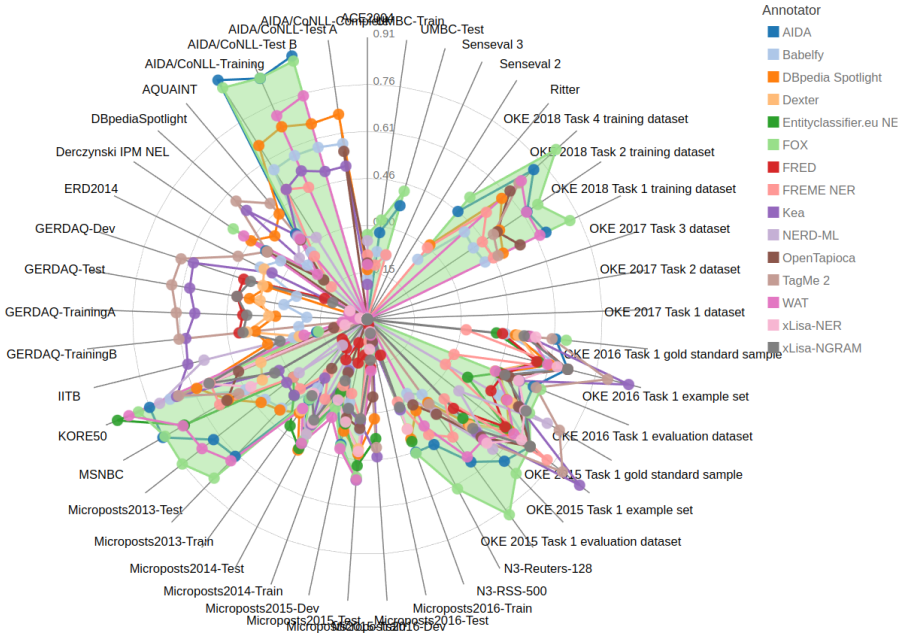


Figure 3.5: Spider diagram of experiment results with strong annotation matching derived from GERBIL's online interface.

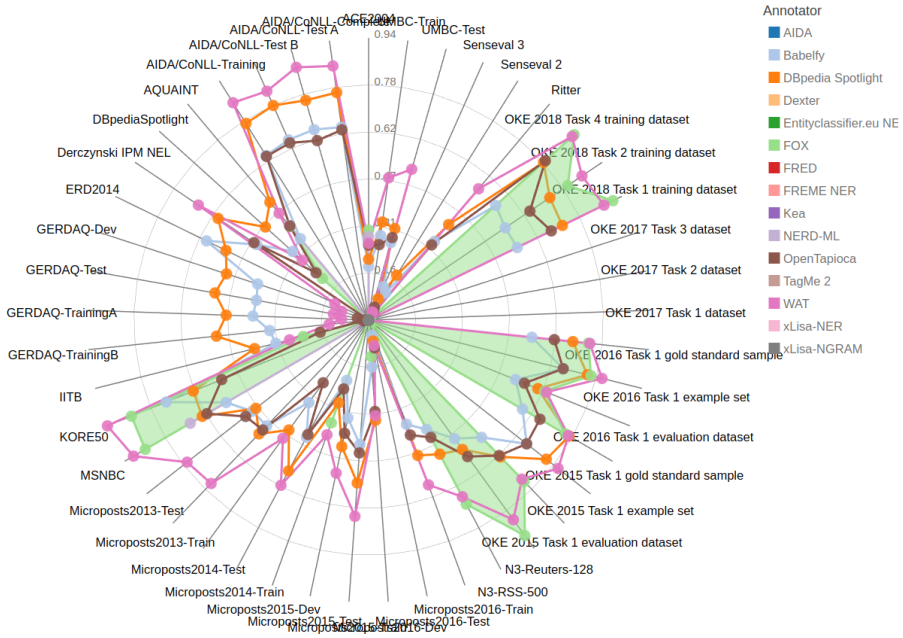


Figure 3.6: Spider diagram of experiment results with weak annotation matching derived from GERBIL's online interface.

4

ENTITY TYPE EXTRACTION AND LINKING FROM UNSTRUCTURED TEXT

In this chapter[1], we present Cetus, a novel pattern-based entity type extraction approach for identifying types of given entities inside a given text source and linking this types to a Knowledge Graph (KG).

4.1 RELATED WORK

Several tools have been introduced with the ability to type Named Entities (NEs), e.g., our approach Fox [259] presented in Chapter 3. However, these tools differ in two major aspects compared to Cetus. First, existing tools contain only very general type hierarchies while our approach generates novel and fine grained classes, see Section 4.3. Second, Cetus is the first tool to mark the part of a given document that contains the type evidence, i.e., a string indicating the choosen type. Thus, there is a clear difference between the entity typing and the type extraction tasks. To the best of our knowledge, Cetus is the first approach to tackle the type extraction task.

Our approach is mainly based on patterns and is inspired by Hearst Patterns [105]. Those patterns match text parts describing hyponym relations between two nouns. In difference to our patterns, the Hearst Patterns have been extracted from a large corpus using a bootstrapping approach. As described in Section 4.2, our patterns are defined for matching text parts describing the type relation of a given entity and have been created manually during an iterative, incremental process.

4.2 PATTERN EXTRACTION

The patterns for identifying the type of an entity inside a text document corpus are generated semi-automatically in an iterative manner. First, Cetus identifies phrases containing entities and their types in a given text source and extracts them. Here we use the DBpedia 2014 abstracts. After sorting these extracted phrases according to the string in between the entity and its type, we analyze them and create the

¶ Parts of this chapter have been published as conference articles [206, 240]. The thesis author co-developed the design of the solution, source code and co-wrote the publication [240] together with the main author Michael Röder.

1 Throughout this chapter, we use the prefix `rdf`, `rdfs`, `yago`, `dul`, `scmsann` and `ex` for the Internationalized Resource Identifiers (IRIs) `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, `http://yago-knowledge.org/resource/`, `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#`, `http://ns.aksw.org/scms/annotations/` and `http://example.com/`.

Table 4.1: Examples of sentence parts found between an entity and its type.

Extracted phrase	Count
`<entity> is_be_vbz a_a_dt <type>`	242 806
`<type> <entity>`	107 082
`<entity> is_be_vbz an_an_dt <type>`	12 981
`<entity> is_be_vbz a_a_dt species_species_n1 of_of_pp-f <type>`	12 554
`<entity> is_be_vbz a_a_dt species_species_n1 of_of_pp-f flowering_flower_j-vvg <type>`	4 069

patterns in an incremental process. The progress of our pattern extraction is measured by the amount of phrases that are covered by our patterns. In the following, we described these steps in more detail.

4.2.1 *Sentence Part Extraction*

For extracting phrases containing entities and their types, abstracts of the English DBpedia 2014 dump file are applied. Every abstract describes an entity it belongs to and, thus, contains a label of the entity and its type. Thus, we assume that abstracts are written properly and contain both information.

First, each abstract is preprocessed individually by removing the text written in brackets, e. g., pronunciations. Afterwards, a deterministic coreference resolution system [149] replaces pronouns with their coreferenced words, e. g., *He studied physics* with *Albert Einstein studied physics*. The last step of the preprocessing is the splitting of abstracts into single sentences.

Second, sentences containing Entity Mentions (EMs) and at least one label of one of its types (`rdf:type`) are processed further. Cetus extracts the part of the sentence between the EM and the type label. Additionally, words, their lemmas and part-of-speech tags of extracted phrases are stored.

After analyzing all abstracts, Cetus counts all different phrases. Table 4.1 shows examples of extracted phrases and how often they have been found inside the English DBpedia. The words inside these parts are encoded as `<word>_<lemma>_<pos-tag>`.

Delving into the extracted phrases reveals insights into the structure of entity type descriptions in DBpedia abstracts. It can be seen that the formulation "*<entity> is a <type>*" occurs most often. The second most common formulation uses a type preceding the entity and is listed as the second example in Table 4.1. The third example is a variant of the first but containing the determine "*an*" instead of "*a*". The fourth example shows that some abstracts contain more complex formulations like "*<entity> is a <type> of <type>*" while the last

```
type : (ADJECTIVE|VERB|ADVERB)* FOREIGN? NOUN+;
```

Listing 4.1: The grammar rule defining a type surface form.

```
is_a_pattern : ENTITY is_is_vbz a_a_dt type;
```

Listing 4.2: First simple version of the *is-a* pattern. `ENTITY` is a marking for the entities position.

example contains an additional adjective that was not a part of the types label, i. e., "*flowering*".

4.2.2 *Grammar Construction*

The aim of creating a grammar is to generate a parser that is able to identify the part of a sentence describing an entities type given the position of the entity inside the sentence. For generating a parser based on our grammar, we are using the ANTLR4 library [214].

Our grammar is based on the following assumptions:

1. A sentence contains an entity and a type. Otherwise the sentence is not part of our grammar language.
2. A type must contain a noun, but can contain additional words that are specifying the meaning of the noun, e. g., adjectives.

The first assumption simplifies the task of defining a grammar since we can focus on the sentences that are important for our task and ignore all others. The second assumption contains the definition of a type surface form, i. e., a string that consists of a noun and optionally additional words. It might seem to be contradictory w.r.t. the last example of Table 4.1 but for the extraction it is important that we extract all words that could be part of the types surface form. Following this assumptions, we can define a type inside the grammar with the rule in Listing 4.1.

A surface form of a type can contain a number of adjectives, verbs or adverbs as well as a foreign word, e. g., the latin word "*sub*". Additionally, a type has one or more nouns.

As mentioned above, the construction of the grammar is designed to be an iterative, incremental, self-improving process. We start with the simple *is-a* pattern that matches the most common phrase "`<entity>` *is a* `<type>`". The definition of this pattern is shown in Listing 4.2.

With this simple grammar, we try to match all phrases extracted beforehand and create a list containing all those phrases that have not been matched so far. Using this list, we extend our grammar to match other phrases. In our example, we extend the simple *is-a*

```
is_a_pattern : ENTITY FORM_OF_BE DETERMINER type_with_dt;

FORM_OF_BE : ~[ \t\r\n]+ '_be_v' ~[ \t\r\n]*;
DETERMINER : ~[ \t\r\n]+ '_' ~[ \t\r\n]+ '_d' ~[ \t\r\n]*;
```

Listing 4.3: Extended version of the *is-a* pattern.

pattern towards matching different temporal forms of the verb "*be*" and different determiners, e. g., "*a*" and "*an*", see Listing 4.3.

With this iterative, incremental process, we further extended the grammar until we covered more than 90% of the extracted phrases. The complete grammar can be found in the projects source code repository.

4.3 TYPE EXTRACTION

The pattern-based type extraction can be separated into two steps: (i) Extracting type evidence strings from the text, i. e., surface forms, and (ii) creating a type hierarchy based on the extracted surface forms.

In the following, we provide more details of each of the two steps.

4.3.1 *Type String Extraction*

To identify the type evidence surface form for a certain entity, CETUS extracts the string containing the type of a given entity in a given text using our grammar (see Section 4.2.2).

Let us assume the following running example document:

"*In 1921, **Albert Einstein** got the Nobel Prize in Physics. He was a German-born theoretical physicist.*"

CETUS processes this document as input with "*Albert Einstein*" marked as EM. First, the deterministic coreference resolution system is applied to replace the pronoun of the second sentence by "*Albert Einstein*".

"*In 1921, **Albert Einstein** got the Nobel Prize in Physics. **Albert Einstein** was a German-born theoretical physicist.*"

After that, the text is split into sentences and the EM is replaced by a placeholder, i. e., "*ENTITY*".

"*In 1921, **ENTITY** got the Nobel Prize in Physics.*"
"***ENTITY** was a German-born theoretical physicist.*"

A parser based on the grammar from Section 4.2.2 is applied to every sentence in the document. While the first sentence is identified as not contained in the language of our grammar, the second sentence is identified to be in the language. Moreover, the parser identifies "*German-born theoretical physicist*" as evidence entity type surface form.

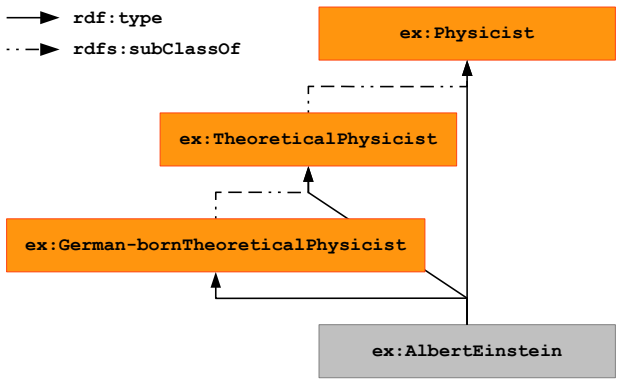


Figure 4.1: Schema of the generated local type hierarchy of the running example document.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex: <http://example.com/> .

ex:AlbertEinstein
   a ex:German-bornTheoreticalPhysicist,
      ex:TheoreticalPhysicist, ex:Physicist .

ex:German-bornTheoreticalPhysicist
   a rdfs:Class ;
   rdfs:subClassOf ex:TheoreticalPhysicist ;
   rdfs:label "German-born theoretical physicist" .

ex:TheoreticalPhysicist
   a rdfs:Class ;
   rdfs:subClassOf ex:Physicist ;
   rdfs:label "theoretical physicist" .

ex:Physicist
   a rdfs:Class ;
   rdfs:label "physicist" .
```

Listing 4.4: The local type hierarchy that is generated from the extracted surface form expressed using RDF/TURTLE serialization.

4.3.2 *Local Type Hierarchy*

Based on the extracted surface form as evidence for the entity type, CETUS creates a type hierarchy and links the given entity to the hierarchy. The type hierarchy comprises types that are generated automatically from the extracted surface form based on the second assumption of Section 4.2.2. Each type is generated by concatenating the words found in the extracted string using camel case. After a type has been created, the first word is removed and the next type is created. Every following type is a super type of the types generated before. Finally, the entity is connected to all generated types.

For our example, three types would be generated and linked to the entity as shown in Figure 4.1 and Listing 4.4.

Table 4.2: Mapping from YAGO to DOLCE+DnS Ultra Lite classes.

YAGO class	DOLCE+DnS Ultra Lite class
yago:wordnet_person_100007846	dul:Person
yago:wordnet_location_100027167	dul:Place
yago:wordnet_organization_108008335	dul:Organization
yago:wordnet_role_100722061	dul:Role

4.4 ENTITY TYPE LINKING USING YAGO

The linking of the generated types to a KG can be done in two different ways. Our first approach, CETUS$_{YAGO}$, uses the labels of the automatically generated types to find a matching type inside another well-known KG. CETUS uses the YAGO ontology [163] which comprises a large class hierarchy and, thus, increases the chance to match one of these classes to one of the labels of the automatically generated type. YAGO itself contains more than 10 mio. entities and exceeds 350.000 classes. Our second approach serves as a baseline to our own baseline approach and uses the FOX [259] framework.

First, an index containing the surface forms of the YAGO classes with a mapping to the class Uniform Resource Identifier [17]s (URIs) is created. Second, CETUS needs to match every generated type using the approach from Section 4.3 to one of the labels of the YAGO classes.

Currently, our approach uses 3-gram string similarity to match labels in the index with those of the generated types as it has been proven to be efficient and effective for such a task [277]. This process retrieves the most similar YAGO classes and similarity scores for every generated type. From these YAGO classes and scores, CETUS chooses the class with the highest similarity score. If two classes have the same score, CETUS chooses the class which is lower inside the local generated type hierarchy. The chosen YAGO class is linked to its local type.

After that, we iterate through the YAGO class hierarchy from the linked class to its root, searching for one of the classes listed in Table 4.2.

If such a class is found, we link it with the corresponding DOLCE+DnS Ultra Lite class. Otherwise, we repeat the search using the YAGO class with the second highest similarity score. The result for our running example can be seen in Figure 4.2.

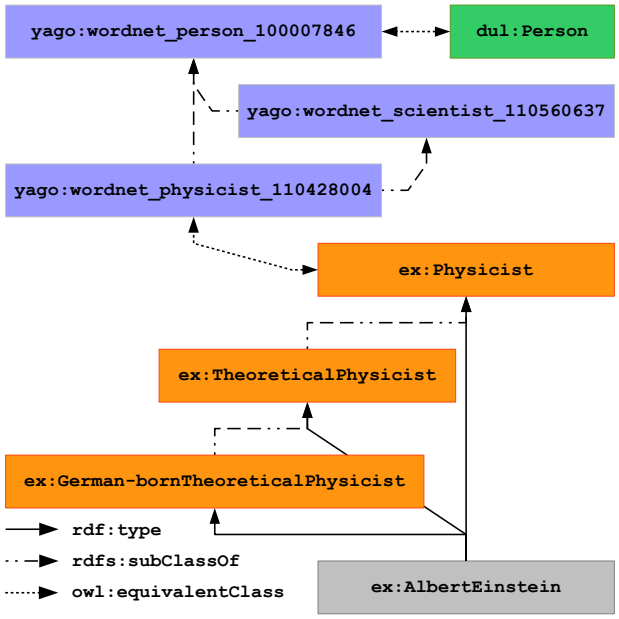


Figure 4.2: Resulting type hierarchy that is created based on the YAGO ontology.

4.5 ENTITY TYPE LINKING USING FOX

A second approach for a type extraction baseline is the usage of one of the various, existing entity typing tools. For our second version CETUS$_{FOX}$, we are using Fox [259–261], a Named Entity Recognition (NER) and typing tool based on ensemble learning over different tools with diverse strengths and weaknesses described in Chapter 3.

CETUS$_{FOX}$ sends a given document to the Fox web-service[2] for retrieving annotations. If an entity inside a document is found and typed by Fox, the type is used to choose one of the four DOLCE+DnS Ultra Lite classes, see Table 4.3. The chosen class is used as super class for the automatically created types. Unfortunately, Fox does not identify roles as entity types in its current version.

Table 4.3: Mapping from Fox classes to DOLCE+DnS Ultra Lite classes.

Fox class	DOLCE+DnS Ultra Lite class
`scmsann:PERSON`	`dul:Person`
`scmsann:LOCATION`	`dul:Place`
`scmsann:ORGANIZATION`	`dul:Organization`

With respect to our running example, the Fox tool marks "*Albert Einstein*" as a person, i. e., with the type $\tau =$`PERSON`. Thus, the cre-

2 https://dice-research.org/FOX

ated types would be defined as subclasses of `dul:Person` as shown in Figure 4.3.

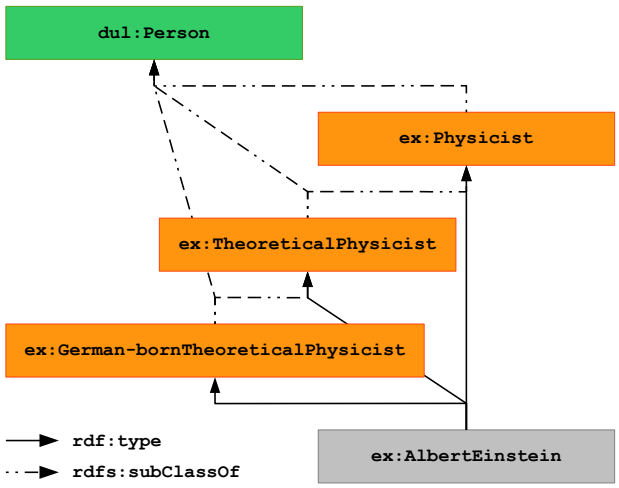


Figure 4.3: Resulting type hierarchy that is created based on the results of Fox.

4.6 EVALUATION

We participated with CETUS as well as FOX—as an off-the-shelf system—in both Open Knowledge Extraction (OKE) Challenge tasks at the International Semantic Web Conference 2015. FOX and two other tools—ADEL [125] and FRED [40]—participated in the first task while CETUS and two other tools—FRED [40] and OAK [87]—participated in the second task of the OKE Challenge 2015. The dataset of the first task used for the evaluation contains 101 documents and 99 documents for the evaluation of the second task. For evaluating the different systems, a local modified version of GERBIL [279] has been used.

4.6.1 *OKE Challenge 2015 Task 1*

First, we employed the off-the-shelf framework FOX to show that FOX is able to identify the relevant DOLCE types. The evaluation results of the first task are shown in Table 4.4 and the sub tasks for FOX are depicted in Table 4.5.

In the entity recognition sub task, FOX performs well regarding Precision (with $Pr_{\text{mic}} \sim 0.96$ and $Pr_{\text{mac}} \sim 0.92$) and reaches nearly the Recall of the best system ADEL. Unfortunately, FOX supports only three of the four entity types in the OKE challenge in its current version. Thus, the Recall and consequently the F-measure for entity linking and typing are low. We assume that the lack of supported entity types leads to FOX' inability to reach the best performance in the OKE Challenge 2015 task 1.

Table 4.4: Results of the OKE Challenge 2015 task 1

System	Micro			Macro		
	$F\text{-}score_{mic}$	Pr_{mic}	Re_{mic}	$F\text{-}score_{mac}$	Pr_{mac}	Re_{mac}
ADEL	**0.61**	**0.69**	**0.54**	**0.60**	**0.69**	**0.54**
FOX	0.50	0.66	0.41	0.48	0.63	0.41
FRED	0.35	0.47	0.28	0.23	0.31	0.18

Table 4.5: Results for the different sub tasks of task 1

System	Micro			Macro		
	$F\text{-}score_{mic}$	Pr_{mic}	Re_{mic}	$F\text{-}score_{mac}$	Pr_{mac}	Re_{mac}
Fox (Recognition)	0.68	0.96	0.52	0.65	0.92	0.53
Fox (Linking)	0.50	0.70	0.38	0.46	0.65	0.38
Fox (Typing)	0.35	0.35	0.35	0.37	0.37	0.37

4.6.2 *OKE Challenge 2015 Task 2*

The official results contained only the results of CETUS $_{YAGO}$[3]. Thus, we set up an instance of GERBIL and repeated the evaluation for both versions of CETUS. The results can be seen in Table 4.6. The tables show that both versions of CETUS outperform the other participants regarding the F-measure.

Table 4.6: Results of the OKE Challenge 2015 task 2

System	Micro			Macro		
	$F\text{-}score_{mic}$	Pr_{mic}	Re_{mic}	$F\text{-}score_{mac}$	Pr_{mac}	Re_{mac}
CETUS $_{YAGO}$	**0.47**	0.45	**0.52**	**0.45**	0.42	**0.53**
CETUS $_{FOX}$	0.46	0.45	0.46	0.44	**0.42**	0.47
OAK	0.44	**0.52**	0.39	0.39	0.40	0.40
FRED	0.30	0.29	0.32	0.27	0.26	0.32

Table 4.7 shows the detailed results of the two steps of CETUS. It can be seen, that the pattern-based recognition of the string containing the type of an entity performs well with $F\text{-}score_{mic}$ of ~ 0.7. However, there is still space for improvement. A large problem for this approach are formulations that have a different grammatical structure than those inside the DBPEDIA abstracts. Thus, a system with a better

3 The results of the challenge can be found at `https://github.com/anuzzolese/oke-challenge#results`.

Table 4.7: Results for the different sub tasks of task 2

System	Micro			Macro		
	$F\text{-}score_{\text{mic}}$	Pr_{mic}	Re_{mic}	$F\text{-}score_{\text{mac}}$	Pr_{mac}	Re_{mac}
CETUS (Type Recognition)	0.70	0.69	0.70	0.66	0.64	0.72
CETUS $_{YAGO}$ (Type Linking)	0.25	0.20	0.34	0.23	0.20	0.34
CETUS $_{FOX}$ (Type Linking)	0.22	0.21	0.22	0.22	0.21	0.22

understanding of the internal structure of the sentence, e. g., by using parse trees, could avoid these problems.

Comparing both type linking approaches, it can be seen that both have a similar Precision (see Table 4.7). But the YAGO-based approach has a higher Recall leading to a slightly higher F-measure. The Fox-based type linking lacks the identification of types different to `PERSON`, `ORGANIZATION` and `LOCATION`. The YAGO-based type linking suffers from two main problems. First, some of the extracted local types cannot be matched to YAGO types. This might be solved by using a better search strategy for finding YAGO types with a similar label, e. g., trigram similarity. The second point of failure is the mapping from YAGO to DOLCE types. For some YAGO types there are no linked DOLCE types while for others the linked DOLCE types are very high inside the hierarchy leading to a coarse typing result and, thus, to a lower precision. A further improvement of the mapping between YAGO and DOLCE types could reduce these problems.

Figures 4.4 to 4.5 depict the micro F-measure ($F\text{-}score_{\text{mic}}$) of the challenge task 1 computed with GERBIL's benchmarks for strong and weak entity matching.

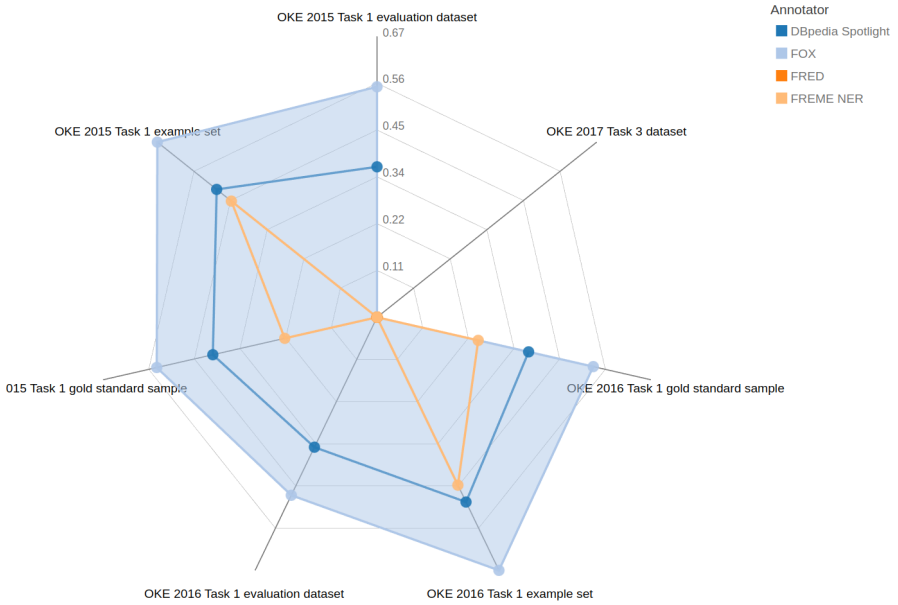


Figure 4.4: Micro F-measure ($F\text{-}score_{\text{mic}}$) of task 1 with GERBIL's strong entity matching benchmark.

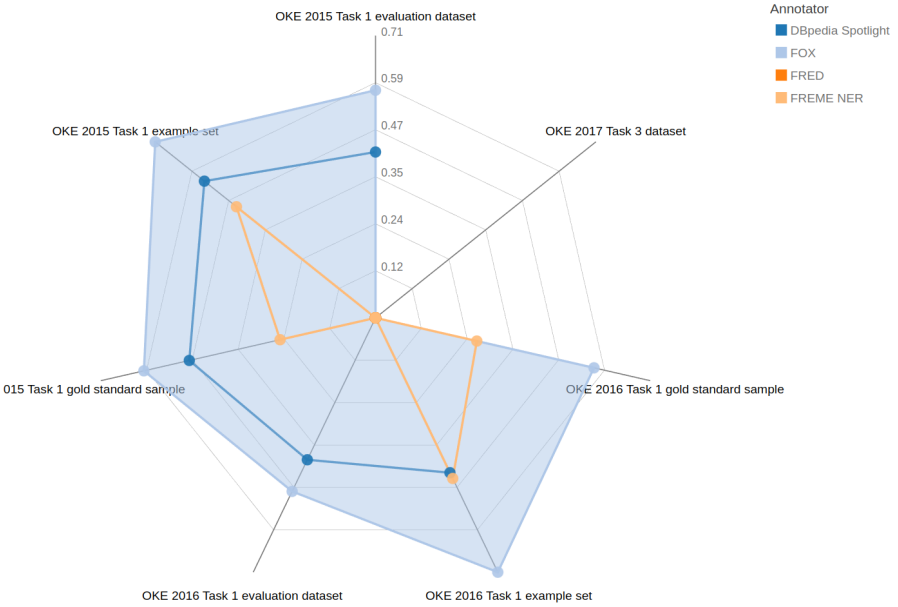


Figure 4.5: Micro F-measure ($F\text{-}score_{\text{mic}}$) of task 1 with GERBIL's weak entity matching benchmark.

5

TOWARDS HOLISTIC ENTITY LINKING: SURVEY AND DIRECTIONS

Recent surveys about Entity Linking (EL) present a good overview of existing approaches, datasets, benchmarks [248, 291], and EL approaches for a specific type of text document, such as microblog posts [51]. However, these surveys do not consider in detail emerging strategies that influence new EL approaches and can be regarded as ways to consider several facets of the EL task concomitantly in more holistic EL processes. Thus, we present a more detailed literature review and analysis of emerging holistic approaches for the EL task. Such a holistic view may provide extra information for EL approaches to tackle ambiguous Entity Mentions (EMs), for example, in texts with limited context and lots of noise (e. g., typos, grammatical errors, extensive use of slangs, acronyms) like social media posts.

5.1 NAMED ENTITY LINKING AND DISAMBIGUATION

The EL task links each relevant EM, for instance "*Jordan*", found in a sentence of a text corpus (i. e., *text sources*) to a descriptor of what that EM refers to in the context where it appears. The entity descriptors can be taken, for instance, from a Knowledge Graph (KG) (e. g., DBpedia[1] [9, 150], YAGO[2] [67], Freebase[3] [19], WikiData[4] [280]). For example, the EM "*Jordan*" may refer to the DBpedia entity descriptor `dbr:Michael_Jordan` of the basketball player *Michael Jeffrey Jordan* or to `dbr:Jordan` the country *Hashemite Kingdom of Jordan*.

Commonly, the EMs are recognized one step before by the Named Entity Recognition (NER) task (see Section 3.2.1), which is responsible for identifying and classifying these EMs with their respective types, for instance, τ=`PERSON`.

Some works consider that EL is a combination of the NER and Named Entity Disambiguation (NED) tasks [36, 129, 303], while others consider that EL is just the disambiguation of the mentions (also called NED task) [248, 273]. In this work, we consider the latter definition for EL where the EMs are recognized one step before by the NER task.

¶ Parts of this chapter have been published as journal article [207]. The thesis author co-developed the design of the solution and co-wrote the publication together with the main author.

1 https://wiki.dbpedia.org

2 http://www.yago-knowledge.org/

3 https://developers.google.com/freebase/

4 https://www.wikidata.org

5.2 ENTITY LINKING

Given[5] [248] a set of entities $\mathcal{E} = \{\varepsilon_1, \varepsilon_2, \ldots\}$ (e. g., within a KG, see Section 2.2) and for each sentence $\mathfrak{s} = \langle \omega_1, \omega_2, \ldots \rangle$ in a given text corpus $\mathscr{C}$ a set of EMs $\mathscr{M}_\mathfrak{s} = \{m_1, m_2, \ldots\} \subseteq 2^\mathfrak{s}$ expressing entities. The EMs are given by $m_1 = \langle \omega_i \rangle_{i=k,\ldots,l}$ and $m_2 = \langle \omega_j \rangle_{j=m,\ldots,n}$ with $1 \leq k \leq l \leq m \leq n$.

The EL task aims to map each EM $m \in \mathscr{M}_\mathfrak{s}$ in a sentence $\mathfrak{s}$ to its corresponding entity $\varepsilon \in \mathcal{E}$. In case that the corresponding entity ε for a mention m does not exist in $\mathcal{E}$ (i. e., $\varepsilon \notin \mathcal{E}$), m is labeled as "*NIL*", whose meaning is that it cannot be linked, i. e., it is not provided in $\mathcal{E}$.

5.3 EMERGING HOLISTIC APPROACHES

According to Cambridge Dictionary[6], holism is "the belief that each thing is a whole that is more important than the parts that make it up". From this concept, we understand that holism in EL involves several kinds of input (e. g., text documents to be annotated, their associated data and metadata, KGs), a variety of relevant features that can be extracted from these inputs, and a myriad of methods that can be employed in the data processing for EL. Although several works have been employing some degree of holism in the EL task, as detailed in Section 5.4, to the best of our knowledge, they were never analyzed and summarized from this viewpoint.

This work aims to provide a comprehensive review of a variety of EL approaches that exhibit some holism. We classify these approaches according to key aspects that allow an overview of holistic techniques applied to EL and a better understanding of their diversity. These key holistic aspects include the exploitation of distinct inputs and data features, the use of diverse Natural Language Processing (NLP) tasks for information extraction and, the collective disambiguation of mentions on text and knowledge models, such as embeddings. To the best of our knowledge, these aspects of holistic EL approaches have not been described in the literature yet. They are usually implicit in the EL proposals. They can be useful to understand better, classify, and compare these approaches. This work gives insights into how a variety of techniques can be combined into more holistic approaches to improve EL results.

Most of the EL approaches from the literature still rely directly on WIKIPEDIA to determine entities, and many of them employ well-known NLP tasks. On the other hand, there is a recent trend to use embeddings. Meanwhile, there are also efforts to boost EL power and reliability by considering semantic coherence of entities in collective EL processes. Based on these findings, we propose some pillars for future

5 We follow the definition of Shen et al. (2015) [248] for the EL task.

6 https://dictionary.cambridge.org/dictionary/english/holism

holistic EL approaches, that include: (i) handling EL (and semantic annotation) as a general process which can be tailored for different kinds of data (e. g., news, social media posts) and goals, by appropriately selecting, composing and tuning suitable approaches for each one of its constituent tasks; (ii) better-exploiting word embeddings aligned with knowledge embeddings for EL; (iii) using context information extracted from texts and their associated data and metadata (e. g., source, location, time) to disambiguate collections of related mentions holistically (e. g., in the same document or the documents of the same author, geographically or historically related) based on measures of the semantic coherence of the entity candidates. We also outline a holistic EL approach that exploits these pillars to disambiguate more EMs more accurately.

5.3.1 *Bibliographical Review Procedure*

The steps of the methodology employed in the systematic bibliographical review are presented in Figure 5.1. The search for papers was done in Google Scholar[7], the ACM Digital Library[8], Springer Link[9] and, Scopus[10]. We chose these platforms because they gave the best results for preliminary searches, including books, conference, proceedings and journals papers. The search string used was *"(Entity Linking"* OR *"Named Entity Disambiguation")* AND *"text document"*.

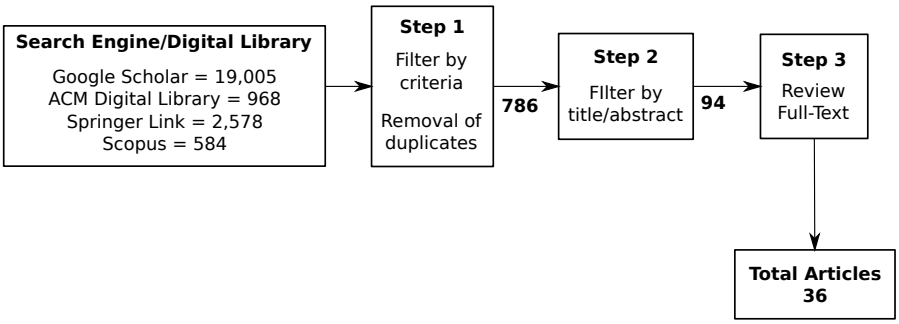


Figure 5.1: Steps followed to retrieve the articles analyzed in this work.

Although the focus of this work is holistic approaches for EL. However, most articles include not the word *holistic* in their title or contents, justifying its exclusion from the keyword search. Moreover, we have not considered articles tagged as *preview-only* in Springer Link.

In total, we found 23135 articles. To reduce the number of articles (Step 1), we removed duplicates using the Mendeley[11] tool. Then, we considered only articles that satisfied the following criteria: (i) *peer-*

7 https://scholar.google.com.br/
8 https://dl.acm.org/
9 https://link.springer.com/
10 https://www.scopus.com/
11 https://www.mendeley.com/

reviewed or *published*; (ii) published from 2005 onwards; (iii) written in English; (iv) EL approaches that annotate textual data. Regarding criteria (ii), we have found only a few articles about EL before 2005 and, most of them, adopt a manual or semi-automatic method. In this work, we focus only on fully automatic EL approaches. Step 1 yielded 786 articles.

In Step 2, we manually analyzed the title and abstract of the 786 articles resulting from Step 1 and removed those not related to the EL task, resulting in 84 articles. Lastly, in Step 3, we reviewed the remaining 94 articles and selected the 36 ones having any of the holistic aspects presented in Section 5.4.

5.4 HOLISTIC ENTITY LINKING

The goal of this section is to draw how holism can manifest and contribute to better results in EL approaches. First, in Section 5.4.1, we exemplify the potential benefits of holism in EL, in a real-world scenario where EMs must be disambiguated in microblog posts. Then, in Section 5.4.2, we delineates key aspects of holistic EL approaches that we have derived from our studies and that establish essential criteria to identify, classify and analyze holistic EL approaches.

5.4.1 *Motivating Example*

Figure 5.2 illustrates how a holistic view can be helpful for EL on real microblog posts by considering several data features, methods, and semantic coherence to disambiguate EMs. Assume that different methods and tools annotated the two tweets presented on the top of the figure, using the information resources represented as labeled boxes. The green dashed lines represent the links from each mention to the respective resource used to describe it correctly, while red dotted lines represent links to resource descriptions that refer to incorrect disambiguations of mentions. The blue and yellow boxes refer to, respectively, concepts and entities from a KG. The gray boxes are word senses described in a lexical base. The resource description corresponding to the mention "*uncountable*" is not shown in the figure just for simplicity.

Different data features are considered by different NLP tasks and annotation approaches for the tweet on the left. Geographic coordinates and the indication of a place in the tweet metadata can also be used to determine the PoI from where the tweet was sent, which is, in this case, the city of "*New York*". This, allied with the annotation of the mention "*NBA*" (National Basketball Association), provides evidence to disambiguate the ambiguous mention "*Jordan*" to its correct description, i. e., the basketball player *Michael Jeffrey Jordan* to

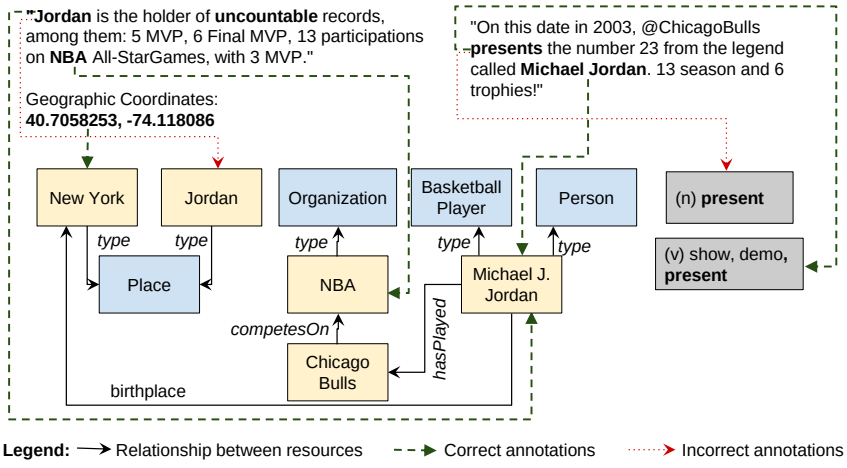


Figure 5.2: Collaborative disambiguation of EMs based on coherence.

`dbr:Michael_Jordan`, instead of the country *Hashemite Kingdom of Jordan* to `dbr:Jordan`.

Then, if the tweet on the left is related to the one on the right (e. g., both come from the same user around the same time), we can use the annotations generated for the former to help the EL task on the latter (and vice-versa). Therefore, the previous annotations are used as what we can call a *semantic context* to disambiguate the mention "*Michael Jordan*" successfully to its correct description, based on semantic coherence. However, though the tweet on the right has the mention "*@ChicagoBulls*" (a basketball team where *Michael Jeffrey Jordan* played), which could also help to assemble the semantic context. However, most tools that we have tried could not identify this mention. It occurs due to the use of the special character "@" and the lack of spaces to separate the words constituting the mention "*@ChicagoBulls*". Noise data like this, misspellings, and grammar errors can hinder the performance of current EL approaches. More holistic strategies can help face such challenges by bringing more contextual information that can be useful in the EL process.

5.4.2 *Key Aspects of Holistic Entity Linking*

The previous example evidences some key aspects of holistic EL approaches that we have identified in our studies. They include the exploitation of distinct inputs and data features, the use of diverse NLP tasks. However, these aspects of holistic EL approaches have not been adequately described in the literature yet. They are usually implicit in the EL proposals. We believe that they can be useful to understand better, classify, and compare these approaches, besides giving insights about promising research directions. Hence, in this work, we define these key aspects of holistic EL approaches as follows.

DISTINCT INPUTS AND DATA FEATURES: Different EL approaches can have distinct inputs. Some approaches are tailored to annotate some specific kinds of text documents (e. g., microblog posts) and may also exploit data and metadata associated with them (e. g., the location of the user posting a tweet, the tweet timestamp). Besides, alternative external inputs can be used to provide entity descriptions and further information/knowledge to help EL. Most current EL approaches employ particular features extracted from a limited number of inputs. Nevertheless, synergetic exploitation of distinct data and data features can boost EL results, especially if context information is limited, as in some microblog posts.

DIVERSE NLP TASKS: The EL task is usually preceded by the NER task, which can rely on other NLP tasks and tools to identify mentions in the text. Moreover, several other NLP tasks (e. g., Word Sense Disambiguation (WSD), Entity Saliency (ES)) can be used before, after, or concomitantly with the EL task to improve its results.

DISMABIGUATION METHODS: The disambiguation step of the EL task may employ a myriad of methods, that can be combined or not. Two promising directions are collective disambiguation and embedding-based methods. Collective disambiguation considers several EMs simultaneously and the coherence of entity candidates to drive the disambiguation process. Embedding-based methods use as inputs word, entity, or KG embeddings. Embeddings-based methods can exploit global relations in a more efficient way than traditional approaches like graph-based ones.

5.5 HOLISTIC ENTITY LINKING APPROACHES

This section reviews works that present holistic approaches to tackle the EL task. These works were collected in an extensive and systematic review, as explained in Section 5.3.1. We considered works that already exploit, explicitly or implicitly, any form of holism in their EL processes. Each subsection refers to one of the aspects of holistic EL described in Section 5.4. The works are organized among these subsections according to their premises, purposes, and proposed approaches. Some works comprise more than one aspect of holism. In such cases, the work is described in the section referring to the holistic aspect that we consider most exploited or relevant in that work.

5.5.1 *Distinct Inputs and Data Features*

EL can rely on a variety of features extracted from the text documents to be annotated, as well as their associated data and metadata. For example, the temporal context of microblog posts (e. g., tweets) [116, 271], their ordering, and their associated data. According to Hua et al. (2015) [116], EL approaches that focus just on the context around EMs are unsuitable for queries and tweets. This occurs because both queries and tweets are concise, and, therefore, present little context around the EMs. Therefore, the authors propose the use of other features besides that. More specifically, they propose the use of entity popularity and recency to determine user interests by his/her social interactions. Hua et al. (2015) [116] define entity recency as the recent popularity of a specific entity. Their user interests model captures the most relevant entities for the user. It is produced through the analysis of the user social interactions. The scores of the entity candidates during the EL task are obtained by summing the scores of entity popularity, entity recency, and user interests. The correct entity candidate for its respective mention is the one with the highest score.

Tran et al. (2015) [271] aim to annotate hashtags found in tweets, instead of EMs. To disambiguate hashtags, they propose the use of a ranking learning algorithm using temporal information. According to the authors, the meaning of a hashtag can change depending on the posting time. For example, the hashtag *#sochi* usually refers to the city of Sochi, in Russia. However, around 2014, that hashtag was extensively used to talk about the Winter Olympic Games that happened in that city in 2014. To perform such disambiguation, the authors use WIKIPEDIA pages, their edit history, and their page view statistics. Based on them, the proposed method builds a graph, called *influence graph*, having the entity candidates for the hashtags as nodes and their hyperlinks as edges. The edges and several similarity measures determine the influence of an entity candidate on the graph. This graph is used to train a ranking learning algorithm, which is similar to the PageRank algorithm. Then, the entity candidates with the biggest influence are chosen as the correct disambiguation for their respective hashtags.

5.5.2 *Diverse Natural Language Processing Tasks*

Since the EL task requires EMs to be previously spotted and demarcated in the text by the NER task, which can be done manually or automatically, most works consider these tasks independently, with NER completely preceding EL. However, some papers disagree with such an independent approach [138, 161, 166, 283], arguing that "mutual dependency between the two tasks is ignored" [161] and that "errors caused by NER will propagate to EL without the possibility of

recovery" [138]. Consequently, the EL task does not take advantage of the features extracted for the NER task, except the EMs, which are recognized and sometimes also classified. Based on this, Luo et al. 2015 [161] propose the Joint Entity Recognition and Linking (JERL) model. According to the authors, NER is usually defined as finding a sequence of EMs and labeling them (with their classes), while the EL task is defined as a ranking task. Thus, they consider that the biggest challenge to model the NER and EL tasks jointly is to combine the sequence labeling and ranking tasks. To achieve this, the JERL model extends a semi Conditional Random Field (CRF) model for modeling the entities distribution and "mutual dependency over-segmentation". To infer which entity candidate is the correct one to describe a EM, Luo et al. (2015) [161] extend the Viterbi algorithm [77].

Meanwhile, Wang & Iwaihara (2019) [283] and Martins et al. (2019) [166] propose a joint neural network model to tackle both tasks simultaneously. In Wang & Iwaihara (2018) [283], the proposed model is a deep neural network based on Tree recursive Neural Networks (TNNs) and Convolutional Neural Networks (CNNs). For the NER task, the authors use the system BRNN-CNN [156], a deep neural network model for NER that also uses TNNs. For the EL task, the authors propose a model that computes the semantic similarity between the recognized mentions and their respective entity candidates. This is achieved by comparing the representation of: (i) context (text around the Named Entity (NE) or introductory text of an entity), performed by TNNs with Long Short-Term Memory (LSTM) neural network; (ii) the whole document, performed by CNNs with an attention mechanism and; (iii) type of the NE and the entity candidate. Lastly, Wang & Iwaihara (2019) [283] combine the loss functions of the NER and the EL models to perform their joint training.

In Martins et al. (2019) [166], the authors extend the Stack-LSTM model [61], which has been used for the NER task [143], to tackle both tasks. They augment the Stack-LSTM with two bi-LSTM, whose inputs are word embeddings, to improve the NER task. The mentions, together with the entity embedding of the entity candidates of their respective mentions, are inputs for an affine transformation layer. The output of such a layer is the score for each entity candidate. The candidate with the highest score is selected as the correct one. As the approach of Kolitsas et al. (2018) [138] presents a jointly model for word and entity embeddings, it will be described in Section 5.5.3.2.

WSD is an NLP task similar to EL. Both tasks aim to annotate parts of a text with semantically well-described resources. What distinguishes these tasks are their slightly distinct purposes (namely, disambiguate EMs for EL and disambiguate word senses for WSD) and the different types of resources used for the semantic annotation (KGs for EL and lexicons like Wordnet for WSD). Most of the existing works in the literature handle these tasks separately. According to

Moro et al. (2014) [180], it leads to duplicated efforts, such as applying NLP tasks to extract features that are useful for the EL and WSD tasks. Besides, better results may be obtained by combining the contextual information manipulated by these tasks. Thus, they propose an approach that combines the EL and WSD tasks, aiming to improve the results generated by both tasks. This approach has three steps: (i) building the semantic signatures for vertices in a KG called BabelNet [194] that combines a variety of linked data with a multilingual version of the Wordnet; (ii) extracting all the relevant fragments from a given text (e. g., EMs, simple words); and (iii) disambiguating the entity and word sense candidates through the use of the semantic signatures. The semantic signature of a given vertex of Babelnet includes all the KG vertices that are densely connected to it and also among themselves [180]. According to the authors, in a KG, many concepts related to an entity or word sense are not directly connected to it. To avoid this issue, the edges of a KG are weighted using the concept of directed triangles.

The Random Walk with Restart algorithm [269] uses the weights from Moro et al. (2014) [180] to build the semantic signatures. The entities and word sense candidates are identified through the use of superstring matching. Moreover, text fragments can overlap. For example, consider the tweet "*After Game of Thrones and Star Wars episode 8, Dubrovnik as a host to Jame Foxx, Jame Dornan and Leo Di Capri in Robin Hood*". Their approach recognizes the text fragment "*Game of Thrones*" of this tweet as an entity. On the other hand, the words "*Game*" and "*Thrones*" are separetelly recognized as lexical resources. For the disambiguation of the entities and word sense candidates, the candidates are connected when the semantic signature of one has the other. After the connection of semantic signatures, a novel densest subgraph heuristic is applied to the resulting network of semantic signatures. The scores for the semantic signatures are compared, and the entities and lexical resources disambiguated.

Although less popular, there are several other (combinations of) semantic annotation tasks in the literature, for example, Entity Discovery (ED) [288] and ES [32, 272, 273]. The ED task tries to identify and group together EMs that refer to a same entity that does not exist in a KG [288]. Wick et al. (2013) [288] propose a joint model to tackle both EL and ED. For this, the authors propose the use of hierarchical trees built from EMs and their entity candidates. The leaves of the tree are the entity candidates. The internal nodes are summaries of the entity candidate features. The root is the EM. A temperature-regulated Markov Chain Monte Carlo [288] was used for disambiguation and discovery of new entities.

The ES task determines the relevance of the EMs according to with their importance to interpret the contents of a given text document. Usually, ES is a task performed after the EL task. However, some

papers argue that the combination of EL and ES can improve both tasks [32, 272, 273]. Since Chen et al. (2018) [32] present a bilinear model to joint learn word and entity embeddings, their approach is described in Section 5.5.3.2. On the other hand, in Trani et al. (2016) [272] and Trani et al. (2018) [273], the authors propose a Salient Entity Linking algorithm to perform the EL and ES tasks. Their proposal has two steps: Candidate Pruning and Saliency Linking. The Candidate Pruning step aims to reduce the number of entity candidates for each mention in a text document. For this, the authors use a supervised technique that classifies the entity candidates as relevant or irrelevant. The Saliency Linking step also uses a supervised technique, that predicts the entity candidates as *top relevant*, *highly relevant*, *partially relevant* and *not relevant*. One entity candidate will be considered as incorrect if its classification is *not relevant*. Therefore, the proposal [272, 273] addresses both the EL and the ES tasks simultaneously.

5.5.3 *Disambiguation Methods*

This section reviews EL proposals presenting approaches for collective disambiguation and some embedding-based approaches. We highlight that these two types of approaches can be combined to improve the EL task results further, as presented by works discussed in the following sections.

5.5.3.1 *Collective Disambiguation Methods*

Several works propose methods and algorithms for EL that disambiguate the EMs in a document separately, i. e., considering only the textual context around each mention. However, some proposals consider that the EMs in the same document or related documents are semantically related [35, 63, 70, 85, 96, 100, 119, 127, 157, 160, 215, 222, 229, 287, 295].

One way to consider EMs collectively in a document is through the exploitation of links between entities available in a KG (e. g., DBpedia, a KG specially built for EL on specific data). In Han et al. (2011) [100] and Guo & Barbosa (2014) [96], the EMs, entity candidates, and hyperlinks are modeled as a graph (called Referent Graph) in the former, while the same elements are modeled as a graphical model in the latter. In Han et al. (2011) [100] and Ganea et al. (2016) [85], evidence collected from the document to be annotated and from textual contents associated with the entity candidates are propagated to their respective representations. In Guo & Barbosa (2014) [96], otherwise, the authors exploit the links between entities to build graphs, called semantic signatures, of the entity candidates and the documents to be annotated. To disambiguate the candidates, after the propagation of the evidence, the candidates that achieve the best score [100] or have the highest probability [85] are chosen as the correct

entity representation for their respective EMs. Meanwhile, in Guo & Barbosa (2014) [96], the candidates whose semantic signature presents the highest similarity with the semantic signature of the document to be annotated are chosen.

Similarly to [85, 96, 100], Rama-Maneiro et al. (2020) [229] build a graph representing the links between entity candidates to disambiguate the EMs collectively. However, Rama-Maneiro et al. (2020) [229] exploit facts present in DBpedia to model the graph, exploiting some of its existing relations. The authors use the graph to calculate the degree of centrality to identify the most important node in the graph, i. e., the most relevant entity candidate. The authors avoid the combinatorial explosion when building the entity candidates graph by employing several strategies, such as using only the most relevant DBpedia relations, indexing paths between DBpedia nodes and, *"only considering entity candidates that are related to the topic of the document"*. The last strategy is achieved by an inference system that compares the context of an entity candidate (previously built from Wikipedia) with the text document. This way, the authors employ both topic coherence and node centrality in the disambiguation step. Lastly, the authors argue that their approach is capable of building entity candidate graphs up to 8 relations of the distance between entity candidates, while other approaches only achieve 2 or 3 relations of distance.

Although Wikipedia hyperlinks are useful to build graphs used in the disambiguation step of collective approaches, Vaigh et al. (2019) [63] argue that the lack of semantics of these hyperlinks hinders existing approaches. The authors propose the use of semantic relations present in KGs to improve collective EL approaches. Their approach employs a binary logistic regression classifier whose inputs are similarities between the word embedding representations of each mention with each one of its entity candidates, as local score, and, the semantic relatedness between entity candidates, as a global score. The semantic relatedness is calculated by considering the number of relations between entity candidates r_i and r_j divided by the total number of facts in which r_i takes part. To avoid a combinatorial explosion, the authors aggregate the semantic relatedness between entity candidates with a number of global features.

In the papers [85, 96, 100], the pairwise scores between entity candidates, including incorrect entities, are considered. According to Fang et al. (2019) [70] and Yang et al. (2019) [295], this strategy increases the complexity of the approaches and introduces noise in the results. Thus, both papers propose a sequential collective EL approach, in which they disambiguate the EMs considered less ambiguous and use them to help the disambiguation of the following and more ambiguous EMs. The approach proposed in Fang et al. (2019) [70] is based on an LSTM deep neural network with reinforcement learning with word and entity embedding as inputs. Meanwhile, the approach proposed

in Yang et al. (2019) [295] is based on an FFNN with reinforcement learning and attention mechanisms. The inputs are word embeddings representing words and EMs and, entity embeddings representing the already disambiguated EMs and entity candidates.

Similarly to the proposals of [70, 295], Phan et al. (2019) [222] also do not consider the pairwise scores between all pairs of entity candidates. However, instead of considering a sequential collective EL approach, Phan et al. (2019) [222] propose a collective approach by disambiguating the EMs in pairs, considering the pair with the highest confidence in each step. This way, their approach produces a minimum spanning tree of entity candidates that correctly disambiguate the EMs. Their approach is based in a graph whose vertices are the entity candidates, and edges connect entity candidates from distinctly EMs. Edge weights are used as a semantic distance score. The semantic distance score is based on local confidence and pairwise coherence score. The local confidence is calculated by Gradient Boost Tree, which uses the popularity of an entity candidate and the semantic similarity between the word embedding that surrounds the EM and the entity embedding that represents the entity candidate. The pairwise coherence score can be represented by Wikipedia Linked-Based measures [177], the logarithm of the Normalized Jaccard Similarity, and the cosine similarity between entity embeddings. To disambiguate the EMs, the authors employ a heuristic to find the minimum spanning tree. This heuristic is similar to the Kruskal algorithm [141], with the difference that, every time a pair of vertices is selected, the remaining vertices represent entity candidates for the same EMs are removed.

Both Huang et al. (2014) [119] and Chong et al. (2017) [35] also use the idea of handling the EMs collectively to annotate tweets semantically. However, due to a lack of context in tweets, such an approach does not work well considering single tweets. Therefore, both works consider a set of semantically related tweets to employ the collective approach. According to Huang et al. (2014) [119], it is challenging to create high quality labeled data to train supervised learning approaches for the EL task due to several factors, including missing and ambiguous resources in a KG and the difficulty to determine the prominence of the mentions in the text. Thus, the authors propose a graph-based semi-supervised learning approach to disambiguate mentions in tweets collectively. Their approach is based on three principles: (i) local compatibility between a mention and its candidate resources; (ii) coreference and (iii) semantic relatedness between different mentions. In Chong et al. (2017) [35], the authors consider that tweets posted geographically and timely close to each other can be semantically related. The proposed method builds a graph with tweets close to each other both in space and time. Moreover, the method considers that semantic relatedness among the entity candidates occurs

both inside each tweet (Intra-tweet coherence) and among different tweets (Inter-tweet coherence).

Although a single tweet provides little context information, Kalloubi et al. (2016) [127] proposes a collective approach for disambiguating the mentions in single tweets. It is justified because the focus of their work is to annotate tweets to retrieve the ones satisfying a query provided by a user. A weighted graph is built, with the entity candidates for all mentions spotted as nodes and the relationships between them as edges. The entity candidates are obtained from DBpedia. Their EL method identifies the most relevant entity in the graph for the respective tweet. This entity, called the central node, is used to calculate the weights of the remaining nodes. An analogous process is applied to the queries of the user. Then, the weighted graph of the query is compared with the weighted graphs of several tweets. The tweet graphs most similar to the user query graph are considered the most relevant to the user and retrieved.

The collective approaches to disambiguate mentions exploit the links between concepts and entities of KGs. However, some KGs for closed domains (e. g., medical, enterprise) have only a few links, if any, between their entities. Consequently, approaches that rely heavily on such links produce unsatisfactory results using these KGs [157]. To circumvent this problem, Li et al. (2016) [157] propose an approach that gathers evidence of EMs in the document to be annotated (e. g., TF-IDF score, relevant words around the EMs) and, with such evidence, produces a generative model that simulates the cross-document links among the entities in a KG. An extended version of the Gibbs Sampling is used to disambiguate the entities by inferring the entity that better describes each EM.

Different from the other collective approaches, Wei et al. (2019) [287] apply a candidate selection before the disambiguation step. The authors generate a graph with the entity candidates for each mention. Queries executed on Wikipedia and Freebase returns these candidates. The entity candidates are the nodes of the graphs, and they are connected when their respective Wikipedia pages are linked to each other. The authors apply PageRank to select the candidates in the entity candidate graph built for each mention. Only the candidates with higher ranks are considered in the disambiguation step, which is done by an FFNN (Feed-Forward Neural Network). The inputs of the FFNN are the word embeddings of the text; each entity candidate description encoded as a 128-dimension vector and the embedding of left contexts and right contexts (Dual) by Fixed-Size Ordinally Forgetting Encoding (FOFE) [302], i. e., Dual-FOFE. The use of the left and right context is also applied in the approach proposed by Liu et al. (2019) [160], which encode the entity embedding, based on the entity context and its description in Wikipedia pages. They employ these embeddings in their collective disambiguation approach. The entity

embedding generation is divided into two neural network models: (i) a Long Short-Term Memory (LSTM) model to encode the entity context, that is composed by the left and right context of the EM and; (ii) a Convolutional Neural Network (CNN) model for encoding the entity description. The embeddings are fed to a local model, based on a novel CRF attention network, which produces a local score for the entity candidates. These local scores are used in a Forward-Backward algorithm to calculate the global score and disambiguate all the mentions collectively.

Although collective approaches may provide better results than non-collective approaches, Parravicini et al. (2019) [215] argue that the existing EL approaches are not scalable enough for application/domains that present real-time requirements. They consider as real-time if an EL approach can recognize EMs and disambiguate them in a whole text in less than one second. Therefore, the authors propose a collective EL approach based on graph embedding and scalable for real-time requirements. For the disambiguation, the authors use cosine similarity to verify the similarities among the entity candidates of all EMs. However, this approach is impracticable when the document presents a high number EMs. As exemplified by them, if a document presents 10 EMs, and each mention presents 10 entity candidates, they must evaluate 10^{10} distinct combinations. For circumventing this, they propose a "heuristic optimization algorithm based on state-space search exploration". This algorithm creates a few numbers of random combinations of entity candidates of distinct EMs (for simplicity, we will call these combinations as a tuple of entity candidates) and picks the one with the better score. Next, for the picked tuple, they randomly select a position in the tuple (the position represents an EM in the text) and verify if other entity candidates for the same mention have a better score for the tuple. In this case, the entity candidate is replaced by the better one. This optimization algorithm ends after a pre-specified number of runs or when the score of the tuple does not improve after a few runs. This optimization algorithm allows their approach to run in less than one second, and, therefore, is viable for real-time applications.

5.5.3.2 *Embedding-based Methods*

A recent strategy for disambiguation is the jointly use of different types of embeddings, like word and entity embeddings [32, 33, 69, 70, 138, 145, 146, 179, 186, 292, 307], document and graph embeddings [247], entity and knowledge embeddings [250], and word and knowledge embedding [208]. In Fang et al. (2016) [69], the authors combine the entity model with the word model by using two alignment techniques proposed by Wang et al. (2014) [286] (based on WIKIPEDIA anchors and entity names, respectively) and one alignment technique from Zhong et al. (2015) [304] (based on entity descriptions). With the models

aligned and a few features selected for disambiguation, the authors select the best entity candidate for a given mention in a two-layer disambiguation model. Similarly to Fang et al. (2016) [69], Yamada et al. (2016) [292] propose an embedding model based on skip-gram [175, 176] (skip-gram model, KG model, and anchor context model). Besides employing the embedding model, the authors also propose and exploit textual context similarity and coherence to disambiguate EMs by using the learning-to-rank algorithm GBRT (Gradient Boosted Regression Trees) [81].

Shi et al. (2020) [250] take a further step and aligns four embedding models to disambiguate EMs in sentence level. This proposal employs Feature-Entity embedding, which represents the context of entity pages in WIKIPEDIA by using WORD2VEC. This model is similar to the entity embeddings used in other approaches. Mention-Entity embedding represents the context for a given mention already disambiguated (i. e., $\langle mention, entity \rangle$). This embedding model employs l_2-regularization and Hinge Loss. The knowledge embedding, by its turn, represents the facts in a KG, in this case, YAGO. The authors propose knowledge embedding training similar to WORD2VEC. Lastly, coherence embedding represents the interactions between disambiguated mentions in the same sentence. In Moreno et al. (2017) [179], the authors proposed an embedding model called Extended Anchor Text (EAT), which is also based on the model proposed by Mikolov et al. (2013) [176]. However, differently from [69, 292], the proposal of Moreno et al. (2017) [179] is based on one model.

Some approaches consider that both words and entities are in the same distributive space [69, 179, 292]. However, Chen et al. (2018) [32] criticize such an assumption. They consider that words and entities are in different distributive spaces because entity surface names can consist of multiple words, and the occurrence scales of words and entities in a text are different. Therefore, the authors propose a Bilinear Joint Learning Model (BJLM), an extension of the skip-gram model [175, 176]. According to the authors, the bilinear model *"simulates the interactions between the word distributive space and the entity distributive space"*.

Kolitsas et al. (2018) [138] propose a neural model that uses both embeddings. Their neural model is composed of a bi-LSTM (bidirectional Long Short-Term Memory) and shallows FFNN. The word embeddings of the mentions and the words around them are fed to the bi-LSTM to generate word embeddings aware of their context. Briefly, the context-aware word embeddings of the EMs, the entity embeddings of the entity candidates and other features relevant for EL are fed to an FFNN to get the local score of the entity candidates for a given mention. Lastly, the local score and a partially global mention-entity score are fed to the last FFNN, whose output is used to disambiguate mentions. Mueller & Durrett [186] also do not align the word and

entity embeddings. The authors jointly train both embeddings by using the WORD2VECF technique [152] on WIKIPEDIA pages. This allows both embeddings to be in the same distributive space. To perform the EL task, the authors employ the jointly trained word and entity embeddings and lexical features in a Gated Recurrent Unit (GRU) with an attention mechanism. The authors obtained state-of-the-art results in the *WikilinksNED* dataset [65].

In most KGs, different relations $r \in \mathcal{R}$ connect entities. According to Le & Titov (2018) [146], these relations can improve the results of EL approaches. Thus, they propose a CRF model that considers word, entity, and relation embeddings. Furthermore, Le & Titov (2018) [146] proposes three ways to represent relations: general form, relation-norm, and mention-norm. Given a set $\mathcal{M}$ of EMs in a *text source* and a set $\mathcal{R}$ of latent relations, the pairwise score (m_i, m_j), where $m_i, m_j \in \mathcal{M}$, is given by the weighted sum of relation-specific pairwise scores. The relation-norm and mention-norm are the general form with normalization over, respectively, the relation and the mentions.

Besides relations that connect entities in a KG, most entities have at least one type (e. g., `dbo:Person`), which indicates their classification in a given ontology. However, Chen et al. (2020) [33] stress that existing approaches do not exploit sufficiently entity types in the disambiguation step of EL and propose to imbue entity types information into BERT (Bidirectional Encoder Representations from Transformers) pre-trained entity embedding [53]. They apply the entity similarity score calculated from the entity embeddings into the local context model of the disambiguation step proposed by Ganea & Hofmann (2017) [86]. Chen et al. (2020) [33] consider that the immediate context that surrounds a mention in its entity page (e. g., WIKIPEDIA page) may summarize its types and replace the EM in its page by the token [MASK]. Then, they extract the uppermost layer representing the token [MASK] to represent the entity type.

Zhua & Iglesias (2018) [307] argue that current unsupervised EL approaches are not effective for short texts, like queries and social media posts. According to them, this occurs because these approaches depend mainly on features like context similarity and relatedness between the entities. However, short texts have limited context, and few EMs, limiting the use of such features. To circumvent these limitations, Zhua & Iglesias (2018) [307] propose an approach based on the contextual similarity between the mention and entity descriptions, which was introduced by them in the so-called Semantic Contextual Similarity-based Named Entity Disambiguation. They also present an approach based on a new embedding model, called Category2Vec, which learns categories from joint embeddings of KG resources and words, based on the entity abstracts and entity categories. Both approaches achieve better results compared with other unsupervised EL

approaches and present competitive results against EL approaches in general.

Instead of the joint use of the word and entity embedding, Sevgili et al. (2019) [247] propose the jointly use of document and graph embeddings. The authors present a neural model that exploits both embeddings to disambiguate the EMs. A single FFNN layer composes their neural model, and its inputs are the document vector of the context that surrounds the EM; the document vector of the EM itself; the document vector of a long abstract of the entity candidate and; the graph embedding of the entity candidate. The doc2vec [147] technique generates the document embeddings using English WIKIPEDIA pages, while the DeepWalk [220] technique generates the graph embeddings using DBPEDIA triples. Lastly, Sevgili et al. (2019) [247] improves the previous approach with graph embeddings. The improvement with graph embeddings results in slightly better Precision and Recall.

Differently from approaches that use different techniques to produce different embeddings and then align them, Oliveira et al. (2020) [208] jointly trains both word and knowledge embeddings and exploit them together in a deep neural network to disambiguate EMs in microblog posts. They do so by employing the fastText technique, which is capable to jointly train both word and knowledge embeddings in the same vector space. This allows skipping the alignment step performed by other approaches. To exploit both word and knowledge embeddings concomitantly, the authors replace the EMs in the microblog posts by their respective entity candidates, one at a time. Word embeddings represent the words that surround the mentions, while knowledge embeddings represent the entity candidates. These embedded representations of microblog posts are fed into a bi-LSTM, which is followed by an FFNN. If an entity candidate fits the post context, the neural network classifies it as correct. The results surpass most state-of-the-art approaches.

Most EL approaches that use machine learning focus on existing labeled data for training. However, Le & Titov (2019) [145] argue that such approaches fail in domains where few labeled data exist, like scientific domains. To employ EL in such domains, they tackle the EL task as multi-instance learning [56]. In their approach, the multi-instance learning first classifies bags of examples, depending on if they contain or not the correct entity for a given EM. Then, it classifies the instances of a bag that supposedly contains the correct entity. To achieve this, two sets of entity candidates are generated for each EM. The first set, named E^+, is composed by entity candidates found by a surface-match heuristic proposed by Riedel et al. (2010) [232]. The heuristic guarantee that most of the time, the correct entity for a EM is in E^+. The second set, named E^-, is composed of randomly retrieved entity candidates and does not contain the correct EM. To disambiguate the entity candidates, the authors propose a neural network

model based on a bi-LSTM and an FFNN. The bi-LSTM encodes the context surrounding the EMs, represented by word embeddings. The FFNN has as inputs the bi-LSTM's output and the entity embedding representation of the entity candidates. The binary noise detection classifier is trained jointly with the previous neural network model to improve the results.

5.6 COMPARATIVE ANALYSIS OF HOLISTIC ENTITY LINKING

Table B.1 provides a comparison summary of proposals found in the literature about the EL task that presents some holism, as reported in Section 5.5. We list them in chronological ordering, presenting the columns according to relevant aspects of holism that we have proposed in Section 5.4. The column *External Input* refers to the databases or KGs that are used to semantically enrich the data to be annotated, i. e., the sources of resources that can semantically describe what is mentioned in the text. Column *NLP tasks* refers to NLP tasks that are combined or preceded the EL task. Lastly, the column *Method* refers to the methods used to disambiguate the entity candidates. Other methods and tools used to generate features or preprocessing the data are not shown in this table because they are outside the scope of this work.

The first highlight of Table B.1 is the column *Input*. In this column, it is possible to perceive that most of the works use WIKIPEDIA as the source of the entities to semantically enrich the text to be annotated. Some works use KGs (e. g., FREEBASE, DBPEDIA, YAGO) to complement the information available in WIKIPEDIA [161, 180, 287], to help in the generation of entity embeddings [32, 69, 145, 146], graph/knowledge embedding [208, 215, 250], or category embedding [307]. Although DBPEDIA is the Linked Open Data (LOD) version of WIKIPEDIA [9, 150] (i. e., the KG version of WIKIPEDIA), only the works [127, 208, 215, 247, 307] uses DBPEDIA directly in the EL task. The annotator proposed by Moro et al. (2014) [180] is the only one that uses the KG BABELNET [194], which combines WIKIPEDIA and WordNet, to jointly perform EL and WSD.

Although most of the KGs are encoded as Resource Description Framework [45, 136] (RDF) triples, and, therefore, are machine-readable, we reasoning that WIKIPEDIA is still widely employed by EL approaches for several reasons, including:

- Most KGs have a slow update-cycle[12], and some have been discontinued (e. g., FREEBASE). Meanwhile, it is possible to get updated dumps from WIKIPEDIA every month[13];

12 Until the submission of this work, the last public data available from DBPEDIA is from 2016, while YAGO is 2017

13 https://dumps.wikimedia.org/

- Although several KGs, like DBpedia and BabelNet, include LOD versions of information extracted from Wikipedia, they present far less textual content than the latter. This makes their use difficult for several approaches that depend on such textual content, like approaches that use word and entity embedding;
- Some metadata about the Wikipedia pages, like page views, are used as features to disambiguate entity candidates. There is no guarantee that such metadata will be available in the KGs.

Regarding the column *NLP tasks*, it is possible to notice that most of the works do not present any other NLP tasks besides EL, frequently disregarding even the NER task. This happens because most of the works consider that all the EMs are already recognized. Therefore, such works focus solely on the EM disambiguation. The works that present the NER task propose an end-to-end EL approach, i. e., they propose a new approach for both NER and EL. Lastly, only a few works present an NLP task besides the NER task. More specifically, the ED and ES tasks.

Since Table B.1 is sorted chronologically, we highlight in its last column, *Method*, that the EL approaches are shifting from graph-based methods to approaches that use embeddings of words and entities. Embedding techniques are capable to model local and global interactions between entities or words in low-dimension vectors [42, 82, 93, 243]. Therefore, the use of such embeddings enable approaches to achieve the same or better results as using graph-based methods, with higher scalability. Moreover, some works employ graph embedding [247] and knowledge embedding [69, 208, 250] instead of entity embedding, while Parravicini et al. (2019) [215] uses only graph embedding. These works are essential to show that embeddings generated from DBpedia are a viable option for EL approaches. Differences between graph embedding and knowledge embedding are presented in Section 5.7.2.

Given the variety of currently available EL approaches with several distinguishing characteristics, it may be challenging to decide which one is the most appropriate for a specific application or domain. Thus, we propose a Decision Tree (DT) to help make such a choice based on features that correspond to our 3 holistic aspects of EL approaches, as illustrated in Figure 5.3. If the disambiguated entities need to be semantically coherent, then it is more appropriate to select an approach that makes collective disambiguation instead of non-collective. The choice also depends on the document type: microtext (e. g., social media posts, text snippets) versus any longer or more formal text (e. g., news, books, articles), which is expected to carry more context information and to have fewer typos, grammatical errors, slangs and other kinds of noise. Finally, and optionally, having the NER task integrated with the EL solution may be convenient, for example, to avoid setup of distinct tools. Leaves of the DT in Figure 5.3 refer to

groups of approaches sharing the same selective features. Due to the number of approaches in each group, the citations of the respective works, along with links to repositories containing their open-source code (when available) or links to the GitHub profiles of their author(s) are listed in Table B.2.

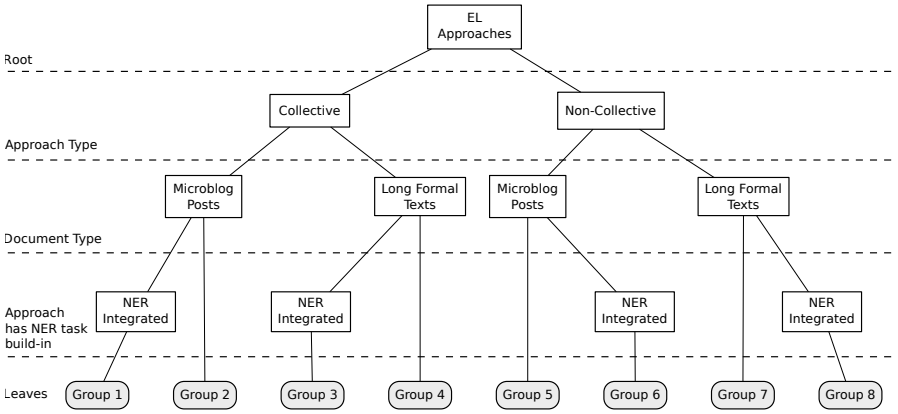


Figure 5.3: Proposed DT to support the selection of EL approaches. White boxes refer to approach characteristics considered in each DT level. Horizontal dashed lines separate the levels. Gray rounded boxes in the leaves refer to groups of works considered analogous with respect to our decision criteria. Works within each group are listed in Table B.2.

Notice that Table B.2 covers all the works discussed in Sections 5.5 and 5.6, grouping them according with the DT of Figure 5.3. Although not providing links to source code, Moro et al. (2014) [180] and Rama-Maneiro et al. (2020) [229] provide Web applications that demonstrate their proposals, namely Babelfy[14] and ABACO[15], respectively. Some papers provide Github links to the tools and models that support their approaches, such as Trani et al. (2016) [272, 273] with Elianto[16] and Dexter[17] and, Yamada et al. (2016) [292] providing their embedding model Wikipedia2vec[18].

In addition to the criteria considered in our DT (Figure 5.3) and the availability of the approach as open-source, the quality of the results may also be a crucial decision factor. Thus, in the following (Section 5.6.1), we provide a performance comparison summary of the approaches analyzed in this work.

5.6.1 *Evaluation of Holistic Entity Linking Approaches*

The works we have analyzed use several distinct metrics to evaluate their EL approaches. However, we noticed that the most used one

14 http://babelfy.org/
15 https://tec.citius.usc.es/abaco/
16 https://github.com/dexter/elianto
17 https://github.com/dexter/dexter
18 https://github.com/wikipedia2vec/wikipedia2vec

is the F-measure. It is is the harmonic mean of Precision and Recall and tolerates uneven class distributions. Thus, we first compare the approaches using the F-measure, considering the formal definition of the EL task presented in [85, 229].

The majority of the analyzed works that use the F-measure to evaluate their approaches also use the GERBIL benchmark system [279] to produce automatic and reliable evaluations that follows the FAIR data principles [289] (FAIR) guidelines. GERBIL calculates the micro and macro F-measure to better evaluate the performance of EL approaches. While the micro F-measure evaluates performance over the whole dataset, the macro F-measure evaluates performance for each document and takes the average.

Table B.4 presents the F-measure of works analyzed in this work (listed in the first column of the table, by the chronological order of their publications) that use this metric to evaluate performance on distinct datasets (listed in alphabetic order in the second line of the table header). The last column of Table B.4 indicates if the performance of the respective approach was evaluated using a benchmark system or not. The micro F-measure is provided by all works (though not for every dataset listed in the table), while only a few works also provide the macro F-measure (which may appear below the respective micro F-measure). When the respective article provides variations of the F-measure (e. g., due to distinct parameter settings in their method), we only present the highest value that was obtained.

Table B.3 summarizes the performance of the works we have analyzed that use other metrics (listed in the first column of the table) to evaluate their approaches instead of the standard F-measure. Notice that the next most used evaluation metric is by far the accuracy. Differently from the F-measure, the accuracy does not tolerate uneven class distributions. Therefore, depending on the dataset used to evaluate the EL approach, the accuracy may provide a less reliable measure of the performance.

Tran et al. (2015) [271] consider the EL task as a ranking problem. Therefore, they evaluate if the correct entity for each mention has the highest score among the n best results returned by their approach. In their work, they consider the following metrics: Precision at 5 (P@5), Precision at 15 (P@15) and, Mean Average Precision (MAP).

The remaining metrics, appearing at the bottom of the first column of Table B.3, are variations of the F-measure. In Wick et al. (2013), the authors tackle both the EL and the ED tasks. Due to this, they employ the Pairwise F-measure metric for performance evaluation. Unlike the standard F-measure, the Pairwise F-measure takes into account pairs of EMs in the text document considered to refer to the same entity. The metrics $CEAFmCF1$ and $NERLCF1$ are the standard F-measure restricted to specific types of experiments performed on

the *TAC* dataset. *CEAFmC* denotes *typed_mention_ceaf* and *NERLC* denotes *strong_typed_all_match*.

Among all the works we have analyzed, only Kalloubi et al. (2016) [127] is not present in either Table B.4 nor Table B.3. It happens because the authors propose an EL approach to semantically enrich tweets for improving their retrieval. Therefore, their experiments evaluate the quality of their retrieval approach and not the EL task itself.

The authors of Chong et al. (2017) [35] aim to measure how their collective EL approach, based on temporal and geospatial features, compare with a non-collective EL approach for tweets. Therefore, they measure the ratio of positive and negative changes in their approach over a non-collective approach. They consider as a positive change when their approach fixes an incorrectly disambiguated entity yield by the baseline. Conversely, a negative change is when their approach transforms a correctly disambiguated entity into an incorrect one.

Finally, Table B.5 provides pointers to further details about the datasets mentioned in Table B.4 and in Table B.3, as links to the respective home pages, when they are available. When such a link is unavailable, we provide a reference to the paper or challenge in which the dataset appears.

5.7 POTENTIAL PILLARS FOR FUTURE HOLISTIC APPROACHES

For the best of our knowledge, the trends for EL holistic systems have not been adequately identified and described yet. Therefore, this section describes the pillars that we have identified from our bibliographical review and previous experience.

5.7.1 *The General Semantic Annotation Process*

We have observed that a sequence of tasks always occurs in the EL approaches that we have analyzed, independently of the data used and the specific methods employed to realize these tasks. It can be regarded as a general process for EL, as shown in Figure 5.4. Firstly, for several reasons, the *Text documents* given as input for EL can be noisy (e. g., due to spelling errors), and the *External inputs* used to help the EL task can be heterogeneous (e. g., because they may be from different

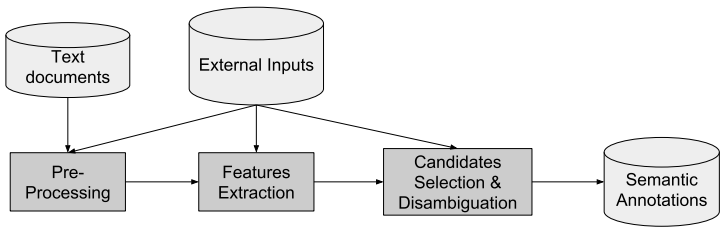


Figure 5.4: General process for EL.

sources). Thus, a *Preprocessing* stage is usually required for cleaning these data and standardizing them in some format. The next stage is to extract the relevant features from the preprocessed data (*Features Extraction*). Looking at resources from external inputs (e. g., WIKIPEDIA, KGs) may contribute to this task to produce good results. The features extracted are used in the *Candidates Selection & Disambiguation* task to disambiguate the possible entity candidates for each EM. This process can be generalized for semantic annotation in general, just by changing the implementation of its tasks and sometimes allowing other kinds of data to be annotated instead of only *Text documents*. The sequence of tasks remains the same, even though each one can be done in very different ways.

5.7.2 *Knowledge Graphs and Knowledge Embedding*

As presented in Section 5.5.2 and Section 5.6, only a few works effectively use a KG (e. g., DBPEDIA, YAGO, FREEBASE, BABELNET) as an external input for EL. Although, we present several reasons for existing approaches to choose WIKIPEDIA instead of KGs as their primary external input. We envision that knowledge bases such as KGs are essential for future holistic EL approaches by taking advantage of KGs that are encoded as RDF triples to quickly exchange the KG (extract) used for EL in accordance with the domain of the text to be annotated. For example, DBPEDIA can be used to annotate news and social media posts, as it presents entities of several domains. Meanwhile, for medical reports, which deal with more specialized knowledge, EL approaches can use medical knowledge graphs [242]. Moreover, the adoption of KGs in future holistic EL approaches enables the use of knowledge embedding on them.

Knowledge embedding [284] is a specialized topic from graph embedding [93], whose objective is to embed the entities and relations of a KG in low-dimension vectors. From the works analyzed in Section 5.4 and compared in Section 5.6, only Fang et al. (2016) [69] employs knowledge embeddings. However, we believe that the full potential of knowledge embedding in the EL task has not been fully explored yet. Knowledge embedding techniques can capture local, long-range, and global statistics of dependencies present in KGs into embeddings [203]. These dependencies may be useful for EL approaches that disambiguate entities collectively. Moreover, we envision that works that employ graph embedding [215, 247] may improve their results by using knowledge embedding, as shown in the papers [208, 250]. This may happen because graph embedding techniques, like DeepWalk and node2vec [95], are meant for any graph and, therefore, may not exploit effectively KG features (e. g., a high number of distinct relations) as well as knowledge embedding. Lastly, word embedding and knowledge embedding techniques, like fastText [124], are pro-

gressing to allow training models not only with facts but also with textual properties of the entities, like entity labels and abstracts. Such combinations of word embeddings with knowledge embeddings may enable improved EL approaches.

5.7.3 *Building and Exploiting Historical Contexts*

A historical context captures the most critical entities for a specific (group of) agent(s) involved in the production of the text to be annotated (e. g., author of a book, social media user). The knowledge and experience expressed in historical contexts can help to disambiguate highly ambiguous EMs. For example, considering the left tweet in Figure 5.2, if the historical context of the tweet sender presents entities related to basketball, this may help to disambiguate the mention "*Jordan*" to the basketball player *Michael J. Jordan*. Historical context can also include entities from old books of an author and help to identify mentions to these old entities in his/her new book. Differently, from the previous potential pillar, some works propose concepts similar to historical contexts, like semantic profiles [1, 2] for social media users. However, to the best of our knowledge, EL approaches have not considered the use of such profiles or historical contexts.

5.7.4 *Holistic Entity Linking Reference Approach*

The combination of potential pillars in a seamless process, where each pillar can efficiently exploit each other, can further enhance the benefits of holism for the EL task. Therefore, we propose as last pillar a *Reference Approach for Holistic EL* based on the pillars previously described.

Figure 5.5 illustrates the reference approach that we propose for holistic EL. It derives from the generic EL process presented in Figure 5.4, but supports all the pillars for semantic approaches, and allows cycles for capturing historical contexts from annotations and use them to disambiguate entity candidates and semantic expansion. Each stage of this process can be adapted to fit some text type (e. g., social media posts, news) or EL approach.

In the *Feature Extraction* task, several tools and methods can be used to recognize different types of EMs (e. g., people, places) in a text. Work-flows can combine them to suit some specific domain better. Besides the recognizing of EMs, we also consider the building and updating historical contexts as feature extraction.

The building of historical contexts takes as input semantic annotations previously created and existing historical contexts in case if it is necessary to update them. Several semantic annotation features can be used to build semantic contexts, like the entity pointed by them, their creation timestamp, which user or application created the

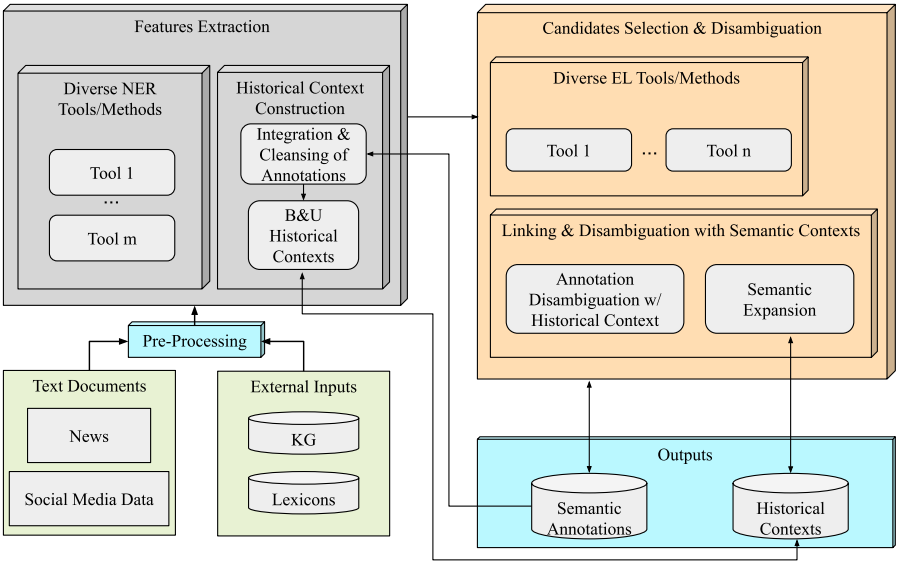


Figure 5.5: Reference Approach for Holistic Entity Linking.

annotation and others. However, it is necessary to clean and integrate the semantic annotations before their use in historical contexts. A repository of semantic annotations can have inconsistency among its annotations. For example, for the same mention in a text document, two EL tools can disambiguate to different entities (e. g., the mention "*Jordan*" in Figure 5.2 can point to the country *Jordan* or the basketball player *Michael J. Jordan*). The output of this step is a new historical context for an agent or an updated existing historical context.

Similarly to the *Features Extraction* stage, the *Resource Linking & Disambiguation* stage also can employ several EL tools/approaches. However, these tools focus on disambiguating mentions recognized in the *Features Extraction* stage. Different from the existing works, we propose the use of historical contexts in the EL task. The entity candidates considered can be compared with the preference of an agent expressed in historical contexts already built to better disambiguate the EMs.

Lastly, we propose a step called *Semantic Expansion*. We envision that historical contexts also can express correlations between some entities. Such correlations can be used to generate new semantic annotations not explicitly expressed in the text. For instance, most of the people go to restaurants to eat or go to shopping malls to work or buy things. Therefore, it is possible to infer the activity of an agent from the places visited by him/her and vice versa. This step adds more context to the text, and that can be useful for data with little contexts, like social media posts.

6

ON EXTRACTING RELATIONS USING DISTRIBUTIONAL SEMANTICS AND A TREE GENERALIZATION

In this chapter, we present our Relation Extraction (RE) approach that follows a closed and binary setting to extract generalized tree patterns from text sources. We present our tree generalization and our filtering with distributional semantics for these patterns to reduce semantic drift in our collection of tree patterns.

6.1 RELATED WORK

Numerous approaches for extracting relations have been developed in the recent past. Detroja et al. (2023) [52] and Han et al. (2020) [101] provide a current overview of a broad spectrum on the topic of RE. Approaches for *closed relation extraction* [89, 192] (in contrast to *open relation extraction* [47, 59, 68, 169, 296]), are based on vocabularies that define relations a priori, i.e., in a domain ontology or an extraction template. Consequently, such systems require no mapping of the extracted relations to a vocabulary and thus produce less uninformative or incoherent elements from unstructured text [12]. *Supervised learning approaches* are a core component of a vast number of relation extraction tools as they offer high precision and recall. The need for manually labeled training data makes these methods not scalable to thousands of relations found on the Web. More promising approaches are *semi-supervised bootstrapping approaches* [44, 89, 234] and *Distant Supervision (DS) approaches* [12, 178, 192], since these do not need a complete manually labeled training corpus. DS has become an important technique because of the availability of large Knowledge Graphs (KGs). It utilizes facts from a KG for labeling mentions of these facts in an unannotated corpus to create a training set. Thus, it fulfills the needs of large-scale applications with minimal human effort but introduce noise examples, i. e., *FP*s. Designing schemas to obtain high quality and high coverage data to train robust RE models still remain a problem to be explored [101]. [193] gives an exhaustive overview of approaches for the RE task.

Boa [89] is a bootstrapping strategy for extracting Resource Description Framework [45, 136] (RDF) from unstructured data. Its idea is to

¶ Parts of this chapter have been published as conference article [262]. The author of this thesis is also the main author of the article and developed the main ideas, designed, and implemented major parts of the solution, and wrote the majority of the publication.

use the Web of Data as background knowledge for the extraction of natural language patterns that represent predicates found on the Web of Data. These patterns are used to extract instance knowledge from natural language text. This knowledge is finally fed back into the Web of Data. BOA provides a repository of natural language representations of predicates found on the Web of Data.

PATTY [192] is a large resource for textual patterns that denote binary relations between entities based on DS. PATTY uses frequent itemset mining and the patterns are semantically typed as well as organized into a subsumption taxonomy with scores, support and confidence. The taxonomy is available online but not in machine-readable data to use it by the community. We asked the authors to provide the source-code and the database with the results as well as the measures but we could not receive it.

One drawback of both state-of-the-art systems, BOA and PATTY, is that one pattern can be matched to several relations which results in a significant number of *FP*s. Thus, this leads to a noisy behavior in applications such as Knowledge Graph Question Answering (KGQA). Another drawback of these two systems is that both extract relations that are enclosed by Named Entities (NEs) only. For instance, both systems cannot find the relation in the sentence "*Michelle Obama and Barack Obama are married.*" as the verb which mentions a relation is not enclosed by the NEs. We address these drawbacks by operating on dependency parse trees and using a generalization approach inspired by [151] which tackles the semantic drift issue faced by many current approaches.

6.2 OVERVIEW

6.2.1 *Closed Binary Relation Extraction*

The RE task extracts relations between entities in text [13, 306]. In our closed binary setting, all relations considered as given by a fixed set of relation types, for instance $\mathcal{R}$. In general[1], given a relation $r \in \mathcal{R}$, a sentence $\mathfrak{s} = \langle \omega_1, \omega_2, \ldots \rangle$ containing at least two Entity Mentions (EMs) $m_1, m_2 \in 2^{\mathfrak{s}}$ expressing two entities with $\varepsilon_1, \varepsilon_2 \in \mathcal{E}$. The EMs are given by $m_1 = \langle \omega_i \rangle_{i=k,k+1,\ldots,l}$ and $m_2 = \langle \omega_j \rangle_{j=m,m+1,\ldots,n}$ with $1 \leq k \leq l \leq m \leq n$.

A mapping function $\chi(.)$ can be given as

$$\chi_r(\Gamma(\mathfrak{s})) = \begin{cases} +1 & \text{if } m_1 \text{ is } r\text{-related to } m_2 \text{ in } \mathfrak{s} \\ -1 & \text{otherwise} \end{cases} \tag{6.1}$$

where

$\Gamma(\mathfrak{s})$ are features extracted from $\mathfrak{s}$.

1 We following the definition of Bach and Badaskar [13] for the RE task.

After performing feature extraction with textual analysis on $\mathfrak{z}$, for instance Part of Speech (POS) tagging or dependency parsing, the mapping function $\chi(.)$ decides if m_1 and m_2 are related to r in $\mathfrak{z}$ or not.

In *supervised approaches* the function $\chi(.)$ can be constructed as a discriminative classifier by training on a labeled set of positive and negative relation examples using feature sets.

6.2.2 *Distant Supervision*

Mintz et al. (2009) [178] proposed the DS paradigm without the requirement of a labeled set of positive and negative relation examples but with a given large semantic database for automatically gathering relation-type labels [27, 219]. DS combines the *supervised paradigm*, by using a feature-based probabilistic classifier, with the *unsupervised paradigm*, by extracting relations from large corpora of any domain.

Traditional DS approaches are limited by their inability to support the modeling of overlapping relations, i. e., for the same pair of entities, there can be multiple valid relations, e. g., FoundedBy(Steve Jobs, Apple) and CEO(Steve Jobs, Apple) [219].

6.2.3 *Tree*

Let $T = (V, A, E, \Phi_A, \psi)$ be an attributed dependency parse tree [134, 135, 172] (i. e., rooted and ordered [55, 297]) with a finite set of vertices V, a set of edges $E \subseteq V \times V$, a vertex labeling function family $\Phi_A = \{\phi_a | a \in A\}$ where A is a finite set of vertex attributes[2] so that $\phi_a : V \to \Sigma_a^*$ as well as with an edge labeling function $\psi : E \to \Sigma_E^*$.

We refer for a specific tree T with $V(T)$ for vertices, $E(T)$ for edges, $\Phi_{A,T}$ for the vertex labeling function family and ψ_T for the edge labeling function. We denote the *root vertex* with $\hat{r}_T$ and the root dependency vertex with $\hat{s}_T$, i. e., the semantic head of the sentence.

6.2.4 *Subtree*

Let T be a tree and $v \in V(T)$. The ordered sequence of child vertices of v in T is denoted by $\hat{c}_T(v) = \langle u_i \rangle_{i=1,2,\ldots}$ with $u \in V(T)$. We then denote by $T(v)$ the subtree $T' = (V', A', E', \Phi_A', \psi')$ with $\hat{r}_{T'} = v$, $V' = \hat{c}_T(v) \cup \{v\}$, $E' = E \cap V' \times V'$, $\Phi_A' = \Phi_{A|V'}$ and $\psi' = \psi_{|E'}$.

6.2.5 $\leq$ *Relation*

For trees T_1 and T_2, we have $T_1 \leq T_2$ iff the following holds:

2 $A \leftarrow$ *label*, *lemma*, *pos*, *ner*, *domain*, *range*, *general* are the vertex attributes used throughout this work.

1. if $\hat{s}_{T_2}$ exists, then:
   a) $\hat{s}_{T_1} = \hat{r}_{T_1}$, $\hat{s}_{T_2} = \hat{r}_{T_2}$
   b) $\phi_{T_1,lemma}(\hat{r}_{T_1}) = \phi_{T_2,lemma}(\hat{r}_{T_2})$
   c) $\phi_{T_1,pos}(\hat{r}_{T_1}) = \phi_{T_2,pos}(\hat{r}_{T_2})$
2. if $\hat{s}_{T_2}$ does not exist, then:
   a) $attribute \leftarrow \phi_{T_2,general}(\hat{r}_{T_2})$
   b) if $attribute = label$, then $A^* \leftarrow \{label, lemma, pos\}$
   c) if $attribute = lemma$, then $A^* \leftarrow \{lemma, pos\}$
   d) for each a in A^*
   $\phi_{T_1,a}(\hat{r}_{T_1}) = \phi_{T_2,a}(\hat{r}_{T_2})$
   e) if $attribute \in \{pos, ner, domain, range\}$, then
   $\phi_{T_1,attribute}(\hat{r}_{T_1}) = \phi_{T_2,attribute}(\hat{r}_{T_2})$
3. for each edge $(\hat{r}_{T_2}, v_2)$ in $E(T_2)$ there exists an edge $(\hat{r}_{T_1}, v_1)$ in $E(T_1)$ with $\psi_{T_2}(\hat{r}_{T_2}, v_2) = \psi_{T_1}(\hat{r}_{T_1}, v_1)$ such that: $T(v_1) \leq T(v_2)$

We define $T_1 \simeq T_2$ as $T_1 \leq T_2$ and $T_2 \leq T_1$. $T_1 < T_2$ is defined as $T_1 \leq T_2$ and $T_1 \not\simeq T_2$.

6.3 APPROACH

This section initially presents an overview of the data flow and subsequently provides insights into each package of our proposed framework.

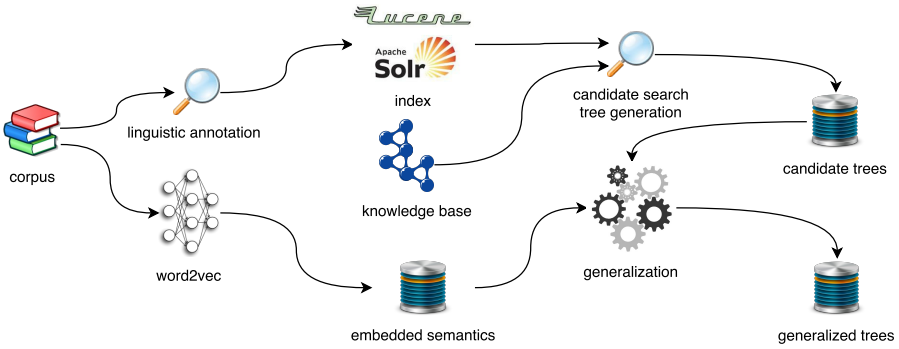


Figure 6.1: The data flow of the proposed framework.

6.3.1 *Overview*

The data flow of our framework, dubbed OCELOT, is depicted in Figure 6.1. The goal is to harvest generalized dependency tree patterns, which are useful for a wide range of applications to extract relations from unstructured text.

OCELOT starts by preprocessing the corpus with Natural Language Processing (NLP) tools to acquire linguistic annotations. These annotations are stored in an index for a fast search. Thereafter, it queries a KG for predicates which are the target relations, related resources as well as labels of these resources. These labels serve as search keywords to query candidate sentences in the index that might contain target relations. The dependency parse trees on these candidate sentences are created and stored. In the generalization step, the candidate trees are generalized by linguistic annotations as well as are scored and ranked. OCELOT relies on DS and thus introduces errors by semantic drift [44, 234]. To reduce this drift, it filters ambiguous trees. OCELOT utilizes embedded semantics by training word2vec [175] on the corpus. The vector representation of the labels from the KG for the predicates as well as from other sources, for instance Wordnet, are used in the generalization step to filter out ambiguities among trees to reduce semantic drift. In the following, we explain each of these steps in detail.

6.3.2 *Linguistic Annotation*

We begin with an input corpus, which is first preprocessed. The core of the preprocessing consists of removing possible markup from the corpus (e. g., Hyper Text Markup Language (HTML) tags). We then sample the frequency distribution of the sentences' length (number of tokens in a sentence including the end punctuation). On this distribution the mean μ and standard deviation σ are calculated to filter out sentences that are very long and thus require long processing time in the framework. OCELOT then selects sentences with a minimum of four[3] and a maximum of $\mu + 2\sigma$ tokens. Linguistic annotations (lemmas, POS tags, NEs)[4] are computed for the selected sentences (with STANFORD in our current implementation). Based on the assumption "*if two entities participate in a relation, at least one sentence that mentions these two entities might express that relation*" stated in [232], we discard sentences which contain less than two NEs. The remaining sentences and annotations are stored in a Solr index for a fast search.

6.3.3 *Candidate Selection*

This step's two main functions are: (i) To find for each target relation $r \in \mathcal{R}$ a set of candidate sentences $\mathscr{C}_r \subseteq \mathscr{C}$ that might express that target relation in a given corpus $\mathscr{C}$ by searching the created index (see

3 The shortest sentence with a relation has at least two tokens for the NE arguments, one token for the relation mention and one for the end punctuation.

4 Seven types are applied (`PLACE`, `PERSON`, `ORGANIZATION`, `MONEY`, `PERCENT`, `DATE`, `TIME`).

Section 6.3.2), and (ii) to parse each set of candidate sentences $\mathscr{C}_r$ to a set of dependency parse trees $\mathcal{T}_r$.

We rely on background knowledge from the given KG $\mathcal{G}$ to search for candidates. In the first step, the predicates in $\mathcal{R}$ with the highest numbers of resource instances are chosen from the KG. These selected predicates serve as target relations in OCELOT. For each target relation $r \in \mathcal{R}$, the candidate selection queries $\mathcal{G}$ for the set of statements where each statement consists r

$$\mathcal{S}_r = \{(s, r, o) : (s, r^*, o) \in \mathcal{S} \rightarrow r = r^*\} \subseteq \mathcal{S}. \tag{6.2}$$

With the labeling function γ, given by the KG, we get the labels for resources that we employ to search in the index. As some extended labeling functions are available for some KGs (e. g., [88] proposes an extension of the method originally proposed in [173] to gather additional labels from DBPEDIA). We assume the existence of such a method which can generate extended labels for any resource $r \in \mathcal{E}$ and call this method $\zeta(r)$. Then, the sets $\mathcal{S}_r$ and $\mathcal{O}_r$ of all labels for all subject respectively object resources of a target relation are given by

$$\mathcal{S}_r = \bigcup_{(s,r,o)\in\mathcal{S}_r} \zeta(s) \cup \gamma(s) \text{ and } \mathcal{O}_r = \bigcup_{(s,r,o)\in\mathcal{S}_r} \zeta(o) \cup \gamma(o). \tag{6.3}$$

Let $\Omega(A, B)$ be the set of elements where each element $\mathfrak{s}$ (i. e., a sentence) belongs to A and each element $\mathfrak{s}$ consist of at least one element that belongs to B and the set of all contiguous subsequences of $\mathfrak{s}$

$$\Omega(A, B) = \{\mathfrak{s} | \mathfrak{s} \in A \wedge \exists b \in (B \cap 2^{\mathfrak{s}})\}. \tag{6.4}$$

Therewith, the set $\mathscr{C}_r$ contains candidate sentences with tokens that mention subject and object resources of a target relation

$$\mathscr{C}_r = \Omega(\mathscr{C}, \mathcal{S}_r) \cap \Omega(\mathscr{C}, \mathcal{O}_r) \subseteq \mathscr{C}. \tag{6.5}$$

To reduce semantic drift, only candidate sentences with tokens which mention subject and object resources and which are tagged as NEs by the linguistic annotation package (see Section 6.3.2) are collected. These sets of candidate sentences for each target relation r are parsed to sets of candidate dependency parse trees $\mathcal{T}_r = \{T_1, T_2, \ldots\}$.

6.3.4 *Embedded Semantics*

This step serves as preprocessing for the subsequent generalization step. We create word-level embeddings on the corpus and apply predicate labels from the KG to find semantically similar words in several sources. We trained the continuous skip-gram model [175] implemented in the open-source software word2vec[5] on the corpus.

5 https://code.google.com/archive/p/word2vec

This model is based on the idea that similar words are more likely to be neighbours if their word level embeddings represent lexical and semantic regularities. Thus, this model predicts words within a certain range to either side of a current word and captures syntactic and semantic regularities of words. We retrieve labels from the KG for each of our target relations as well as from WIKIDATA[6] and merge them for each relation. We call these labels "*seed labels*". Then, we sum up the vector representation of each of the seed labels to one vector, the seed vector. Thereafter, for each seed label we query OxfordDictionary,[7] Wordnik[8] and Wordnet[9] to find similar words. To reduce semantic drift in this step, we rearrange these similar words to the seed labels with the help of the vector representations. Hence, for each similar word we choose its vector representation and calculate the cosine similarity between this vector and all the seed vectors to measure the similarity. We rearrange all similar words to the relation where the cosine similarity between the seed vector of a relation and the vector of the similar word has the highest value.

6.3.5 *Generalization*

The input data of the generalization steps are the candidate trees $\mathcal{T}_r = \{T_1, T_2, \ldots\}$ for each target relation r as well as the results of the embedded semantics module. The goal is to generalize, filter, score and rank the candidate trees. Algorithm 1 and Algorithm 2 define the algorithm to generalize the extracted dependency parse trees.

Algorithm 1 takes two input parameters, i. e., two trees T_1, $T_2 \in \mathcal{T}_r$, and returns a generalized or an empty tree T. In the first line, T is initialized with an empty tree. In the next two lines, the root vertices of both trees T_1 and T_2 are preserved and Algorithm 2 is called to generalize the vertices of both trees. This tree with its generalized vertices is stored in T. In line 4, a set is generated containing all edge labels from outgoing edges of the root vertices that have the same edge labels in both trees. In lines 5 to 7, we iterate over all outgoing edges of the root vertices in the trees that have the same labels. For each combination, Algorithm 1 is recursively computed on the subtrees of vertices which are connected with root vertices in the trees that have the same labels. Here, the new root vertex of the generalize tree is preserved in line 8. Lines 10 to 13 show that only edges which do not subsume another edge are preserved. Finally, line 14 adds the edge to tree T.

Algorithm 2 defines the part of the algorithm to generalize vertices of two trees, T_1 and T_2. This function takes an empty or partly gen-

6 https://www.wikidata.org
7 https://www.oxforddictionaries.com
8 https://www.wordnik.com
9 https://wordnet.princeton.edu

Algorithm 1: `generalize`(T_1, T_2)

Input: Two trees, T_1 and T_2, to generalize
Output: A generalized or empty tree T

```
1  initialize T with V(T) ← ∅ and E(T) ← ∅;
2  v1 ← r̂_{T1}; v2 ← r̂_{T2};
3  generalizeVertices(T, T1, v1, T2, v2) ;
4  L ← {ψ_{T1}(v1, ṽ1) | (v1, ṽ1) ∈ E(T1) ∧ (v2, ṽ2) ∈ E(T2) :
     ψ_{T1}(v1, ṽ1) = ψ_{T2}(v2, ṽ2)} ;
5  foreach l in L do
6      foreach ṽ1 with (v1, ṽ1) ∈ E(T1) and ψ_{T1}(v1, ṽ1) = l do
7          foreach ṽ2 with (v2, ṽ2) ∈ E(T2) and ψ_{T2}(v2, ṽ2) = l do
8              ṽ ← r̂(generalize (T(ṽ1), T(ṽ2))) ;
9              add ← true ;
10             foreach v_p with (v, v_p) ∈ E(T) do
11                 if add then
12                     if T(v_p) ≤ T(ṽ) then add ← false ;
13                     if T(ṽ) < T(v_p) then E(T) ← E(T) \ {(v, v_p)};
14                 end
15             end
16             if add then E(T) ← E(T) ∪ {(v, ṽ)};
17         end
18     end
19 end
20 return T;
```

eralized tree T together with the trees T_1 and T_2 as well as the root vertices of these trees $v_1 = \hat{r}_{T_1}$ and $v_2 = \hat{r}_{T_2}$. The generalized tree is stored in T. In the first line, the function compares the root vertices with the root dependency vertices of the trees. If the vertices have the same labels, a new vertex is created with this label and is set as root along with the root dependency vertex. In case the given vertices differ from the root dependency vertices but have the same label, lemma or POS tag, a new vertex is created and is set with common attributes in lines 9 to 16. In lines 18 to 22, vertices without the same lemma and POS tags are compared and added to the generalized tree in cases where their other attributes are equal.

After the tree generalization steps, Ocelot filters false candidate tree patterns with the embedded semantics package. It retains tree patterns that contain one of the labels that occur in the label set of the corresponding target relation. Through the generalization process, the number of trees that a tree generalizes is observed. The trees are ranked by this number and by the number of vertices. Thus, a generalized tree that generalizes the most trees and has the fewest number of vertices has the highest rank.

Algorithm 2: `generalizeVertices`(T, T_1, v_1, T_2, v_2)

if $\hat{s}_{T_1} = v_1$ **and** $\hat{s}_{T_2} = v_2$ **then**
  if $\phi_{T_1,label}(v_1) = \phi_{T_2,label}(v_2)$ **then**
    initialize a new vertex v ;
    $\hat{s}_T \leftarrow v; \hat{r}(T) \leftarrow v$;
    $\phi_T, label(v) \leftarrow \phi_{T_1,label}(v_1)$;
    $V(T) \leftarrow V(T) \cup v$;
  end
else
  if $\phi_{T_1,lemma}(v_1) = \phi_{T_2,lemma}(v_2)$ **and** $\phi_{T_1,pos}(v_1) = \phi_{T_2,pos}(v_2)$ **then**
    initialize a new vertex v ;
    $\phi_T, lemma(v) \leftarrow \phi_{T_1,lemma}(v_1)$;
    $\phi_T, pos(v) \leftarrow \phi_{T_1,pos}(v_1)$;
    $V(T) \leftarrow V(T) \cup v$;
    if $\phi_{T_1,label}(v_1) = \phi_{T_2,label}(v_2)$ **then**
      $\phi_{T,label}(v) \leftarrow \phi_{T_1,label}(v_1)$;
      $\phi_{T,general}(v) \leftarrow label$;
    end
    else $\phi_{T,general}(v) \leftarrow lemma$;
  else
    foreach $a \in \{pos, ner, domain, range\}$ **do**
      if $\phi_{T_1,a}(v_1) = \phi_{T_2,a}(v_2)$ **then**
        $\phi_{T,general}(v) \leftarrow a$;
        $\phi_{T,a}(v) \leftarrow \phi_{T_1,a}(v_1)$;
        $V(T) \leftarrow V(T) \cup v$;
    end
  end
end

Figure 6.2 illustrates a generalized tree pattern on two example sentences: "*Bob was born and raised in Leipzig, Saxony.*" and "*Alice was born 2018 at Bellevue Hospital.*". The red edges in the dependency parse trees mark deleted edges, while the remaining edges, along with the linguistic annotations in bold font, symbolize the resulting generalized dependency parse tree pattern. NE arguments are illustrated in curly brackets and POS tags in square brackets.

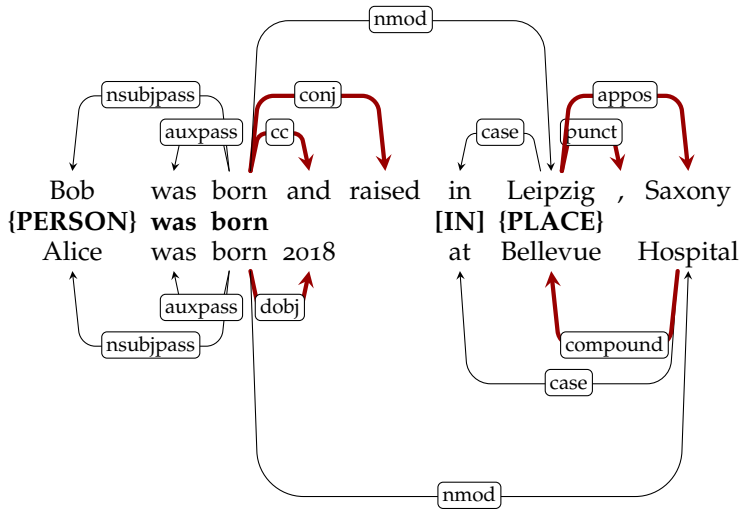


Figure 6.2: The generalization process on two example sentences.

6.4 EVALUATION

In this section, we present our experimental setup as well as the quantitative and qualitative evaluations we carried out.

6.4.1 *Setup*

For our experiments, we applied the English WIKIPEDIA as our corpus, i. e., $\mathscr{C}$, and DBPEDIA as our KG, i. e., $\mathcal{G}$. For the index, we chose the implementation of Apache Solr with Lucene. We then sampled the frequency distribution of the sentences' length (depicted in Figure 6.3) and filtered long and short sentences.

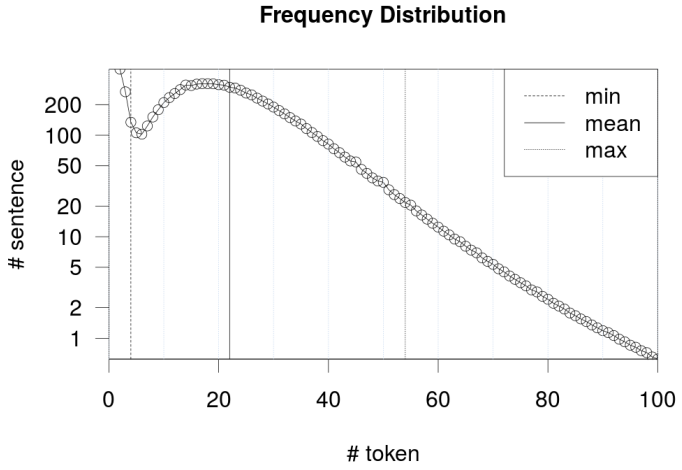


Figure 6.3: Frequency distribution of the sentences' length.

Thereafter our index contained 93,499,905 sentences in total with an average of $\mu = 22$ tokens per sentence and with a standard deviation of $\sigma = 15.8$ tokens. Our pipeline processed 88.44% of the sentences in the index with a maximum number of 54 tokens and not lesser than 4 tokens.

For the target relations queried from the DBpedia KG, we chose the top ten of each combination of resource types[10] we took into account. Thus, we ended up with 90 target relations. Table 6.1 depicts an excerpt with the top-three target relations of each combination from DBpedia.

Table 6.1: Excerpt of top-three predicates for each domain/range combination.

rdfs:domain \ rdfs:range	dbo:Organisation	dbo:Person	dbo:Place
dbo:Organisation	dbo:sisterStation	dbo:bandMember	dbo:hometown
	dbo:affiliation	dbo:formerBandMember	dbo:ground
	dbo:broadcastNetwork	dbo:notableCommander	dbo:headquarter
dbo:Person	dbo:almaMater	dbo:parent	dbo:deathPlace
	dbo:formerTeam	dbo:child	dbo:birthPlace
	dbo:debutTeam	dbo:spouse	dbo:nationality
dbo:Place	dbo:tenant	dbo:leaderName	dbo:district
	dbo:operator	dbo:architect	dbo:locatedInArea
	dbo:governingBody	dbo:saint	dbo:department

6.4.2 *Quantitative Evaluation*

In the quantitative evaluation, we first manually evaluated the filter approach that is based on the embedded semantics package to reduce semantic drift. Then, we compared the F-measure, Precision and Recall of the results with the embedded semantics filter with the results of two state-of-the-art systems, Patty and Boa.

Table 6.2: The number of trees and Precision, without filter (⊖) and with filter (⊛).

k	⊖		⊛	
	# trees	Pr [%]	# trees	Pr [%]
1	55	58.18	19	94.74
2	102	57.84	30	93.33
3	143	57.34	40	95.00
4	182	54.95	47	93.62
5	219	55.25	54	94.44

10 In our approach we utilize dbo:Organisation, dbo:Person and dbo:Place

The Precision, i. e., how many of the generalized trees express the correct relation for the top-k ranked generalized trees with and without the filter approach through the embedded semantics package, is depicted in Table 6.2. Each row shows the top-k ranked trees, i. e., sorted by the number of trees a generalized tree generalizes and the number of vertices in a tree. The columns with $\ominus$ denote the results without the filter and $\circledast$ with the filter. For instance, the top-1 ranked trees without filtering are 55 in total with $Pr_{\text{mac}} = 58.18\%$. With filtering, we obtain 19 top-1 ranked trees with $Pr_{\text{mac}} = 94.74\%$. Our results show that the Precision without filtering decreases with higher values of k but that the Precision with filtering remains more stable. For example, the Precision without filtering for $k = 5$ is 2.93% points lower than for $k = 1$ while it decreases by only 0.3% when filtering is used. Because of the significant increase of the Precision overall with filtering, we decided to filter the trees.

Table 6.3: Precision, Recall and F-measure averaged over `dbo:spouse`, `dbo:birthPlace`, `dbo:deathPlace` and `dbo:subsidiary` for the top k patterns. Best results are in bold font.

k	Boa	Patty	Ocelot
	Pr_{mac} [%]/Re_{mac} [%]/$F\text{-}score_{\text{mac}}$ [%]		
1	75.00/8.120/14.58	75.00/9.550/16.67	**100.0/13.12/22.92**
2	62.50/12.66/20.94	62.50/15.39/24.24	**87.50/21.23/33.64**
3	58.33/18.51/27.86	66.67/24.94/35.36	**91.67/34.35/48.93**
4	56.25/23.05/32.42	62.50/29.48/38.99	**91.67/40.19/54.73**
5	60.00/32.60/41.46	60.00/34.03/42.29	**86.67/43.77/56.55**

We manually compared the patterns of Boa and Patty with the generalized trees of our approach Ocelot. The results are depicted in Table 6.3. We manually assessed the patterns for each tool with the measures Precision, Recall and F-measure. To be comparable with the other systems we created a pattern-like representation from our trees. We compared the top 1–5 patterns for the four target relations (`dbo:spouse`, `dbo:birthPlace`, `dbo:deathPlace` and `dbo:subsidiary`) supported by all three systems. Our approach reached higher values on all five k for all three measures.

6.4.3 *Qualitative Evaluation*

We evaluated the quality of our approach against the state-of-the-art tool Boa. For the relation extraction with Boa, we chose $k = 10$ the top-10 patterns from the Boa index as well as the top-10 from Ocelot. We compared the relation extraction results of Boa and Ocelot on

the first 100 sentences of the top-three viewed articles about persons in WIKIPEDIA. The results are shown in Table 6.4.

Table 6.4: Qualitative relation extraction with BOA and OCELOT.

Examples	BOA	OCELOT
(PERSON) and his wife (PERSON)	×	**dbo:spouse**
(PERSON) and (PERSON) were married	×	**dbo:spouse**
(PERSON) met (PERSON)	dbo:spouse	dbo:spouse
(PERSON) was born in (PLACE)	dbo:deathPlace **dbo:birthPlace**	**dbo:birthPlace**
(PERSON) was born in 1905 in (PLACE)	×	**dbo:birthPlace**
(PERSON) returned to (PLACE)	dbo:deathPlace dbo:birthPlace	×
(PERSON) moved to (PLACE)	dbo:deathPlace dbo:birthPlace	×

We replaced NEs in sentences with their types as this is the preprocessing step for both tools. The × symbol indicates that the system found no relation in the sentence. The bold marked relation types in the table indicate correct extractions.

With BOA, we extracted one correct relation (*TP*), dbo:birthPlace, on one sentence, "*(PERSON) was born in (PLACE)*", but also a wrong extraction (*FP*) for dbo:deathPlace on the same sentence. With OCELOT, we extracted four correct relation (*TP*).

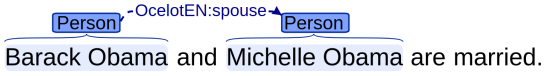


Figure 6.4: Visualized extraction results of BOA, PATTY and ours on an example sentence.

In contrast to tools such as BOA and PATTY, a benefit of our approach is that it finds relations that are not only enclosed by the NEs, e. g., Figure 6.4. Thus, we successfully extracted the relation in: "*(PERSON) and (PERSON) are married*" with our approach OCELOT, but neither with BOA nor PATTY.

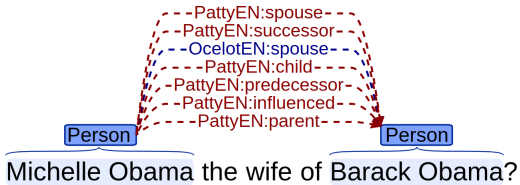


Figure 6.5: Visualized extraction results of BOA, PATTY and ours on an example question.

Another benefit of our approach is that it matches to exactly one relation if there is any match, hence, our approach OCELOT reduces semantic drift compared to the tools BOA and PATTY. Figure 6.5 exemplifies RE with the aforementioned two tools and ours on the example question mentioned in the thesis motivation (Section 1.1). With BOA, we found no match for a relation but we found matches with PATTY to six relations including the correct one. With our approach OCELOT, we found one match to the correct relation.

6.5 ERROR ANALYSIS

Data extracted from semi-structured sources, such as DBPEDIA, often contains inconsistencies as well as misrepresented and incomplete information [298]. For instance, at the time of writing this work, the DBPEDIA resource `dbr:England` is a subtype of `dbo:Person` and a `dbo:MusicalArtist`, instead of being an instance of `dbo:Place` and of `dbo:PopulatedPlace`. Consequently, the data used by our approach for distance supervision was partly erronenous. For example, the labels of `dbr:England` served as labels for target relations with person arguments (e. g., `dbo:spouse`), because `dbr:England` is of the wrong type in DBPEDIA. The integration of multiple KGs and a type check over multiple KGs could potentially solve this type mismatch.

The low Recall of OCELOT might be due to the missing coreference resolution system in the proposed approach. We aim to integrate such an approach into our framework in future works. Due to the filtering of trees with the embedded semantics package, it might be the case that trees counting as *TP* are filtered out because their semantic is not covered by the embedded semantics package. Increasing the number of external sources may increase the Recall of our system.

Part III

CHALLENGES AND APPLICATIONS

7

SEMANTIFYING CONTENT MANAGEMENT SYSTEMS

In this chapter[1], we describe our Semantic Content Management System (SCMS), whose main goals are the extraction of knowledge from unstructured data in any Content Management System (CMS) and the integration of the extracted knowledge into the same CMS. Our underlying framework integrates a highly accurate knowledge extraction pipeline. In addition, it relies on the Resource Description Framework [45, 136] (RDF) and Hypertext Transfer Protocol (HTTP) [74] standards for communication and can thus be integrated in virtually any CMS. We present how our framework is being used in the energy sector. We also evaluate our approach and show that our framework outperforms even commercial software by reaching up to 96% *F-score*.

7.1 RELATED WORK

Information Extraction is the backbone of knowledge extraction and is one of the core tasks of Natural Language Processing (NLP). Three main categories of NLP tools play a central role during the extraction of knowledge from text: Keyphrase Extraction (KE), Named Entity Recognition (NER) and Relation Extraction (RE). The automatic detection of keyphrases (i. e., multi-word units or text fragments that capture the essence of a document) has been an important task of NLP for decades. Still, due to the very ambiguous definition of what an appropriate keyphrase is, current approaches to the extraction of keyphrases still display low F-measures [132]. According to [131], the majority of the approaches to KE implement combinations of statistical, rule-based or heuristic methods [78, 202] on mostly document [167], keyphrase [274] or term cohesion features [213].

NER aims to discover instances of predefined types of entities (e. g., persons, locations, organizations or products) in text. Most NER tools implement one of three main categories of approaches: dictionary-based [8, 281], rule-based [38, 267] and machine-learning

¶ Parts of this chapter have been published as conference article [198]. The thesis author co-developed the design of the solution, source code and co-wrote the publication together with the main author.

1 Throughout this chapter, we use the prefix `rdf`, `rdfs`, `xsd`, `content`, `dc`, `sioc`, `scmsann`, `ctag`, `ann` and `scms` for the Internationalized Resource Identifiers (IRIs) `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, `http://www.w3.org/2001/XMLSchema#`, `http://purl.org/rss/1.0/modules/content/`, `http://purl.org/dc/elements/1.1/`, `http://rdfs.org/sioc/ns#`, `http://ns.aksw.org/scms/annotations/`, `http://commontag.org/ns#`, `http://www.w3.org/2000/10/annotation-ns#` and `http://ns.aksw.org/scms/`.

approaches [188]. Nowadays, the methods of choice are borrowed from supervised machine learning when training examples are available [43, 75, 305]. Yet, due to scarcity of large domain-specific training corpora, semi-supervised [188, 216] and unsupervised machine learning approaches [66, 190] have also been used for extracting Named Entities (NEs) from text.

The extraction of relations from unstructured data builds upon work for NER and KE to determine the entities between which relations might exist. Some early work on pattern extraction relied on supervised machine learning [94]. Yet, such approaches demanded large amount of training data. The subsequent generation of approaches to RE aimed at bootstrapping patterns based on a small number of input patterns and instances [5, 24]. Newer approaches aim to either collect redundancy information from the whole Web [212] or Wikipedia [282, 293] in an unsupervised manner or to use linguistic analysis [102, 201] to harvest generic patterns for relations.

In addition to the work done by the NLP community, several tools and frameworks have been developed explicitly for extracting RDF and RDF in Attributes [3] (RDFa) out of text sources [4]. For example, the Firefox extension Piggy Bank [120] allows to extract RDF from web pages by using screen scrapers. The RDF extracted from these webpages is then stored locally in a Sesame store. The data being stored locally allows the user to merge the data extracted from different websites to perform semantic operations. More recently, the Drupal extension OpenPublish[2] was released. The aim of this extension is to support content publishers with the automatic annotation of their data. For this purpose, OpenPublish utilizes the services provided by OpenCalais[3] to annotate the content of news entries. Epiphany [4] implements a service that annotates web pages automatically with entities found in the Linked Open Data (LOD)-Cloud. Apache Stanbol[4] implements similar functionality on a larger scale by providing synchronous RESTful interfaces that allow CMSs to extract annotations from text.

The main drawback of current frameworks is that they either focus on one particular task (e. g., finding NEs in text) or make use of NLP algorithms without improving upon them. Consequently, they have the same limitations as the NLP approaches discussed above. To the best of our knowledge, our framework is the first framework designed explicitly for the purposes of the Semantic Web (SW) that combines flexibility with accuracy. The flexibility of the SCMS has been shown by its deployment on Drupal[5], Typo3[6] and conX[7]. In

2 http://www.openpublish.com
3 http://www.opencalais.org
4 http://incubator.apache.org/stanbol
5 http://drupal.org
6 http://typo3.org
7 http://conx.at

addition, our framework is able to extract RDF from text sources with an accuracy superior to that of commercial systems as shown by our evaluation. Our framework also provides a machine-learning module that allows to tailor it to new domains and types of NEs. Moreover, SCMS provides dedicated interfaces for interacting (e. g., editing, querying, merging) with the triples extracted, making it usable in a large number of domains and use cases.

7.2 THE SEMANTIC CONTENT MANAGEMENT SYSTEM

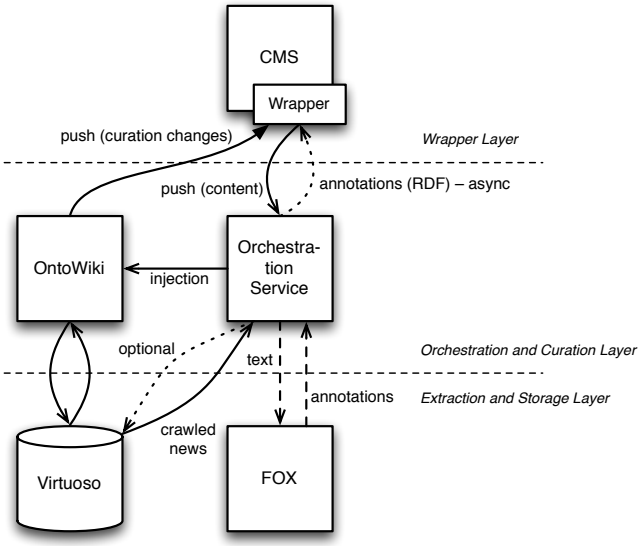


Figure 7.1: Architecture and communication paths of the components in our SCMS.

An overview of the architecture behind our SCMS is given in Figure 7.1. Our underlying framework consists of two layers: an *orchestration and curation* layer and an *extraction and storage* layer. The CMS that is to be extended with semantic capabilities resides upon our framework and must be extended minimally via a *CMS wrapper*. This extension implements the in- and output behavior of the CMS and communicates exclusively with the first layer of our framework, thus making the components of the extraction and storage layer of our framework swappable without any drawback for the users.

The overall goal of the first layer of our SCMS framework is to coordinate the access to the data. It consists of two tools: the orchestration service and the data wiki OntoWiki. The *orchestration service* is the input gate of SCMS. It receives the data that is to be annotated as a RDF message that abides by the vocabulary presented in Section 7.3.2 and returns the results of the framework to the endpoint specified in the RDF message it receives. OntoWiki provides functionality for the manual curation of the results of the knowledge extraction process

and manages the data flow to the *triple store* VIRTUOSO[8], the first component of the *extraction and storage layer*. In addition to a triple store, the second layer contains FOX[9], that uses machine learning to combine and improve upon the results of NLP tools as well as converts these results into an RDF graph by using the vocabularies displayed in Section 7.3.3. VIRTUOSO also contains a crawler that allows to retrieve supplementary knowledge from the Web and link it to the information already available in the CMS by integrating it into the CMS.

In the following, we present the central components of the SCMS stack in more detail.

7.3 TOOLS AND VOCABULARIES

In this section we describe the main components of the SCMS stack and how they fit together. As running example, we use a hypothetical content item contained in a DRUPAL CMS. This node (in DRUPAL terminology) consists of two parts:

- The `title` "*Prometeus*" and
- a `body` that contains the sentence "*The company Prometeus is an energy provider located in the capital of Hungary, i. e., Budapest.*".

Only the body to the content item is to be annotated by the SCMS stack. Note that for reasons of brevity, we will only show the results of the extraction of NEs. Yet, SCMS can also extract keywords, keyphrases and relations.

7.3.1 *Wrapper*

A CMS wrapper (short wrapper) is a component that is tightly integrated into a CMS (see Figure 7.2) and whose role is to ensure the communication between the CMS and the orchestration module of our framework. In this respect, a wrapper has to fulfill three main tasks:

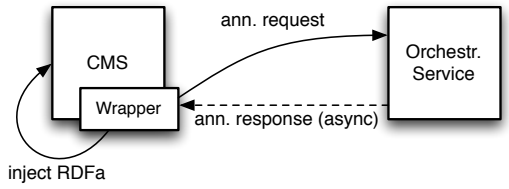


Figure 7.2: Architecture of communication between wrapper, CMS and orchestration service.

8 https://virtuoso.openlinksw.com/
9 https://dice-research.org/FOX

1. *Request generation*: Wrappers usually register for change events to the CMS editing system. Whenever a document has been edited, they generate an annotation request that abides by the vocabulary depicted in Figure 7.3. This request is then sent to the orchestration service.

2. *Response receipt*: Once the annotation has been carried out, the annotation results are sent back to the wrapper. The second of the wrapper's main tasks is consequently to react to those annotation responses and to store the annotations to the document appropriately (e. g., in a triple store). Since the annotation results are sent back asynchronously (i. e., in a separate request), the wrapper must provide a callback Uniform Resource Locator (URL) for this purpose.

3. *Data processing*: Once the data have been received and stored, wrappers usually integrate the annotations into the content items that were processed by the CMS. The integration of annotations is most commonly carried out by injecting the annotations as RDFa into the document's Hyper Text Markup Language (HTML) rendering. The data injection is mostly realized by registering to document viewing events in the respective CMS and writing the RDFa from the wrapper's local triple store into the content items that are being viewed.

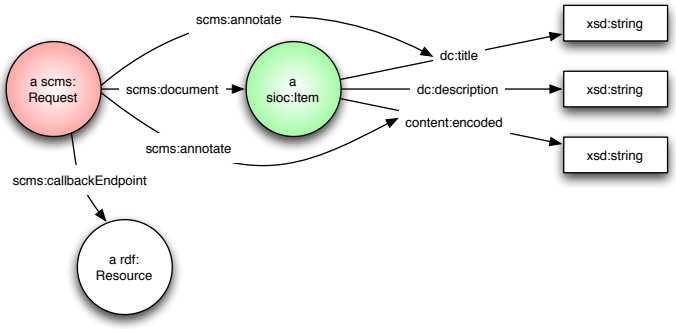


Figure 7.3: Vocabulary used by the wrapper requests.

An example of a wrapper request for our example is shown in Listing 7.1. The `content:encoded` of the Drupal node[10] is to be annotated by Fox. In addition, the whole node is to be stored in the triple store for the purpose of manual processing. Note that the wrapper can choose not to send portions of the content item that are not to be stored in the triple store, e. g., private data. In addition, note that the description of a document is not limited to certain properties or to a certain number thereof, which ensures the high level of flexibility of

10 `http://example.com/drupal/node/10`

the SCMS stack. Moreover, the RDF data extracted by SCMS can be easily merged with any structured information provided natively by the CMS (i. e., metadata such as author information). Consequently, SCMS enables CMS that already provide metadata as RDF to answer complex questions that combine data and metadata, e. g., "*Which authors wrote documents that are related to Budapest?*"

```
@prefix content: <http://purl.org/rss/1.0/modules/content/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@base <http://ns.aksw.org/scms/> .

<http://example.com/wrapperRequest/1> a <Request> ;
   <document> <http://example.com/drupal/node/10> ;
   <callbackEndpoint> <http://example.com/wrapper> ;
   <annotate> content:encoded .

<http://example.com/drupal/node/10> a sioc:Item ;
   dc:title "Prometeus" ;
   content:encoded "The company Prometeus is an energy provider
       located in the capital of Hungary, i.e., Budapest." .
```

Listing 7.1: Example annotation request as sent by the Drupal wrapper.

7.3.2 *Orchestration Service*

The main tasks of the orchestration service are to capture state information and to distribute the data across SCMS' layers. The first of the tasks is due to the Fox framework having been designed to be stateless. The orchestration service captures state information by splitting up each document-based annotation requests by a wrapper into several property-based annotation requests that are sent to Fox. In our example, the orchestration service detects that solely the `content:encoded` property is to be annotated. Then, it reads the content of that property from the wrapper request and generates the annotation request "*The company Prometeus is an energy provider located in the capital of Hungary, i. e., Budapest.*" for Fox. Note that while this property-based annotation request consists exclusively of text or HTML and does not contain any RDF, the response returned by Fox is a RDF document serialized in `RDF/TURTLE` or `RDF/XML`.

The annotation results returned by Fox are combined by the orchestration service into the annotation response. Therewith, the relation between the input document and the annotations extracted by Fox is re-established. When all annotations for a particular request have been received and combined, the annotation response is sent back to the wrapper via the provided callback URL. In addition, the results sent back to the wrapper are stored in OntoWiki to facilitate the curation of annotations extracted automatically. The annotation response generated by the orchestration service for our example is

shown in Listing 7.2. It relies upon the output sent by Fox. The exact meaning of the predicates used by Fox and forwarded by the orchestration service are explained in Section 7.3.3.

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix ctag: <http://commontag.org/ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ann: <http://www.w3.org/2000/10/annotation-ns#> .
@prefix scms: <http://ns.aksw.org/scms/> .
@prefix scmsann: <http://ns.aksw.org/scms/annotations/> .
@prefix tools: <http://ns.aksw.org/scms/tools/> .
@prefix content: <http://purl.org/rss/1.0/modules/content/> .

[] a ann:Annotation , scmsann:LOCATION ;
    scms:annotates <http://example.com/drupal/node/10> ;
    scms:property content:encoded ;
    scms:beginIndex "70"^^xsd:int ;
    scms:endIndex "77"^^xsd:int ;
    scms:means dbr:Hungary ;
    scms:source tools:FOX ;
    ann:body "Hungary"^^xsd:string .

[] a ann:Annotation , scmsann:ORGANIZATION ;
    scms:annotates <http://example.com/drupal/node/10> ;
    scms:property content:encoded ;
    scms:beginIndex "12"^^xsd:int ;
    scms:endIndex "21"^^xsd:int ;
    scms:means <http://scms.eu/Prometeus> ;
    scms:source tools:FOX ;
    ann:body "Prometeus"^^xsd:string .

[] a ann:Annotation , scmsann:LOCATION ;
    scms:annotates <http://example.com/drupal/node/10> ;
    scms:property content:encoded ;
    scms:beginIndex "85"^^xsd:int ;
    scms:endIndex "93"^^xsd:int ;
    scms:means dbr:Budapest> ;
    scms:source tools:FOX ;
    ann:body "Budapest"^^xsd:string .
```

Listing 7.2: Example annotation response as sent by the orchestration service.

7.3.3 *FOX*

The Fox framework is a stateless and extensible framework that encompasses all the NLP functionality necessary to extract knowledge from the content of CMS. Its architecture consists of three layers as shown in Figure 7.4. Fox takes text or HTML as input. This data is sent to the *controller layer*, which implements the functionality necessary to clean the data, i. e., remove HTML and Extensible Markup Language (XML) tags as well as further noise. Once the data has been cleaned, the controller layer begins with the orchestration of the tools in the *tool layer*. Each of the tools is assigned a thread from a thread pool, so as to maximize usage of multi-core CPUs. Every thread runs its tool and generates an event once it has completed its computation. In the event

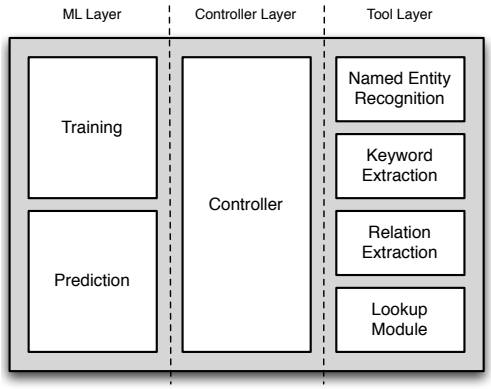


Figure 7.4: Architecture of the Fox framework.

that a tool does not complete after a set time, the corresponding thread is terminated. So far, Fox integrates tools for extracting keywords, entities and relations. The keyword extraction is realized by PoolParty[11] for extracting keywords from a controlled vocabulary, KEA[12] and the Yahoo Term Extraction service[13] for statistical extraction and several other tools. In addition, Fox integrates the Stanford Named Entity Recognizer[14] [75], the Illinois Named Entity Tagger[15] [230] and commercial software for NER. The RE is carried out by using the CARE platform[16].

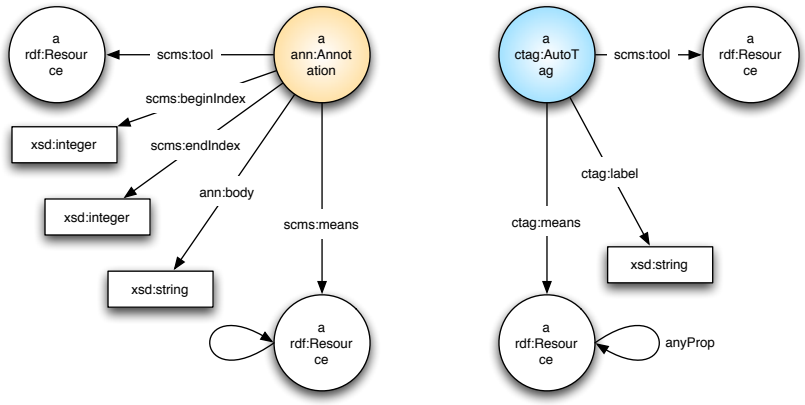
The results from the tool layer are forwarded to the *prediction module* of the *machine-learning layer*. The role of the prediction module is to generate Fox's output based on the output of the tools in Fox's backend. For this purpose, it implements several ensemble learning techniques [57] with which it can combine the output of several tools. Currently, the prediction module carries out this combination by using a feed-forward neural network. The neural network inserted in Fox was trained by using 117 news articles. It reached 89.21% F-measure in an evaluation based on a ten-fold-cross-validation on NER, therewith outperforming even commercial systems[17].

Once the neural network has combined the output of the tool and generated a better prediction of the NEs, the output of Fox is generated by using the vocabularies shown in Figure 7.5. These vocabularies extend the two broadly used vocabularies Annotea[18] and Autotag [19]. In particular, we added the constructs explicated in the following:

11 http://poolparty.biz
12 http://www.nzdl.org/Kea/
13 http://developer.yahoo.com/search/content/V1/termExtraction.html
14 http://nlp.stanford.edu/software/CRF-NER.shtml
15 http://cogcomp.cs.illinois.edu/page/software_view/4
16 http://www.digitaltrowel.com/Technology/
17 More details on the evaluation are provided at https://dice-research.org/FOX
18 http://www.w3.org/2000/10/annotation-ns#
19 http://commontag.org/ns#

- `scms:beginIndex` denotes the index in a literal value string at which a particular annotation or keyphrase begins;
- `scms:endIndex` stands for the index in a literal value string at which a particular annotation or keyphrase ends;
- `scms:means` marks the Uniform Resource Identifier [17] (URI) assigned to a NE identified for an annotation;
- `scms:source` denotes the provenance of the annotation, i. e., the URI of the tool which computed the annotation or even the system ID of the person who curated or created the annotation and
- `scmsann:<class>` is the namespace for the annotation classes, i. e., the entity types location, person, organization and miscellaneous.

Given that the overhead due to the merging of the results via the neural network is of only a few milliseconds and thank to the multi-core architecture of current servers, Fox is almost as time-efficient as state-of-the-art tools. Still, as our evaluation shows, these few milliseconds overhead can lead to an increase of more than 13% F-measure (see Section 7.5). The output of Fox for our example is shown in Listing 7.3. This is the output that is forwarded to the orchestration service, which adds provenance information to the RDF before sending an answer to the callback URI provided by the wrapper. By these means, we ensure that the wrapper can write the RDFa in the write segment of the item content.



(a) named entity annotation (b) keyword annotation

Figure 7.5: Vocabularies used by Fox for representing NEs (a) and keywords (b)

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix ctag: <http://commontag.org/ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ann: <http://www.w3.org/2000/10/annotation-ns#> .
@prefix scms: <http://ns.aksw.org/scms/> .
@prefix scmsann: <http://ns.aksw.org/scms/annotations/> .
@prefix tools: <http://ns.aksw.org/scms/tools/> .


[] a ann:Annotation , scmsann:LOCATION ;
     scms:beginIndex "70"^^xsd:int ;
     scms:endIndex "77"^^xsd:int ;
     scms:means dbr:Hungary ;
     scms:source tools:FOX ;
     ann:body "Hungary"^^xsd:string .

[] a ann:Annotation , scmsann:ORGANIZATION ;
     scms:beginIndex "12"^^xsd:int ;
     scms:endIndex "21"^^xsd:int ;
     scms:means <http://scms.eu/Prometeus> ;
     scms:source tools:FOX ;
     ann:body "Prometeus"^^xsd:string .

[] a ann:Annotation , scmsann:LOCATION ;
     scms:beginIndex "85"^^xsd:int ;
     scms:endIndex "93"^^xsd:int ;
     scms:means dbr:Budapest ;
     scms:source tools:FOX ;
     ann:body "Budapest"^^xsd:string .
```

Listing 7.3: Annotations as returned by Fox in RDF/TURTLE format.

7.3.4 *OntoWiki*

OntoWiki is a semantic data wiki [11] that was designed to facilitate the browsing and editing of RDF graphs. Its browsing features range from arbitrary concept hierarchies to facet-based search and query building interfaces. Semantic content can be created and edited by using the RDFauthor system which has been integrated in OntoWiki [270].

OntoWiki plays two key roles within the SCMS stack. First, it serves as entry point for the triple store. This allows for the triple store to be exchanged without any drawback for the user, leading to an easy customization of our stack. In addition, OntoWiki plays the role of an annotation consolidation and curation tool and is consequently the center of the curation pipeline. To ensure that OntoWiki is always up-to-date, the orchestration service sends its annotation responses to both OntoWiki and the wrapper's callback URI. Thus, OntoWiki is also aware of the wrapper (i. e., its callback URI) and can send the results of any manual curation process back to wrapper. Note that manually curated annotations are saved with a different (if manually created) or supplementary (if manually curated) value in their `scmsann:source` property. This gives consuming tools (e. g., wrappers) a chance to

assign higher trust values to those annotations. In addition, if a new extraction run is performed on the same document, manually created and curated annotations can be kept for further use. Note that the crawler in VIRTUOSO can be used to fetch even more data pertaining to the annotations computed by FOX. This data can be sent directly to FOX and inserted in the VIRTUOSO triple store so as to extend the *semantic data integration* and thus, the usable knowledge for the CMS.

7.4 USE CASE

The SCMS framework is being deployed in the renewable energy sector. The renewable energy and energy efficiency sector requires a large amount of up-to-date and high-quality information and data so as to develop and push the area of clean energy systems worldwide. This information, data and knowledge about clean energy technologies, developments, projects and laws per country worldwide helps policy and decision makers, project developers and financing agencies to make better decisions on investments as well as clean energy projects to set up. The REEEP – the Renewable Energy and Energy Efficiency Partnership[20] is a non-governmental organization that provides the aforementioned information to the respective target groups around the globe. For this purpose, REEEP has developed the `reegle.info` Information Gateway on Renewable Energy and Energy Efficiency [21] that offers country profiles on clean energy, an Actors Catalog that contains the relevant stakeholders in the field per country. Furthermore, it supplies energy statistics and potentials as well as news on clean energy.

The motivation behind applying SCMS to the REEEP data was to facilitate the integration of this data in semantic applications to support efficient decision-making. To achieve this goal, we aimed to expand the `reegle.info` information gateway by adding RDFa to the unstructured information available on the website and by making the same triples available via a SPARQL Protocol And RDF Query Language [37] (SPARQL) endpoint. For our current prototype, we implemented a CMS wrapper for the DRUPAL CMS and imported the actors catalog of reegle within in (see Figure 7.6). This data was then processed by the SCMS stack as follows: All actors and country descriptions were sent to the orchestration service, which forwarded them to FOX. The RDF graph extracted by FOX were sent back to the DRUPAL Wrapper and written via ONTOWIKI into VIRTUOSO. The DRUPAL wrapper then used the keyphrases to extend the set of tags assigned to the corresponding profile in the CMS. The NEs were integrated in the page by using the positional information returned by FOX. By these means, we made the REEEP data accessible for humans

20 `http://www.reeep.org`
21 `http://www.reegle.info`

Figure 7.6: Screenshots of SCMS-enhanced Drupal

(via the Web page) but also for machines (via OntoWiki's integrated SPARQL endpoint and via the RDFa written in the Web pages).

Our approach also makes the automated integration of novel knowledge sources in REEEP possible. To achieve this goal, several selected sources (web sources, blogs and news feeds) are currently being crawled and then analyzed by Fox to extract structured information out of the masses of unstructured text from the Internet.

7.5 EVALUATION

The usability of our approach depends heavily on the quality of the knowledge returned via automated means. Consequently, we evaluated the quality of the RDFa injected into the REEEP data by measuring the precision and recall of SCMS and compared it with that of a state-of-the-art commercial system (CS) whose name cannot be revealed for legal reasons. We chose CS because it outperformed freely available NER tools such as the Stanford Named Entity Recognizer[22] [75] and the Illinois Named Entity Tagger[23] [230] in a prior evaluation on a newspaper corpus. Within that evaluation, Fox reached 89.21% F-measure and was 14% better than CS w.r.t. F-measure[24].

Our evaluation was carried out with two different datasets and we followed a token-wise evaluation of the SCMS system (Section 2.4). In our first evaluation, we measured the performance of both systems on country profiles crawled from the Web, i. e., on information that is to be added automatically to the REEEP knowledge bases. For this

22 http://nlp.stanford.edu/software/CRF-NER.shtml
23 http://cogcomp.cs.illinois.edu/page/software_view/4
24 More details at https://dice-research.org/FOX

Table 7.1: Evaluation results on country and actors profiles. The superior F-measure for each category is in bold font.

		Country Profiles		Actors Profiles	
τ	Measure	Fox	CS	Fox	CS
LOCATION	*Pr* [%]	98.00	100.0	83.33	100.0
	Re [%]	94.23	78.85	90.00	70.00
	F-score [%]	**96.08**	88.17	**86.54**	82.35
ORGANIZATION	*Pr* [%]	73.33	100.0	57.14	90.91
	Re [%]	68.75	40.00	69.23	47.44
	F-score [%]	**70.97**	57.14	**62.72**	62.35
PERSON	*Pr* [%]	–	–	100.0	100.0
	Re [%]	–	–	45.45	54.55
	F-score [%]	–	–	62.50	**70.59**
Overall $\mathscr{T}$	Pr_{mic} [%]	93.97	100.0	85.16	98.20
	Re_{mic} [%]	91.60	74.79	70.64	52.29
	$F\text{-}score_{\text{mic}}$ [%]	**92.77**	85.58	**77.22**	68.24
Overall $\mathscr{T}$	Pr_{mac} [%]	85.67	100.0	80.16	96.97
	Re_{mac} [%]	81.49	59.43	68.23	57.33
	$F\text{-}score_{\text{mac}}$ [%]	**83.53**	72.66	70.59	**71.76**

purpose, we selected 9 country descriptions randomly and annotated 34 sentences manually. These sentences contained 119 NEs tokens, of which 104 were locations and 15 organizations. In our second evaluation, we aimed at measuring how well SCMS performs on the data that can be found currently in the REEEP catalogue. For this purpose, we annotated 23 actors profiles which consisted of 68 sentences manually. The resulting reference data contained 20 location, 78 organization and 11 person tokens. Note that both datasets are of very different nature as the first contains a large number of organizations and a relatively small number of locations while the second consists mainly of locations.

The results of our evaluation are shown in Table 7.1. CS follows a very conservative strategy, which leads to it having very high Precision scores of up to 100% in some experiments. Yet, its conservative strategy leads to a Recall which is mostly significantly inferior to that of SCMS. The only category within which CS outperforms SCMS is the detection of persons in the actors profile data, i. e., $\tau = \texttt{PERSON}$. This is due to it detecting 6 out of the 11 person words in the dataset, while SCMS only detects 5. In all other cases, SCMS outperforms CS by up to 13%

F-measure in the detection of organizations in the country profiles dataset, i. e., $\tau = \texttt{ORGANIZATION}$.

Overall, i. e., $\textit{F-score}_{\text{mic}}$, SCMS outperforms CS by 7% F-measure on country profiles and almost 9% F-measure on actors profiles. Regarding $\textit{F-score}_{\text{mac}}$, SCMS outperforms CS by almost 11% F-measure on country profiles but on the actors profiles CS is slightly better due to the measurement treating all datasets equally independent of the dataset sizes (Section 2.3.3).

8

KNOWLEDGE GRAPH POPULATION — ATTRIBUTE PREDICTION AND VALIDATION

In this chapter[1], we introduce in the Semantic Web (SW) Challenge 2017 and our system, LEOPARD, that participated in this challenge. With LEOPARD, we provided the baseline approach included in the benchmarking platform to provide participants with a reference point to monitor their performance.

8.1 RELATED WORK

This work follows the definition of Knowledge Base Population (KBP) and Knowledge Base Refinement (KBR) in [91]. KBP extends a knowledge base that already consisting of a substantial amount of knowledge with information extracted from text. KBR gathers evidence for or against statements in the KB from text and estimates a confidence for existing statements. A survey of approaches, with a dual look at both the methods being proposed and the evaluation methodologies being used, is given in [218].

State of the art approaches in this area [241, 251] are usually based on Information Extraction components, for instance, Named Entity Recognition (NER), Entity Linking (EL) and Relation Extraction (RE). Those components typically trained with additional instances provided by linked open data and thus reducing the need of manual annotated datasets [178]. A survey of Information Extraction approaches in the context of the SW is provided in [164].

SOCRATES [92] is a web-scale knowledge extraction engine that exploits a mix of deep learning, SW technology and Natural Language Processing (NLP) to understand information in text and integrate knowledge extracted from various sources. The approach is based on supervised deep learning and combines a convolutional neural network with max pooling, word and position embeddings, bag of

¶ Parts of this chapter have been published as journal article [263]. The author of this thesis is also the main author of the article and developed the main ideas, designed, and implemented major parts of the solution, and wrote the majority of the publication.

1 Throughout this chapter, we use the prefix `rdf`, `rdfs`, `xsd`, `permid`, `vcard`, `foaf`, `mdaas`, `owl`, `og`, `permid` and `permidOrg` for the Internationalized Resource Identifiers (IRIs) `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, `http://www.w3.org/2001/XMLSchema#`, `http://permid.org/`, `http://www.w3.org/2006/vcard/ns#`, `http://xmlns.com/foaf/spec/`, `http://ont.thomsonreuters.com/mdaas/`, `http://www.w3.org/2002/07/owl#`, `https://ogp.me/ns#`, `http://permid.org/` and `http://permid.org/ontology/organization/`.

words, NER and EL with DBpedia Spotlight [46], DuckDuckGo[2] and regular expressions. At the time of writing, Socrates is a closed source system from IBM Research.

In contrast to Socrates, Leopard is an open source system that uses a mixture of heuristics for information extraction without embeddings and without deep learning. Leopard is a rule-based supervised approach with a simple precision ranking of the heuristics, thus it selects the best performing heuristic for a specific extraction task it integrates. It utilizes, among other things, the extended multilingual version of Fox [259–261] for NER and EL in five languages (DE, EN, ES, FR, NL).

DeFacto [88, 256], Deep Fact Validation, is an algorithm for validating statements by finding confirming sources for it on the web. It is a system that generates natural language out of Resource Description Framework [45, 136] (RDF) by relying on the Boa framework [89] which uses a database of lexicalizations for predicates in DBpedia. Based on those lexicalizations, DeFacto transforms statements in DBpedia to natural language sentences. It uses the Bing web search engine to find web pages containing those sentences with the aim of computing the trustworthiness of RDF triples by using the Web as background knowledge. RDF triples without or only a very few web pages supporting the corresponding sentences are then assigned a low confidence score. Because Boa, integrated in DeFacto, is not supporting the relations applied in the 2017 SW Challenge, Leopard is not using DeFacto.

8.2 SEMANTIC WEB CHALLENGE 2017

In this section, we give an overview of the dataset and the two SW Challenge tasks. Then, we present the challenge evaluation and benchmarking process.

8.2.1 *Dataset*

The evaluation of participating systems in the challenge was carried out on a Knowledge Graph (KG) owned and exposed by Thomson Reuters. The KG is currently composed of a public part exposed at `http://permid.org` and a private part. While the participants were free to use any dataset of their choice for training, the core training dataset was the public part of the Thomson Reuters KG. This graph provided the instance knowledge necessary to derive an implicit understanding of the semantics of the relations at the core of the two tasks. A portion of the private part of the graph was integrated as ground truth in Gerbil and used as test data during the benchmarking phase of the challenge.

2 `https://duckduckgo.com`

8.2.2 *Task 1: Attribute Prediction*

In this task, the data consisted of statements about subject entities of the type organization. The task's aim was to predict the values for three attributes.

```
@prefix permid: <http://permid.org/> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .

permid:1-5045055688
  vcard:organization-name "Servicemaster Home Service Center LLC" ;
  vcard:hasURL <http://www.servicemasterclean.com> .
```

Listing 8.1: An excerpt of the task 1 example data.

```
@prefix mdaas: <http://ont.thomsonreuters.com/mdaas/> .
@prefix permid: <http://permid.org/> .
@prefix permidOrg: <http://permid.org/ontology/organization/> .

permid:1-5045055688
  mdaas:isDomiciledIn "United States"@en ;
  permidOrg:hasHeadquartersPhoneNumber "00019015973000" ;
  permidOrg:hasLatestOrganizationFoundedDate "1999" .
```

Listing 8.2: An excerpt of the task 1 example result data.

In more detail, the name and website Uniform Resource Locator (URL) of organizations were given as input (see Listing 8.1) and the goal of this task was to provide the values for the country in which an organization's headquarters is located, the headquarters phone number and the latest date of incorporation of an organization (see Listing 8.2).

8.2.3 *Task 2: Attribute Validation*

In task two, the input data consisted of statements about entities of the type organization. The task's aim was to provide an assessment of the correctness of the statements about the given entities.

```
@prefix mdaas: <http://ont.thomsonreuters.com/mdaas/> .
@prefix permid: <http://permid.org/> .
@prefix permidOrg: <http://permid.org/ontology/organization/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .

permid:1-5009196497
  vcard:organization-name "Wall Trojan Products" ;
  mdaas:isDomiciledIn "United States"@en .

<http://swc2017.aksw.org/task2/dataset/s-789147186>
  a rdf:Statement ;
  rdf:subject permid:1-5009196497 ;
  rdf:predicate permidOrg:hasHeadquartersPhoneNumber;
  rdf:object "00019169202121" .
```

Listing 8.3: An excerpt of the task 2 example data.

```
<http://swc2017.aksw.org/task2/dataset/s-789147186>
  <http://swc2017.aksw.org/hasTruthValue>
    "0.0"^^<http://www.w3.org/2001/XMLSchema#double> .
```

Listing 8.4: An excerpt of the task 2 example result data.

In particular, the name of an organization and the country in which the organization's headquarters is located were given. In addition, the headquarters phone number, the website URL and the latest date of incorporation of the organization were given (see Listing 8.3) to validate the correctness by providing a confidence score between 0 and 1 for the assessment (see Listing 8.4).

8.2.4 *Evaluation and Benchmarking*

The challenge evaluation and benchmarking process was carried out by the open source benchmarking platform GERBIL [238, 279] that follows the FAIR data principles [289] (FAIR). The Key Performance Indicator (KPI) for the evaluation on the first task was the F-measure. For the second task, the Area under the Receiver Operating Characteristic Curve (AUC-ROC) was used. For benchmarking, additionally measurements were provided by the platform for instance, Precision, the number of documents that cause errors $\lightning$, the average milliseconds to handle a document and the AUC-ROC as metrics to evaluate the performance of the algorithms. To participate in the challenge, the results of at least one of the two tasks needed to be uploaded through the user interface of the platform.

8.3 APPROACH

In the following, we present our baseline solution, LEOPARD. We begin with an overview of the data flow through the packages it relies upon. Subsequently, we provide details of each package.

8.3.1 *Overview of the Method*

Figure 8.1 depicts the data flow through our system. LEOPARD consists of three main packages, *Data Acquisition*, *Attribute Extraction* and *Scoring and Ranking*. The fourth package, *Attribute Validation*, is based on the combination of the other packages and is thus not present in the figure.

With the first package, *Data Acquisition*, we gathered all website URLs from the given test data and crawled each website in the Document Web to store parts of the website content. Then, we applied diverse text extraction modules, which are integrated in the *Attribute Extraction* package, to extract values from the stored website contents for four attributes. With the *Scoring and Ranking* package, we

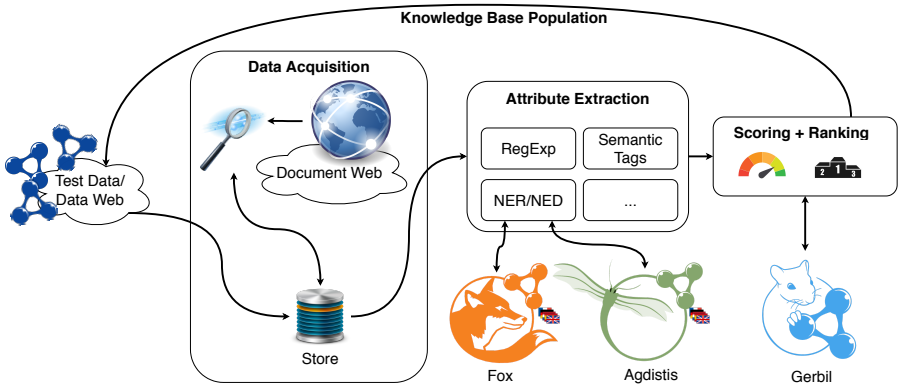


Figure 8.1: Data flow through our system LEOPARD.

scored each extraction module by its performance and used these performance scores to rank the extraction modules.

The extracted attribute values from the highest ranked modules were chosen. We reused this three main packages in the fourth packages *Attribute Validation* for task two and created an assessment of the correctness of the extracted values.

8.3.2 *Data Acquisition*

For both tasks, we gathered the given website URLs from the test data. For each entity in the test data, we stored the URL as well as additional information given in the task's data (e. g., in task one, the organization name and in task two, the county in which the organization's headquarters is located, the given phone number). Then, we crawled each website with its subpages and added the website contents together with its markups to our store. We chose the subpage URLs by the order they occurred on a website, i. e., we took the content within the Hyper Text Markup Language (HTML) body tag of a website together with the menu structure into account and searched within this content for URLs with the same domain. We repeated this process for URLs we found on the website up to a fix number of URLs, thus decreased the run time of this data acquisition process by limiting the number of URLs.

8.3.3 *Attribute Extraction*

This is the core package of LEOPARD and consists of diverse text extraction methods. We applied this package on the stored website contents to predict the values for the four attributes:

1. `permidOrg:hasHeadquartersPhoneNumber`
2. `permidOrg:hasLatestOrganizationFoundedDate`
3. `vcard:hasURL`

4. `mdaas:isDomiciledIn`

Thus, this package consists of four types of extraction modules, each for a specific attribute. We give insights in each module in the following subsections.

8.3.3.1 *hasHeadquartersPhoneNumber*

For both tasks we tried to extract the headquarters phone number from the websites. The extraction was conducted by searching the subpage contents of a website for hyper-references starting with the keyword `\tel` for phone number. Furthermore, we took semantic HTML tags and linked data included in the website into account as well as we applied Google's phone library, libphonenumber[3]. In all cases, we chose the phone number that occurred with the highest frequency on a website. In case multiple phone numbers had the same frequency, we chose the one that occurred at the first place on a website.

8.3.3.2 *hasLatestOrganizationFoundedDate*

We applied regular expressions to the text of a website as well as to the title of a website to extract the value of this attribute for both tasks. We chose the smallest four digit-number in the interval $[1900, 2018]$.

In another method, we searched for the keyword `\founded` in the text of a website and in its HTML markup to extract the year behind this keyword. For instance, in some cases the latest date of incorporation of an organization was given by the Open Graph protocol[4] in the header metadata `og:description` of a website.

8.3.3.3 *hasURL*

In both tasks, the URLs for the organization websites were given. In the first task, the correct ones were given, but in the second task, the aim was to validate the correctness of the given URLs. Therefore, we extracted the value for `mdaas:isDomiciledIn` from the subpages of a given website with the respective text extraction module described in the next subsection (see Section 8.3.3.4). In case the extracted value was the same as the given correct one, we assigned the given value for `vcard:hasURL` as correct, otherwise as incorrect.

8.3.3.4 *isDomiciledIn*

The correct values of this attribute were given in task two. For the first task, we predicted the value for `mdaas:isDomiciledIn` by semantic markups used in the websites, by NER and disambiguation,

3 `https://github.com/googlei18n/libphonenumber`
4 `http://ogp.me`

by a simple mapping function of the Top Level Domain (TLD) as well as by the country code of the extracted phone number.

One method was to inspect the website's markup to find semantic HTML tags on each subpage. In this process, we took the FOAF[5] vocabulary, SCHEMA.ORG[6] vocabulary, and the OPEN GRAPH[7] protocol into account.

A language detection approach[8] [253] integrated in FOX, which is based on N-grams to detect languages, was another method applied on each subpage. In case the detected language was supported by FOX, we sent for each subpage of a website its text to FOX for extracting entities of the type τ = `LOCATION` and link these entities to the class `dbo:Place` contained in DBPEDIA. Then, we queried DBPEDIA for each entity to find the country of the extracted entity. We counted the frequency of the countries that occurred on a website and chose the one with the highest frequency. Additionally, we used a simple mapping function that maps the TLD of a given website URL to its county with the help of a lookup table (e. g., `de` for Germany, `us` for United States). In case the table had no entry for the TLD (e. g., `com`, `net`, `org`), we chose United States as default value.

8.3.4 *Scoring and Ranking*

We applied this package in both challenge tasks. The evaluation and benchmarking platform offered a leaderboard and provided participants with several measures, e. g., Precision, Recall and F-measure (see Section 2.3). Hence, we used the leaderboard to compete with other participants and to optimize our baseline. For this purpose, we applied each information extraction method to the test data and uploaded the partial results to the platform to get the Precision of the partial solution extracted by the specific method. The Precision of a specific method served as score and we ranked each method by its score. In case multiple methods extracted a value for an attribute, we chose the value received by the extraction method with the highest rank, thus the highest Precision.

8.3.5 *Attribute Validation*

For the participation in the second task, we reused the packages from the first task, *Attribute Extraction* and *Scoring and Ranking*, to extract the values from the websites and to use the values that come from the extraction method with the highest rank. We compared the extracted values with the given values in the statements of task's two dataset.

5 `http://xmlns.com/foaf/spec/`
6 `http://schema.org/`
7 `https://ogp.me/ns#`
8 `https://github.com/optimaize/language-detector`

In case a value for an attribute was equal, we assumed that the value was correct, otherwise we assumed that it was incorrect. Thus, we provided either 1 or 0 for the assessment of the correctness.

8.4 RESULTS

The task one dataset contained statements about 14,425 organizations (PermIDs) with 14,392 unique names and 13,953 unique website URLs. The task two dataset contained statements about 14,351 organizations with 14,309 unique names and 41,734 statements that needed an assessment.

We observed multiple duplicates in the datasets. For instance, one organization name occurred 17 times in the task one and 30 times in the task two dataset. The website URL of this organization occurred 79 times in the task one and 75 times the task two dataset.

We crawled and downloaded 228,687 subpages on 23,847 website URLs in total. Figure 8.2 provides an overview of the number of subpages of the website URLs. For instance, 35.83% (8545) of the websites had just one single page and 29.34% (6996) had 20 or more subpages, thus 70.66% of the websites had lesser than 20 subpages. In our experiments we choose 20 URLs in total for one website URL to limit the run time of our approach. An analysis of the impact of this limitation was out of the scope of this work.

In this downloaded data, we detected more than 40 different languages and observed 17 TLDs without a country entry in our lookup table (e. g., `com`, `net`, `org`).

Eleven systems participated in the first and four systems in the second task of the 2017 SW Challenge. The names and results of the best three systems are provided in Table 8.1 for task one and in Table 8.2 for task two. To the best of our knowledge, the two systems SOCRATES-KI and MATCHSOUP in task two were submitted by

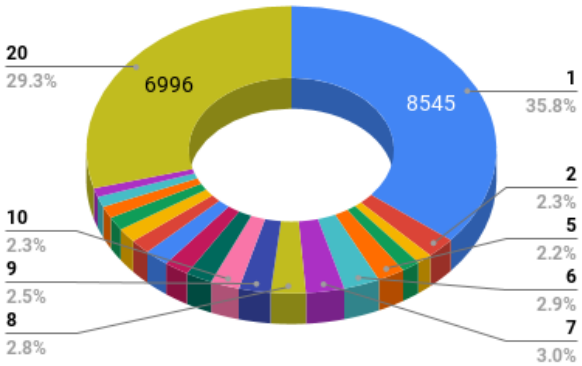


Figure 8.2: Overview of the number of subpages of a website.

	F-score [%]	*Pr* [%]	*Re* [%]
SOCRATES	55.40	×	×
LEOPARD	53.42	75.28	41.40
DISCO	53.32	×	×

Table 8.1: Results on task 1, attribute prediction.

	AUC-ROC [%]
SOCRATES-KI	68.01
MATCHSOUP	65.18
LEOPARD	53.09

Table 8.2: Results on task 2, attribute validation.

same research group and therefore count as one participation in the challenge.

Thus, LEOPARD achieved the second-best score in both challenge tasks with 53.42% F-measure in task one and 53.09% AUC-ROC in task two behind IBM's system SOCRATES with 55.40% F-measure in task one and 68.01% AUC-ROC in task two. In task one, SOCRATES obtained 1.98% F-measure points more than LEOPARD. Regarding LEOPARD, it is noteworthy that the achieved Precision with 75.28% is considerable higher than the Recall with 41.40%. In task two, SOCRATES obtained a 28.10% larger AUC-ROC than LEOPARD.

The evaluation results are available in GERBIL for task one[9] and task two[10].

8.5 ERROR ANALYSIS

LEOPARD applied Fox for NER and AGDISTIS [277] for EL to DBPEDIA. Data extracted from semi-structured sources, such as DBPEDIA, often contains inconsistencies as well as misrepresented and incomplete information [298]. The integration of multiple KGs and a type check over multiple KGs appears to be a solution to this issue.

For instance, at the time of writing this report, the DBPEDIA resource `dbr:England` is of the type `dbo:MusicalArtist` (i. e., a subtype of `dbo:Person`) instead of the correct type `dbo:PopulatedPlace` (i. e., a subtype of `dbo:Place`). One method in LEOPARD extracted places in the subpages of a website and linked those places to DBPEDIA. Then, with those links, it queried the KG to find the countries to the

9 http://swc2017.aksw.org/gerbil/experiment?id=201709200049
10 http://swc2017.aksw.org/gerbil/experiment?id=201709110026

extracted places. This method ignored resources of other types than `dbo:Place`. Thus, `dbr:England` was ignored.

Fox supports in its current version five languages (DE, EN, ES, FR, NL) and is a component in Leopard. Thus, the performance of Leopard was better on websites in one of those five languages. Hence, the amount of websites in languages that were supported influenced in the the overall performance. The results of the first task exposed a lack of Recall performance of Leopard. This low Recall arose from the gap of the language support, because the datasets consisted of websites in more than 40 different languages but Fox supports just five. The extension of Fox with more languages or the integration of tools with a broader language support such as DBpedia Spotlight is a solution to close this gap.

Our system performed better when many subpages of a website were available, since some methods inside our solution are based on an analysis of the frequency distributions on websites. Without subpages such a counting was not possible.

When Leopard found the international phone number of an organization on a website, the performance to find the country in which the organization incorporates by a look up of the country code of the phone number was very high. Thus, the performance on the extraction of the country in which the organization incorporates was lower on websites with phone number without a country code.

However, our system achieved a poor performance on websites implemented in Javascript only. Our crawler, which downloaded the subpages of a website, was not supporting such scenarios. Solutions to this issue are the integration of a headless browser, a web browser without a graphical user interface, that has the possibility to execute websites implemented in Javascript, or the integration of OXPath [83] into Leopard. Moreover, the support to find information in other sources than text (e. g., images) is not implemented in our solution so far but is a promising approach towards increasing its Recall.

9

OPEN KNOWLEDGE EXTRACTION CHALLENGE 2017

In this chapter[1], we describe the Open Knowledge Extraction (OKE) challenge.

9.1 RELATED WORK

To push the state of the art in knowledge extraction from natural language text, the OKE challenge aims to trigger attention from the knowledge extraction community and foster their broader integration with the Semantic Web (SW) community. Therefore, the OKE challenge has the ambition to provide a reference framework for research on knowledge extraction from text for the SW by defining a number of tasks (typically from information and knowledge extraction), taking into account specific SW requirements.

The first [206] and second [205] OKE challenge were both composed of two tasks, *Entity Recognition, Linking and Typing for Knowledge Base population* and *Class Induction and entity typing for Vocabulary and Knowledge Base enrichment*. In the first version, the challenge had four participants, ADEL [223], CETUS [240], FRED [40] and OAK [87]. In the second version the challenge had five participants, a new version of ADEL [225], MANNHEIM [71], WESTLAB-TASK1 [30], WESTLAB-TASK2 [98], and the baseline with CETUS from the former year.

We present the third edition of the OKE challenge in this chapter. The fourth OKE challenge took place at the 15th Extended Semantic Web Conference in 2018 and reused the first two tasks from the former challenge, *Focused* and *Broader Named Entity Identification and Linking* as well as defined two new tasks, *Relation Extraction* and *Knowledge Extraction*. In this challenge the participation in terms of number of competing systems remained quite limited with two, we believe that the challenge is a success in the hybridization of SW technologies with knowledge extraction methods. However, this challenge is not part of this thesis.

¶ Parts of this chapter have been published as conference article [264, 265].

1 Throughout this chapter, we use the prefix `owl`, `schema`, `rdf`, `rdfs`, `xsd`, `dbo`, `dbr`, `nif`, `mo`, `artist` and `itsrdf` for the Internationalized Resource Identifiers (IRIs) `http://www.w3.org/2002/07/owl#`, `http://schema.org/`, `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, `http://www.w3.org/2001/XMLSchema#`, `http://dbpedia.org/ontology/`, `http://dbpedia.org/resource/`, `http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#`, `http://purl.org/ontology/mo/`, `http://musicbrainz.org/artist` and `http://www.w3.org/2005/11/its/rdf#`.

9.2 OPEN KNOWLEDGE EXTRACTION CHALLENGE TASKS

The OKE challenge, at its third edition, comprises three tasks. Each task composes of Named Entity Recognition (NER) and Entity Linking (EL) to a Knowledge Graph (KG). The KG in the first two tasks is the WIKIPEDIA based DBPEDIA and in the third task the MUSICBRAINZ[2] based LINKED BRAINZ.

However, for measuring system performances in different perspectives based on the size and noise of the data in the first two tasks, the tasks are subdivided into two scenarios, Scenario A and Scenario B.

In contrast to the size of the data in Scenario A that is small due to the curated data generation process involved, the size of the data in Scenario B is large induced by the automated data generation process with the help of BENGAL[3,4] [196] to produce synthetic data. The KG utilized in the third task is provided by the challenge (see Section 9.3.1) and dubbed MBL.

Both, the given input and the expected output are expressed with the help of the NLP Interchange Format [108] (NIF) vocabulary and ontology in an Resource Description Framework [45, 136] (RDF) serialization, for instance `RDF/TURTLE`. A participating system is not expected to perform any preprocessing (e. g., pronoun resolution [110]) on the input data. In case a resource for an entity is missing in the KG, a participating system is expected to generate a Uniform Resource Identifier [17] (URI) with the namespace for this emerging entity as follows: `niw:<entity>`, where <entity> is the identifier to be set for the emerging entity.

For carrying out the evaluation, the OKE challenge is using the HOBBIT benchmarking platform [200] and the benchmark implementation of the HOBBIT project[5] which rely on the GERBIL evaluation framework [279].

9.2.1 *Task 1: Focused Named Entity Identification and Linking*

The first task aims at the identification of Entity Mentions (EMs) with the NER task and the linking of these EMs with the EL task to a given set of entity descriptions. It is a two-step process with the identification of EMs (Recognition) and the linking of those EMs to resources in DBPEDIA (D2KB). A competing system is expected to identify EMs in sentences of a given text document $d = \langle s_1, s_2, \ldots \rangle$ by its start and end index, further to generate a URI to link each identified entity to DBPEDIA if possible or generate a URI for an emerging entity.

2 http://musicbrainz.org
3 https://github.com/dice-group/BENGAL
4 http://project-hobbit.eu/wp-content/uploads/2017/04/D2.2.1.pdf
5 http://project-hobbit.eu

The task is limited to a subset of resources in DBpedia, i. e., resources of the DBpedia ontology types: `dbo:Person`, `dbo:Place` and `dbo:Organisation`.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/
    nif-core#> .

<http://example.com/example-task1#char=0,91>
 a nif:RFC5147String , nif:String , nif:Context ;
 nif:beginIndex "0"^^xsd:nonNegativeInteger ;
 nif:endIndex "124"^^xsd:nonNegativeInteger ;
 nif:isString "Leibniz was born in Leipzig in 1646 and attended the
     University of Leipzig from 1661-1666."@en .
```

Listing 9.1: Example request document in task 1.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/
    nif-core#> .

<http://example.com/example-task1#char=0,7>
 a nif:RFC5147String , nif:String ;
 nif:anchorOf "Leibniz"@en ;
 nif:beginIndex "0"^^xsd:nonNegativeInteger ;
 nif:endIndex "7"^^xsd:nonNegativeInteger ;
 nif:referenceContext <http://example.com/example-task1#char=0,91> ;
 itsrdf:taIdentRef dbr:Gottfried_Wilhelm_Leibniz .

<http://example.com/example-task1#char=20,27>
 a nif:RFC5147String , nif:String ;
 nif:anchorOf "Leipzig"@en ;
 nif:beginIndex "20"^^xsd:nonNegativeInteger ;
 nif:endIndex "27"^^xsd:nonNegativeInteger ;
 nif:referenceContext <http://example.com/example-task1#char=0,91> ;
 itsrdf:taIdentRef dbr:Leipzig .

<http://example.com/example-task1#char=53,74>
 a nif:RFC5147String , nif:String ;
 nif:anchorOf "University of Leipzig"@en ;
 nif:beginIndex "53"^^xsd:nonNegativeInteger ;
 nif:endIndex "74"^^xsd:nonNegativeInteger ;
 nif:referenceContext <http://example.com/example-task1#char=0,91> ;
 itsrdf:taIdentRef dbr:Leipzig_University .
```

Listing 9.2: Example of the expected response document in task 1.

Listing 9.1 is an example request document d of task 1 and Listing 9.2 is the expected response document for the given request document. Both documents are formalized with the NIF.

9.2.2 *Task 2: Broader Named Entity Identification and Linking*

This task extends the former task towards the DBpedia ontology types. Beside the three types of the first task, a competing system might have to identify other types of entities and to link these entities as

Table 9.1: Types, subtypes examples and instance examples for task 2.

Type	Subtypes	Instances
dbo:Activity	dbo:Game, dbo:Sport	dbr:Baseball, dbr:Chess
dbo:Agent	dbo:Organisation, dbo:Person	dbr:Leipzig_University
dbo:Award	dbo:Decoration, dbo:NobelPrize	dbr:Humanitas_Prize
dbo:Disease		dbr:Diabetes_mellitus
dbo:EthnicGroup		dbr:Javanese_people
dbo:Event	dbo:Competition	dbr:Battle_of_Leipzig
dbo:Language	dbo:ProgrammingLanguage	dbr:English_language
dbo:MeanOfTransportation	dbo:Aircraft, dbo:Train	dbr:Airbus_A300
dbo:PersonFunction	dbo:PoliticalFunction	dbr:PoliticalFunction
dbo:Place	dbo:Monument, dbr:WineRegion	dbr:Beaujolais, dbr:Leipzig
dbo:Species	dbo:Animal, dbo:Bacteria	dbr:Cat, dbr:Cucumibacter
dbo:Work	dbo:Artwork, dbo:Film	dbr:Actrius, dbr:Debian

well. In the first column in Table 9.1, a complete list of types that are considered in this task is provided. The middle column contains example subtypes of the corresponding class if any such class is available and the last column contains example instances in DBpedia for the related class respectively subtypes.

9.2.3 *Task 3: Focused Musical Named Entity Recognition and Entity Linking*

Task 3 composes of two subtasks: (i) Focused musical Named Entity (NE) identification and classification in Section 9.2.3.1, and (ii) linking to the Mbl KG that is based on MusicBrainz in Section 9.2.3.2. Thus the domain of this task is music and a competing system has to fulfill both tasks in order to participate. Listing 9.3 is an example input document and Listing 9.4 the expected annotated document for the given input, both formalized with the NIF.

9.2.3.1 *Task 3A: Focused Musical Named Entity Recognition*

This subtask consists of the identification (Recognition) and classification (Typing) of NEs. The task is limited to a subset of resources in Mbl, i. e., resources of the Mbl ontology types: `mo:Artist`, `mo:Album` and `mo:Song`. A competing system is expected to identify elements in a given text by its start and end index, further to assign one of the three types to each element.

9.2.3.2 *Task 3B: Musical Entity Linking*

In this subtask a participating system has to link the recognised entities of the former subtask to the corresponding resources in Mbl if existing or to generate a URI for the emerging entity.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/
    nif-core#> .
```

```
<http://example.com/example-task3#char=0,40>
 a nif:RFC5147String , nif:String , nif:Context ;
 nif:beginIndex "0"^^xsd:nonNegativeInteger ;
 nif:endIndex "40"^^xsd:nonNegativeInteger ;
 nif:isString "When Simon & Garfunkel split in 1970,..."@en .
```

Listing 9.3: Example request document in task 3

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix mo: <http://purl.org/ontology/mo/> .
@prefix artist: <http://musicbrainz.org/artist> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/
    nif-core#> .

<http://example.com/example-task3#char=5,22>
 a nif:RFC5147String , nif:String ;
 nif:anchorOf "Simon & Garfunkel"@en ;
 nif:beginIndex "5"^^xsd:nonNegativeInteger ;
 nif:endIndex "22"^^xsd:nonNegativeInteger ;
 nif:referenceContext <http://example.com/example-task3#char=0,40> ;
 itsrdf:taIdentRef artist:5d02f264-e225-41ff-83f7-d9b1f0b1874a ;
 itsrdf:taClassRef mo:MusicArtist .
```

Listing 9.4: Example of the expected response document in task 3.

9.3 EVALUATION

Overall, we follow two main evaluation approaches: subjective and objective. The subjective evaluation is based on paper reviews and the objective evaluation is based on computing relevance measures.

The KGs DBpedia and Mbl are used and the performance of a system is measured using Recall, Precision, F-measure and β. Note that we reuse the ability of the Gerbil project enabling the benchmarking of systems that link to another KG than DBpedia as long as there exist sameAs links (for instance, `owl:sameAs` or `schema:sameAs`) between the two KGs [239].

9.3.1 *Datasets*

The documents in the datasets might contain emerging entities, i. e., entities that are not part of the KG. These entities have to be marked and a URI has to be generated for them.

The datasets for the challenge are available at the challenge website[6]. Table 9.2 shows all the datasets available on the site assigned to the tasks and scenarios.

The music KG Mbl used in task 3 is provided by the challenge at the website in the file `\MusicBrainzRDF.tar.gz` as well.

6 `http://hobbitdata.informatik.uni-leipzig.de/oke2017-challenge/`

Table 9.2: Datasets of the challenge.

Task	Scenario	File
1	A	task1/A/training.tar.gz
		task1/A/evaluation.tar.gz
	B	task1/B/scenario-b-eval.zip
2	A	task2/A/training.tar.gz
		task2/A/evaluation.tar.gz
	B	task2/B/scenario-b-eval.zip
3	A	task3/A/training.tar.gz
		task3/A/evaluation.tar.gz

9.3.2 *Measures*

To assess the performance of the different algorithms, we computed on the evaluation datasets the following values for each document (i. e., for each sequence of sentences) $d \in D$: TP_d, TN_d, FP_d, FN_d and we computed t_d, the time (in seconds) the annotation system needed for the annotation of d. We micro average the performances over the documents with the measures formalized in Equations (2.6) to (2.8), where in this case, the number of datasets n corresponds to the number of documents $|D|$. The macro averages for the performance measures can be retrieved from the official HOBBIT SPARQL Protocol And RDF Query Language [37] (SPARQL) endpoint[7].

Further, we calculated β in Equation (9.1), the amount of F-measure points a system achieves per second for a given amount of documents. Let D be a set of documents for which β should be calculated. Let $F\text{-}score_d$ be the F-measure a benchmarked annotation system achieved for a given document $d \in D$ and let t_d be the time (in seconds) the annotation system needed for the annotation of d. β is defined as:

Definition 9.1 (β)

$$\beta = \frac{\sum_{d \in D} F\text{-}score_d}{\sum_{d \in D} t_d} \quad . \tag{9.1}$$

Last, we computed the number of documents $\varkappa$ that cause errors in the algorithms.

For matching the EM positions of the benchmarked system and the correct entity markings of the datasets we used the *weak annotation matching* defined in [279]. Thus, an entity is counted as having the correct position, if its position overlaps with the correct position of the entity inside the dataset.

7 http://db.project-hobbit.eu/sparql

For example, our dataset considered "*Gottfried Wilhelm Leibniz*". If a tool generated a URI for the emerging entity "*Wilhelm Leibniz*" and omitted "*Gottfried*", it was assigned as a match.

9.3.3 *Platform*

The benchmark suite for NER and EL implemented within HOBBIT[8] [200] reuses some of the concepts developed within the open-source project GERBIL. These concepts were migrated and adapted to the HOBBIT architecture. The Platform provides two different implementations of the benchmark described in the following subsections. It calculates values of Precision, Recall and F-measure, measures the time a system needs to answer a request and counts the number of documents $\varkappa$ that cause errors in the benchmarked system.

9.3.3.1 *Scenario A: Quality-focused benchmarking*

The first type of benchmarking provided by our suite focuses on the measurement of quality a system achieves on a given set of documents. We assume that each benchmark dataset consists of a set of documents. The documents are sent to the benchmarked system one at a time. The benchmarked system generates a response and sends it back before receiving the next document. That means that the benchmarked system can be configured to concentrate all its resources on a single request and does not need to scale to a large number of requests. In this benchmarking, Scenario A, we rely on manually created gold standards.

The goal in this scenario is to achieve a high F-measure in a quality-focused benchmarking.

9.3.3.2 *Scenario B: Performance-focused benchmarking*

The second approach to benchmarking implemented by our platform aims to put a high load on the benchmarked system and to evaluate its runtime and quality in terms of Precision, Recall and F-measure. This approach hence focuses on the ability of a system to annotate documents in parallel with an increasing amount of load.

The benchmark creates a large amount of synthetic documents from the given KB using BENGAL[9]. These documents are sent to the system in parallel without waiting for responses for previous requests but with predefined delays between the single documents. During a first phase of the benchmark, the generated work load equals 1 document per second. After the 80 documents of this first phase have been sent, the next phase is started using half of the delay of the previous time. This is done for 6 phases. In the seventh and last phase all

8 http://project-hobbit.eu/wp-content/uploads/2017/04/D2.2.1.pdf
9 https://github.com/dice-group/BENGAL

80 documents of the phase are sent without a delay, this leads to workloads of $\{1, 2, 4, 8, 16, 32, 80\}$ documents per second during the different phases.

The performance of a system is measured by β which is defined in equation 9.1. The scenarios goal is to achieve a high β value in a performance-focused benchmarking.

9.4 PARTICIPANTS

The challenge attracted four research groups. Two systems were not passing the subjective evaluation. The two remaining groups participated with there system in the challenge, ADEL and FOX.

9.4.1 *Adel*

ADEL [227] is an adaptive entity recognition and linking framework based on an hybrid approach that combines various extraction methods to improve the recognition level and an efficient KG indexing process to increase the efficiency of the linking step [224, 226]. It deals with fine-grained entity types, either generic or domain specific. It also can flexibly disambiguate entities from different KGs.

9.4.2 *FOX*

FOX [259] has been introduced in 2014 as an ensemble learning-based approach combining several diverse state of the art NER approaches and is based on the work in [198]. The FOX framework[10][260] outperforms the current state of the art entity recognizers. It relies on AGDISTIS [277] to perform EL. AGDISTIS is a pure EL approach (D2KB) based on string similarity measures, an expansion heuristic for labels to cope with co-referencing and the graph-based HITS algorithm. The authors published datasets[11] along with their source code and an API[12]. AGDISTIS can only be used for the D2KB task. FOX together with AGDISTIS can be use on the A2KB and the RT2KB task. FOX serves as the baseline system in this OKE challenge.

9.5 RESULTS

In this section we present the results the participating systems reach on the three OKE challenge tasks. Tables 9.3 and 9.4 comprise the results for task 1 and 2 on both scenarios A and B. Tables 9.5 and 9.6 comprise the results for task 3A and 3B. The tables show the overall

10 https://dice-research.org/FOX
11 https://github.com/dice-group/n3-collection
12 https://github.com/dice-group/AGDISTIS

measures for Precision, Recall and F-measure in the first three rows. The last two rows in each table show the averaged time in seconds a system needs to perform a document and the errors a system triggers. Further Tables 9.3 to 9.5 show the interim results for step (i) in the next three rows and for step (ii) in the following three rows. For task 3.2 there are no interim results since there are no interim steps in this subtask.

9.5.1 *Task 1*

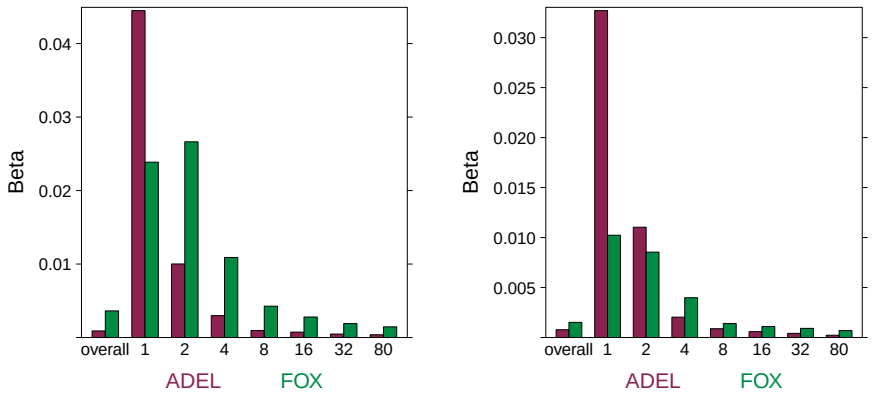
The measured values for scenario A in Table 9.3 show that ADEL outperforms FOX slightly with +1.09% F-measure in step (i) Recognition. In step (ii) D2KB, FOX outperforms ADEL clearly with +16.82% F-measure. Overall, FOX outperforms ADEL with *F-score* +18.29% in task 1 scenario A.

In scenario B, the results are similar to scenario A. In step (i) ADEL outperforms FOX slightly as well as FOX outperforms ADEL clearly in step (ii). Overall, FOX reaches the highest value in scenario B with 65.15% F-measure while ADEL reaches 20.12% F-measure. With 6 and 1 errors, the error rates of ADEL and FOX are low compared to the number of 560 documents they had to annotate in this scenario.

Table 9.3: Results on task 1.

Experiment	Measures	Scenario A		Scenario B	
		ADEL[13]	FOX[14]	ADEL[15]	FOX[16]
A2KB	Pr_{mic} [%]	33.24	53.61	18.28	59.12
	Re_{mic} [%]	30.18	46.72	22.36	72.51
	$F\text{-}score_{\text{mic}}$ [%]	31.64	**49.93**	20.12	**65.15**
Recognition	Pr_{mic} [%]	91.62	92.47	74.39	73.27
	Re_{mic} [%]	83.20	80.58	90.98	89.85
	$F\text{-}score_{\text{mic}}$ [%]	**87.21**	86.12	**81.85**	80.72
D2KB	Pr_{mic} [%]	40.15	61.96	28.03	93.87
	Re_{mic} [%]	27.82	41.47	19.26	66.99
	$F\text{-}score_{\text{mic}}$ [%]	32.87	**49.69**	22.83	**78.19**
	t [s/doc]	7.98	6.98	231.31	179.29
	$\varkappa$ [#]	0	0	6	1

Figure 9.1 depicts on the left side the detailed results for task 1 in scenario B. Surprisingly, ADEL reaches a clearly higher β value than FOX in the first phase for one document request per second. This is caused by the fast runtime of ADEL compensating its lower F-measure during that phase. In the following phases, the runtime of both systems increases—a clear sign that they are receiving requests to



(a) β on task 1.

(b) β on task 2.

Figure 9.1: β values on several numbers of requests and overall.

annotate document while they are still working on other documents. However, compared to Fox, the time that Adel needs per document increases much more. Since the F-measure of both systems are similar over all phases but the time needed per document of Fox does not increase as much as it does for Adel the β value of Fox remains higher than the value for Adel. The observation of the increasing of processing time can be also seen in the comparison of the overall values of scenario A and B. While in A, Adel needs 14% more time per document on average in scenario A this increases to 29% in scenario B. Together with the higher F-measure, the lower runtime of Fox leads to an overall β value which is four times higher than the value of Adel.

9.5.2 *Task 2*

The measured values for scenario A in Table 9.4 show that Adel outperforms Fox slightly with +4.83% F-measure in step (i) Recognition. In step (ii) D2KB, Fox outperforms Adel clearly with +14.02% F-measure. Overall, Fox outperforms Adel with +16.02% in task 2 scenario A. In difference to task 1, Adel is nearly twice as fast as Fox in scenario A.

In scenario B, the results are similar to A. In step (i) Adel outperforms Fox as well as Fox outperforms Adel clearly in step (ii). Overall, Fox reaches the highest value in scenario B with 42.22% F-measure while Adel reaches 18.15% F-measure.

Figure 9.1 depicts on the right side the detailed results for task 2 for scenario B. Similar to task 1, Adel reaches a clearly higher β value than Fox in the first two phases. This is again caused by the lower runtime of Adel that compensates its lower F-measure. In all other phases Fox reaches a higher β value because as in task 1 the runtime of Adel increases much more than the runtime of Fox when it receives many requests in a short amount of time. Overall, Fox nearly reaches

Table 9.4: Results on task 2.

Experiment	Measures	Scenario A		Scenario B	
		ADEL[17]	FOX[18]	ADEL[19]	FOX[20]
A2KB	Pr_{mic} [%]	31.40	56.15	17.44	44.90
	Re_{mic} [%]	28.14	38.53	18.93	39.83
	$F\text{-}score_{\text{mic}}$ [%]	29.68	**45.70**	18.15	**42.22**
Recognition	Pr_{mic} [%]	87.68	95.90	72.31	74.64
	Re_{mic} [%]	78.57	65.80	78.50	66.21
	$F\text{-}score_{\text{mic}}$ [%]	**82.88**	78.05	**75.27**	70.17
D2KB	Pr_{mic} [%]	39.93	63.42	28.57	82.38
	Re_{mic} [%]	25.76	35.28	17.47	36.92
	$F\text{-}score_{\text{mic}}$ [%]	31.32	**45.34**	21.68	**51.00**
	t [s/doc]	4.60	7.66	261.48	245.99
	$\varkappa$ [#]	0	1	57	0

a β value twice as high as the value achieved by ADEL. It is also worth noting that this is the only experiment, in which the error rate of one of the systems is increased. For 57 of the 560 documents, ADEL responded with an error code. Nearly all of these errors—9, 26 and 21—occurred during the last three phases. Since the documents are chosen randomly and ADEL reported nearly no errors in the phases before, it is possible that they are related to the high load that ADEL receives during these phases.

9.5.3 *Task 3*

Task 3 is composed of two subtask, 3A and 3B. In the following, we first summarize the results on the first subtask and then on the second.

9.5.3.1 *Task 3A*

The measured values for task 3A are depicted in Table 9.5. FOX reaches a higher F-measure than ADEL, 55.27% to 47.66% in step (i). In step (ii) ADEL reaches a higher F-measure, since FOX is not supporting this subtask due to the lack of the support of the music entity types.

Overall, ADEL reaches the highest value with 27.12% F-measure on this task.

9.5.3.2 *Task 3.2*

The measured values for task 3.2 are depicted in Table 9.6. Both systems, ADEL and FOX, reach low performance on this task. ADEL achieves 5.83% and FOX a slightly higher value with 6.66%.

Table 9.5: Results on task 3A.

Experiment	Measures	ADEL[21]	FOX[22]
RT2KB	Pr_{mic} [%]	26.99	0
	Re_{mic} [%]	27.24	0
	$F\text{-}score_{\text{mic}}$ [%]	**27.12**	0
Recognition	Pr_{mic} [%]	35.03	63.02
	Re_{mic} [%]	74.57	49.21
	$F\text{-}score_{\text{mic}}$ [%]	47.66	**55.27**
Typing	Pr_{mic} [%]	64.33	0
	Re_{mic} [%]	64.91	0
	$F\text{-}score_{\text{mic}}$ [%]	**64.62**	0
	t [s/doc]	37.19	7.82
	$\varkappa$ [#]	16	0

Table 9.6: Results on the D2KB experiment in task 3.2.

Measures	ADEL[23]	FOX[24]
Pr_{mic} [%]	6.82	10.10
Re_{mic} [%]	5.10	4.97
$F\text{-}score_{\text{mic}}$ [%]	5.83	**6.66**
t [s/doc]	36.96	9.15
$\varkappa$ [#]	16	0

It is noteworthy that FOX processed the documents faster with 9.15 s/doc in this subtask than ADEL with 39.96 s/doc. Additionally, FOX encountered no errors in comparison to ADEL for which 16 errors have been reported.

9.5.4 *Overall*

The winner of task 1 and 2 in both scenarios A and B is FOX. For task 3A the winner is ADEL, since FOX is not supporting all subtasks. For task 3B the winner is FOX again. Since the advantage ADEL has in task 3A is larger than the difference between FOX and ADEL in task 3B, ADEL is the overall winner of task 3.

The results on task 1 and 2 suggest, that the Recognition component in ADEL achieved a higher F-measure than the respective component in FOX, but its linking component showed a worse performance than than the respective component in FOX. Thus, it would be interesting

to investigate the performance of the composition of the Recognition component of ADEL together with the linking component in FOX in this tasks.

The results on task 3 in the music domain suggest that the Recognition component of FOX achieved a better F-measure than ADEL. While FOX is not supporting the music entity types in its current version. Thus, it would be interesting to investigate the performance of an extended version that supports this types compared to ADEL in this task.

Part IV

SYNOPSIS

10 SYNOPSIS

The thesis's primary objectives are to research, develop, and evaluate strategies that improve the quality of knowledge extraction from unstructured text sources. These objectives require novel solutions to extract high-quality structured data with semantic meaning from text, whereby minimal human effort, scalability, and high precision are desirable characteristics.

In the following, we outline our proposed solutions that address these requirements and fulfill these characteristics. Further, we outline our applications that utilize our proposed solutions, and additionally, we outline competitions in which we have participated with these applications.

10.1 SUMMARY

We addressed **RQ1** by presenting our approach in *Chapter 3*. An approach to increase the performance of state-of-the-art Named Entity Recognition (NER) tool based on ensemble learning and by presenting a thorough evaluation of this approach on the NER task, which is one of the four core tasks (Section 1.1) for Knowledge Graph (KG) creation and population. On all datasets, we showed that ensemble learning achieves higher F-measures than the best basic NER tools integrated in our system and higher F-measures compared to a simple Voting approach [290] with the majority vote rule [133] (MVote) strategy on the outcome of the integrated basic tool. Our results suggest that Multilayer Perceptron [233] (MLP) and AdaBoostM1 [79] with J48 as base classifier (ABM1) work best for the task at hand. We have integrated the results of this evaluation into the Fox framework[1]. The main advantages of our framework are that it is not limited to the integration of NER tools or ensemble learning algorithms and can be easily extended, for instance, for event extraction. Moreover, it provides additional features such as linked data and a web service to be used by the community.

We dealt with the issue described in **RQ2** by proposing Cetus in *Chapter 4*, a pattern-based type extraction that can be used as a baseline for other approaches. Cetus comprising offline, knowledge-driven type pattern extraction from text sources based on grammar rules, an analysis of input text to extract types, and the mapping of the extracted type evidence to a subset of the DOLCE+DnS Ultra

1 Our framework can be found at `https://dice-research.org/FOX`

Lite ontology classes. We implement[2] and compare two approaches for the third step using the YAGO ontology as well as the FOX entity recognition tool. Both versions, CETUS$_{YAGO}$ and CETUS$_{FOX}$, have been explained in detail. We showed how CETUS$_{YAGO}$ uses a label matching for determining a super type for the automatically generated classes while CETUS$_{FOX}$ is based on one of the various, existing entity typing tools.

We focused on closing the research gap described in **RQ3** by providing a comprehensive review of holistic Entity Linking (EL) and showing key aspects of the topic in *Chapter 5*. We reviewed and compared EL approaches that present some (potential for) holism, aiming to motivate, inspire, and give some directions for research in this field. We classified these approaches according to holism aspects that we have identified in our studies. Holistic approaches have the potential to boost EL by exploiting several data features and processing methods to make the highest possible number of semantically coherent links. Besides, we proposed potential pillars for future holistic EL approaches and a reference approach for holistic EL that exploits all these pillars.

We addressed **RQ4** by proposing OCELOT, a distant supervised Relation Extraction (RE) approach based on distributional semantics and a tree generalization in *Chapter 6*. In our approach, we use training data obtained from a reference KG to derive dependency parse trees that might express a relation. We then use a novel generalization algorithm to construct dependency tree patterns for a relation, and additionally, eliminate false candidate patterns from the generalized dependency tree patterns with distributional semantics. We evaluate the performance in experiments on a large corpus using ninety target relations. Our evaluation results suggest that our approach achieves a higher performance compared to two state-of-the-art systems. Moreover, our results also underpin the scalability of our approach. In a two-fold evaluation, quantitative and qualitative, we showed that our approach harvests generalized dependency tree patterns of high quality, and that it extracts relations from unstructured text with its generalized trees of higher precision than two state-of-the-art systems. Moreover, we provide the source code[3] of our approach together with the version numbers of all utilized tools and all settings as well as the datasets used in this work. We have now integrated the results of our work, the generalized trees, into the FOX framework, and additionally, integrated PATTY and BOA, thanks to the flexibility of the FOX framework that now provides our extraction results in a well-defined machine-processable semantics.

2 CETUS is open source and can be found at `https://github.com/dice-group/Cetus`.

3 OCELOT is open source and can be found at `https://github.com/dice-group/Ocelot`.

We presented the SCMS framework in *Chapter 7*, an approach for extracting knowledge from text sources of any Content Management System (CMS) contents and for integrating this extracted knowledge back into the same CMS. We presented the architecture of our approach and explained how each of its components work. In addition, we explained the vocabularies utilized by the components of our framework. We presented one use case for the SCMS system, i. e., how SCMS is used in the renewable energy sector. The SCMS stack abides by the criteria of accuracy and flexibility. The flexibility of our approach is ensured by the combination of Resource Description Framework [45, 136] (RDF) messages that can be easily extended and of standard Web communication protocols. The accuracy of SCMS was demonstrated in an evaluation on actor and country profiles, within which SCMS outperformed even commercial software.

In *Chapter 8*, we reported on the participation of LEOPARD[4] to the Semantic Web Challenge at the 16th International Semantic Web Conference. LEOPARD is our approach to predict and validate attributes for knowledge graph population. LEOPARD was designed as a baseline for the challenge and combines diverse text extraction methods with a simple precision ranking and utilizes sources from the multilingual Document Web as well as from the multilingual Data Web. The rather simple design of LEOPARD showed a good performance against ten other systems in the first task and against 3 systems in the second task. Despite being designed to be a baseline, surprisingly, LEOPARD reached the second-best score in both challenge tasks (53.42% F-measure and 53.09% AUC) behind IBM's system SOCRATES (55.40% F-measure and 68.01% AUC).

In *Chapter 9*, we presented the Open Knowledge Extraction (OKE) challenge that attracted four research groups coming from Knowledge Extraction and Semantic Web communities. We presented the challenge tasks, datasets, participated systems and results[5].

10.2 CONCLUSION AND FUTURE WORK

Our proposed approaches in this thesis, along with their implementations have become integral parts of a broad spectrum of applications [26, 90, 130, 185, 198, 240, 249, 275, 277, 278] and contributed in current trends and challenges. FOX is now a tool that is mentioned very frequently in the literature of question answering systems over linked data and documents, reported by Dimitrakis et al. (2020) [58]. The provided demo service has received over 1 million calls per month from organizations around the world [261]. Recently in the medical

4 LEOPARD is open source and can be found at `https://github.com/dice-group/Leopard`

5 More details can be found at the challenge website `https://project-hobbit.eu/open-challenges/oke-open-challenge/`

domain, our NER approach based on ensemble learning in [259] inspired Kaplar et al. (2022) [128]. The proposed baseline approach LEOPARD [263] for predicting knowledge graph attributes may be capable of further improving the predictive performance of vandalism detectors by exploiting the edit history of the knowledge graph and its structure, Heindorf et al. (2019) [106]. Huaman et al. (2021) [118] proposed an approach that computes a confidence score for every triple and instance in the KG, and later introduces a practical framework for knowledge graph curation [117], influenced by approaches such as LEOPARD. Some of the analyzed EL approaches already employ some holism. However, these approaches do not adequately cover all the potential pillars proposed in the thesis. For example, regarding the variety of data features that can be exploited in the EL process, few approaches exploit the fact that some social media posts (e. g., tweets) can be associated with geographic coordinates or labels of specific places, which can help to disambiguate some Entity Mentions (EMs). Moreover, holistic EL approaches could consider historical contexts determined by previous dependable annotations while exploiting technologies for collective disambiguation of entities based on coherence. Future approaches for EL could also tackle challenges such as the use of knowledge present in a variety of models such as KGs and embeddings. We proposed a closed RE approach based on distant supervision by using distributed semantics and a tree generalization process. It extracts sets of trees from a corpus where each set expresses a target relation from a knowledge base. These trees are then generalized using both a tree generalization approach and distributional semantics. One of the main advantages of this paradigm is that it is less sensitive to semantic drift, the problem described in Section 1.1. Whereas each of the computed generalized tree patterns matches to only one relation in our approach instead to several relations as in the case of the state-of-the-art systems, BOA and PATTY. With our approach, we improve upon the precision achieved by the state of the art while keeping the distant supervision and scalability it abides by. With our contribution, we push forward the quality of RE and thus the quality of applications in research areas such as KG creation and population. Our proposed SCMS approach can be extended by adding support for negative statements, i. e., statements that are not correct but can be found in different knowledge sources across the data landscape analyzed by our framework. In addition, the feedback generated by users will be integrated in the training of the framework to make it even more accurate over time.

Part V

APPENDIX

A

MOTIVATION

THE LINKED OPEN DATA (LOD)-CLOUD

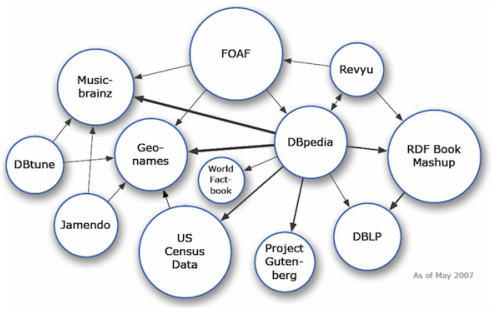


Figure A.1: The LOD-Cloud with 12 datasets, for instance, DBPEDIA and MUSICBRAINZ, as of May 01, 2007.

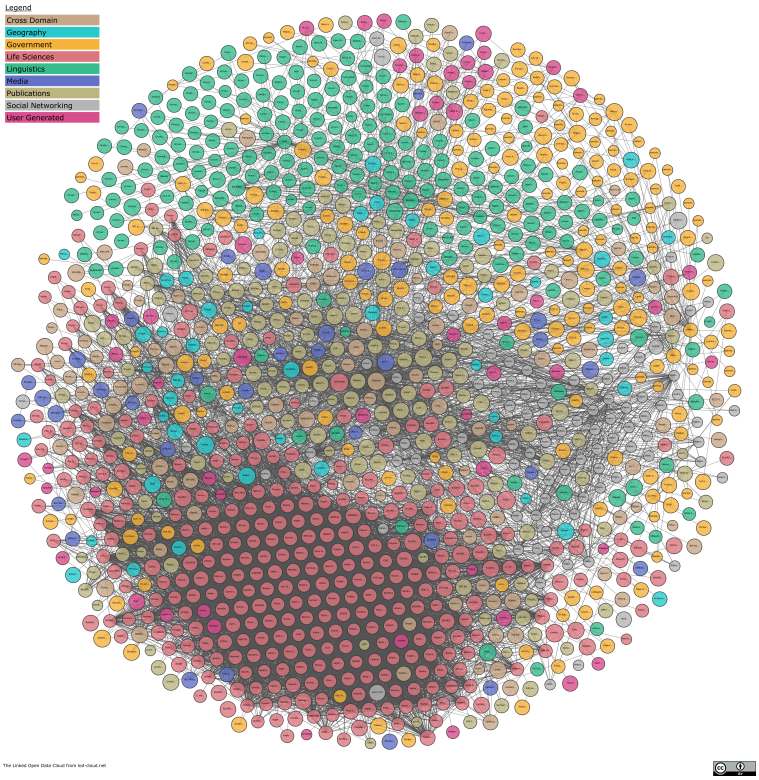


Figure A.2: The LOD-Cloud with 1,255 datasets as of May 20, 2020.

EXAMPLE SEMANTIC REPRESENTATION

```
@prefix its: <http://www.w3.org/2005/11/its/rdf#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/
    nif-core#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
[ a nif:Phrase ;
      nif:anchorOf "Michelle Obama" ;
      nif:beginIndex "3";
      its:taIdentRef <http://dbpedia.org/resource/Michelle_Obama> ] .
[ a rdf:Statement ;
      rdf:subject <http://dbpedia.org/resource/Michelle_Obama> ;
      rdf:predicate <http://dbpedia.org/ontology/spouse> ;
      rdf:object <http://dbpedia.org/resource/Barack_Obama> ] .
[ a nif:Phrase ;
      nif:anchorOf "Barack Obama" ;
      nif:beginIndex "30";
      its:taIdentRef <http://dbpedia.org/resource/Barack_Obama> ].
```

Listing A.1: Example semantic representation of the question *"Is Michelle Obama the wife of Barack Obama?"* in an RDF serialization, i. e., RDF/TURTLE.

EXAMPLE QUERY

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
ASK WHERE { dbr:Michelle_Obama dbo:spouse dbr:Barack_Obama}
```

Listing A.2: A SPARQL query in an RDF serialization (i. e., RDF/TURTLE) to ask DBpedia for the trueness of the statement.

B

HOLISTIC ENTITY LINKING

COMPARISON OF HOLISTIC APPROACHES FOR ENTITY LINKING

Table B.1: Comparison of holistic approaches for EL

Work	Input	NLP tasks	Method
Han et al. 2011 [100]	Wikipedia		Random Graph Walk
Wick et al. 2013 [288]	Wikilinks, Wikipedia	Entity Discovery	Markov Chain Monte Carlo
Moro et al. 2014 [180]	BabelNet	NER, WSD	Random Graph Walk with Restart
Huang et al. 2014 [119]	Wikipedia		Semi-supervised Graph Regularization
Guo & Barbosa 2014 [96]	Wikipedia		Random Graph Walk with Restart
Hua et al. 2015 [116]	Wikipedia		Ranking algorihtm based on user interest, entity popularity and entity recency
Luo et al. 2015 [161]	Wikipedia, Freebase, Sartori	NER	Semi-Conditional Random Fields extended for model entity distribution and mutual dependency over segmentation
Tran et al. 2015 [271]	Wikipedia	NER	Random Graph Walk
Kalloubi et al. 2016 [127]	DBpedia	NER	Graph Centrality Scoring
Ganea et al. 2016 [85]	Wikipedia		Markov Network (Factor Graph) + loopy belief propagation
Li et al. 2016 [157]	Linkless Wikipedia		Gibbs Sampling
Trani et al. 2016 [272, 273]	Wikipedia	Entity Saliency	Gradient Boosting Regression Tree
Fang et al. 2016 [69]	Wikipedia, Freebase		Logistic regression for two-layer model (Word Embedding, Knowledge Embedding)
Yamada et al. 2016 [292]	Wikipedia		Gradient Boosted Regression Trees (Word Embedding, Entity Embedding)
Chong et al. 2017 [35]	Wikipedia	NER (TweetNLP)	Objective function over a graph
Moreno et al. 2017 [179]	Wikipedia		Binary classifiers (Word Embedding, Entity Embedding)
Chen et al. 2018 [32]	Wikipedia, Freebase	Entity Saliency	Pairwise boosting regression tree (Word Embedding, Entity Embedding)
Le & Titov 2018 [146]	Wikipedia, Yago		Conditional random field, loopy belief propagation (Word Embedding, Entity Embedding)
Kolitsas et al. 2018 [138]	Wikipedia	NER	Shallow FFNN and LSTM (Word Embedding, Entity Embedding)
Zhu & Iglesias 2018 [307]	DBpedia		Semantic contextual similarity algorithm (Word Embedding, Category Embedding)
Mueller & Durrett 2018 [186]	Wikipedia		GRU (Word Embedding, Entity Embedding)
Martins et al. 2019 [166]	Wikipedia	NER	Stack-LSTM (Word Embedding, Entity Embedding)
Sevgili et al. 2019 [247]	Wikipedia, DBpedia		FFNN (Word Embedding, Graph Embedding)
Wang & Iwaihara 2019 [283]	Wikipedia	NER	TNN and CNN (Word embedding)
Wei et al. 2019 [287]	Wikipedia, Freebase		FFNN (Word Embedding)
Parravicini et al. 2019 [215]	DBpedia		Semantic similarity (Graph Embedding) and state-space search heuristic
Liu et al. 2019 [160]	Wikipedia		Forward-Backward algorithm (Entity Embedding)
Fang et al. 2019 [70]	Wikipedia		LSTM, Reinforcement Learning (Word Embedding, Entity Embedding)
Yang et al. 2019 [295]	Wikipedia		FFNN, Reinforcement Learning (Word Embedding, Entity Embedding)
Vaigh et al. 2019 [63]	Wikipedia, BaseKB		Binary logistic regression classifier (Word Embedding)
Phan et al. 2019 [222]	Wikipedia		Minimum Spanning Tree (Word Embedding, Entity Embedding)
Le & Titov 2019 [145]	Freebase		bi-LSTM, FFNN (Word Embedding, Entity Embedding)
Chen et al. 2020 [33]	Wikipedia		Conditional Random Field (Word Embedding, Entity Embedding)
Shi et al. 2020 [250]	Wikipedia, Yago		Vector Similarity (Entity Embedding, Knowledge Embedding)
Oliveira et al. 2020 [208]	DBpedia		bi-LSTM (Word Embedding, Knowledge Embedding)
Rama-Maneiro et al. 2020 [229]	Wikipedia, DBpedia		Graph Centrality Scoring, Topic similarity

GROUPS OF WORKS FOR ENTITY LINKING

Table B.2: Groups of works determined by our Decision Tree (DT). When available, a link to the source code of the respective approach is provided in the second column. Otherwise, the Github profile of each author is provided in the third column, as a link on the respective full name.

Work	Source code repository	Github of authors
Group 1		
Kalloubi et al. 2016 [127]		https://github.com/fahdkalloubi-ENSA
Chong et al. 2017 [35]		William Cohen
Group 2		
Huang et al. 2014 [119]		Chin-Yew Lin
Group 3		
Moro et al. 2014 [180]		Alessandro Raganato, Roberto Navigli
Group 4		
Han et al. 2011 [100]		
Guo & Barbosa 2014 [96]		Zhaochen Guo, Denilson Barbosa
Li et al. 2016 [157]		Shulong Tan, Huan Sun, Dan Roth
Ganea et al. 2016 [85]	https://github.com/dalab/pboh-entity-linking	
Vaigh et al. 2019 [63]	https://gitlab.inria.fr/celvaigh/ukbscael2019	
Wei et al. 2019 [287]		
Fang et al. 2019 [70]		
Parravicini et al. 2019 [215]		Alberto Parravicini, Davide B Bartolini, Rhicheek Patra, Marco D. Santambrogio
Phan et al. 2019 [222]		Ti Ray, Jialong Han
Yang et al. 2019 [295]	https://github.com/YoungXiyuan/DCA	
Liu et al. 2019 [160]		
Rama-Maneiro et al. 2020 [229]		Efren Rama-Maneiro, Juan C. Vidal
Group 5		
Oliveira et al. 2020 [208]	https://github.com/ItaloLopes/optic	
Group 6		
Tran et al. 2015 [271]		Tuan Tran, Nam K. Tran, Asmelash T. Hadgu, Robert Jäschke
Group 7		
Wick et al. 2013 [288]		Sameer Singh, Harshal Pandya, Andrew McCallum
Hua et al. 2015 [116]		Wen Hua
Trani et al. 2016 [272, 273]		
Fang et al. 2016 [69]		Dilin Wang
Yamada et al. 2016 [292]		
Moreno et al. 2017 [179]		Jose Moreno, Romaric Besançon, Romain Beaumont, Anne-Laure Ligozat, Xavier Tannier
Chen et al. 2018 [32]		
Le & Titov 2018 [146]	https://github.com/lephong/mulrel-nel	
Zhu & Iglesias 2018 [307]		Carlos A. Iglesias
Mueller & Durrett 2018 [186]	https://github.com/davidandym/wikilinks-ned	
Sevgili et al. 2019 [247]	https://github.com/uhh-lt/kb2vec	
Le & Titov 2019 [145]	https://github.com/lephong/dl4el	
Chen et al. 2020 [33]		Chen-Yew Lin, Chen Shuang, Junpeng Wang
Shi et al. 2020 [250]		
Group 8		
Luo et al. 2015 [161]		Chin-Yew Lin
Kolitsas et al. 2018 [138]	https://github.com/dalab/end2end_neural_el	
Martins et al. 2019 [166]		Pedro H. Martins, Zita Marinho, André F. T. Martins
Wang & Iwaihara 2019 [283]		

EVALUATION OF HOLISTIC APPROACHES

Table B.3: EL approaches evaluated with other metrics instead of F-measure. The symbol ≈ indicates that the values are approximate because the works provide them only in graphs.

Metric	Dataset	Work											
		Wick et al. 2013	Luo et al. 2015	Hua et al. 2015	Tran et al. 2015	Fang et al. 2016	Li et al. 2016	Yamada et al. 2016	Chong et al. 2017	Chen et al. 2018	Mueller & Durrett 2018	Wei et al. 2019	Shi et al. 2020
Accuracy	Custom Tweets (Hua)			≈0.62									
	Custom Tweets (Li)						≈ 0.79						
	TAC-KBP 2009						≈ 0.81						
	CoNLL (pershina)							0.93					
	CoNLL 2013							0.91					0.83
	TAC 2010							0.85					
	CoNLL									0.93			
	TAC 2010									0.88			
	KBP 2010					0.88							
	Wikilinks dev										0.74		
	Wikilinks test										0.75		
	Wikipedia + Wikilinks (Custom)	0.98											
P@1	CoNLL 2013		0.84										
P@5	Custom Tweets (Tran)				0.64								
P@15					0.49								
MAP@15					0.43								
CEAFmC F1	TAC-EDL 2016											0.67	
	TAC-EDL 2017											0.70	
NERLC F1	TAC-EDL 2016											0.65	
	TAC-EDL 2017											0.68	
Change Ratio	Custom Tweets (Chong)								12				

Table B.4: Evaluation of holistic EL approaches using $F\text{-}score_{\text{mic}}$ / $F\text{-}score_{\text{mac}}$. Cells with just one value refer to $F\text{-}score_{\text{mic}}$.

Work	Dataset																													Benchmark
	ACE 2004	AQUAINT	MSNBC	AIDA/CoNLL-Test B	N3-Reuters-128	N3-RSS-500	AIDA/CoNLL	WNED-CWEB	AIDA/CoNLL-Test A	DBpediaSpotlight	KORE50	WNED-WIKI	CoNLL 2003	IITB	Micropost2014-Test	Micropost216-Test	AIDA/CoNLL-Training	Custom Tweets (Meij)	Derczynski	Micropost2014-Train	Micropost2015-Test	OKE 2015	OKE 2015 Task 1 eval set	OKE 2016	OKE 2016 Task 1 eval set	OKE 2018 Task 1 train set	OKE 2018 Task 2 train set	OKE 2018 Task 4 train set	TAC-EDL 2015	Wikinews	
Han et al. 2011														0.73																	
Moro et al. 2014							0.82				0.71																				
Huang et al. 2014																		0.525													
Guo & Barbosa 2014	0.87 0.88	0.88 0.88	0.92 0.92																												
Ganea et al. 2016	0.87 0.90	0.86 0.86	0.89 0.89	0.87 0.86	0.76 0.83	0.71 0.78	0.86 0.86		0.86 0.85	0.79 0.80	0.61 0.55			0.62 0.61	0.74 0.84		0.86 0.87			0.73 0.81											GERBIL
Trani et al. 2016													0.72																0.72	
Fang et al. 2016	0.80	0.85	0.75																												
Moreno et al. 2017																													0.74		
Le & Titov 2018	0.90	0.88	0.93	0.93				0.77				0.78																			
Kolitsas et al. 2018			0.73 0.72	0.82 0.82	0.54 0.54	0.46 0.42			0.86 0.89		0.40 0.46								0.48 0.42			0.62 0.66		0.57 0.58							GERBIL
Zhu & Iglesias 2018						0.59										0.59															
Martins et al. 2019							0.81 0.81																								GERBIL
Sevgili et al. 2019					0.66 0.61					0.79 0.79																					GERBIL
Wang & Iwaihara 2019	0.76	0.76					0.81																								GERBIL
Parravicini et al 2019	0.84	0.86	0.92		0.82	0.72																									GERBIL
Liu et al. 2019	0.86	0.87		0.87				0.73	0.91																						
Fang et al. 2019	0.91	0.87	0.92					0.78				0.82																			
Yang et al. 2019	0.90	0.88	0.94					0.75				0.78																			
Vaigh et al. 2019				0.87	0.79	0.79			0.90																						GERBIL
Phan et al. 2019	0.88	0.87	0.91		0.85	0.82				0.84	0.78				0.81																GERBIL
Le & Titov 2019							0.37																								
Chen et al. 2020	0.88	0.89	0.93	0.93				0.77				0.80																			
Oliveira et al. 2020															0.29 0.57	0.50 0.95					0.33 0.45										GERBIL
Rama-Maneiro et al. 202	0.78	0.76	0.85	0.70	0.69	0.67				0.83				0.44									0.71		0.73	0.76	0.70	0.84			GERBIL

DATASETS EMPLOYED BY ENTITY LINKING APPROACHES

Table B.5: Datasets employed by EL approaches to evaluate their performance

Dataset Name	Link
ACE 2004	https://cogcomp.seas.upenn.edu/page/resource_view/4
AQUAINT	https://catalog.ldc.upenn.edu/LDC2002T31
MSNBC	https://cogcomp.seas.upenn.edu/page/resource_view/4
AIDA/CoNLL	https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/
N3-Reuters-128	https://github.com/AKSW/n3-collection
N3-RSS-500	https://github.com/AKSW/n3-collection
IITB	http://www.cse.iitb.ac.in/~soumen/doc/CSAW/Annot/
WNED-CWEB	Guo & Barbosa 2018 [97]
CoNLL 2013	http://www.cnts.ua.ac.be/conll2003/
DbpediaSpotlight	http://www.yovisto.com/labs/ner-benchmarks/
KORE50	http://www.yovisto.com/labs/ner-benchmarks/
WNED-Wiki	Guo & Barbosa 2018 [97]
Microposts2014	http://scc-research.lancaster.ac.uk/workshops/microposts2014/
Custom Tweets (Meij)	Meij et al. 2012 [171]
Derczynski	http://www.derczynski.com/sheffield/resources/ipm_nel.tar.gz
Microposts2015	http://scc-research.lancaster.ac.uk/workshops/microposts2015/
Microposts2016	http://microposts2016.seas.upenn.edu/challenge.html
OKE 2015	Open Knowledge Extraction at ESWC 2015
OKE 2016	Open Knowledge Extraction at ESWC 2016
OKE 2018	https://project-hobbit.eu/open-challenges/oke-open-challenge/
TAC-EDL 2015	https://tac.nist.gov/2015/KBP/data.html
TAC KBP 2010	https://tac.nist.gov/2010/KBP/
Wikipedia: test	Eshel et al. 2017 [65]
Wikinews	Trani et al. (2016) [272]
Custom Tweets (Hua)	Hua et al. 2015 [116]
Custom Tweets (Li)	Li et al. 2013 [158]
TAC KBP 2009	http://pmcnamee.net/kbp.html
CoNLL (Pershina)	Pershina et al. 2015 [221]
Wikilinks	http://www.iesl.cs.umass.edu/data/data-wiki-links
Custom Tweets (Tran)	Tran et al. 2015 [271]
Wikipedia + Wikilinks	Wick et al. 2013 [288]
TAC EDL 2016	http://nlp.cs.rpi.edu/kbp/2016/
TAC KBP 2017	http://nlp.cs.rpi.edu/kbp/2017/

ACRONYMS

Notation	Description	Page List
ABM1	AdaBoostM1 [79] with J48 as base classifier.	45, 47–53, 149
AUC-ROC	Area under the Receiver Operating Characteristic Curve.	126, 131
BG	Bagging [21] with J48 as base classifier.	45, 47–49, 51
CMS	Content Management System.	xx, 33, 109–115, 119, 151
DS	Distant Supervision.	32, 93–95, 97
DT	Decision Tree.	xix, xxii, 45, 85, 86, 158, 163
DTable	Decision Table [137].	45, 47–49, 51
ED	Entity Discovery.	75, 85, 87
EL	Entity Linking.	xix, xxi, xxii, 28–32, 42, 67–91, 123, 124, 131, 134, 139, 140, 150, 152, 157, 159–161
EM	Entity Mention.	xix, 56, 58, 67–85, 87, 89–91, 94, 134, 138, 152
ES	Entity Saliency.	72, 75, 76, 85
FAIR	FAIR data principles [289].	87, 126
FT	Functional Trees [84, 144].	45–49, 51
HTML	Hyper Text Markup Language.	97, 113–115, 127–129
IRI	Internationalized Resource Identifier.	55, 109, 123, 133
J48	A pruned C4.5 DT [228].	45, 47–53, 163

Notation	Description	Page List
KE	Keyphrase Extraction.	109, 110
KG	Knowledge Graph.	27–33, 35, 55, 60, 67, 68, 70, 72, 74–79, 81, 82, 84, 85, 89, 93, 97–99, 102, 103, 106, 124, 131, 134, 136, 137, 140, 149, 150, 152, 170
KGQA	Knowledge Graph Question Answering.	29, 30, 94
KPI	Key Performance Indicator.	36, 44, 45, 126
LMT	Logistic Model Trees [144, 266].	45, 47–49, 51
LOD	Linked Open Data.	xx, 28, 110, 155
LOG	Logistic Regression [29].	45, 47–49, 51, 164
LogD	Additive LOG [80].	45, 47–49, 51
MLP	Multilayer Perceptron [233].	45–49, 51–53, 149
MVote	Voting approach [290] with the majority vote rule [133].	45–51, 149
NB	Naïve Bayes [123].	45, 47–51
NE	Named Entity.	xx, 41, 42, 44, 55, 74, 94, 97, 98, 101, 105, 110–112, 116, 117, 119, 121, 136
NED	Named Entity Disambiguation.	67
NER	Named Entity Recognition.	28–32, 41–46, 48, 50, 51, 53, 61, 67, 72–74, 85, 109, 110, 116, 120, 123, 124, 128, 131, 134, 139, 140, 149, 152
NIF	Natural Language Processing (NLP) Interchange Format [108].	134–136
NLP	Natural Language Processing.	31, 68, 70–72, 74, 75, 84, 85, 97, 109, 110, 112, 115, 123, 134, 164
OKE	Open Knowledge Extraction.	xxi, 33, 62, 63, 133, 134, 140, 151

Notation	Description	Page List
OWL	Web Ontology Language [170, 217].	27, 28
POS	Part of Speech.	42, 95, 97, 100, 101
RDF	Resource Description Framework [45, 136].	xxii, xxiii, 27, 29, 33, 35, 59, 84, 89, 93, 109–112, 114, 117–119, 124, 134, 151, 156, 165
RDFa	RDF in Attributes [3].	110, 113, 117, 119, 120
RDFS	RDF Schema [23].	27
RE	Relation Extraction.	28, 29, 32, 93, 94, 106, 109, 110, 123, 150, 152
RF	Random Forest [22].	45–53
SCMS	Semantic Content Management System.	xx, 109–112, 114, 118–122
SMO	Sequential Minimal Optimization [104].	45, 47–49, 51
SPARQL	SPARQL Protocol And RDF Query Language [37].	xxiii, 29, 119, 120, 138, 156
SVM	Support Vector Machine [31].	45, 47–51
SW	Semantic Web.	110, 123, 124, 130, 133
TLD	Top Level Domain.	129, 130
TVote	Voting approach at entity type level.	45, 47–49, 51
URI	Uniform Resource Identifier [17].	60, 117, 118, 134, 136, 137, 139
URL	Uniform Resource Locator.	113, 114, 125–130
W3C	World Wide Web Consortium.	27
WSD	Word Sense Disambiguation.	72, 74, 75, 84
XML	Extensible Markup Language.	114, 115

RDF NAMESPACES

Notation	Description	Page List
ann	http://www.w3.org/2000/10/annotation-ns#.	109
artist	http://musicbrainz.org/artist.	133
content	http://purl.org/rss/1.0/modules/content/.	109, 113, 114
ctag	http://commontag.org/ns#.	109
dbo	http://dbpedia.org/ontology/.	xxi, 29, 82, 103–106, 129, 131–133, 135, 136
dbr	http://dbpedia.org/resource/.	29, 67, 71, 106, 131–133, 136
dc	http://purl.org/dc/elements/1.1/.	109
dul	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#.	55, 60–62
ex	http://example.com/.	55
foaf	http://xmlns.com/foaf/spec/.	123, 129
itsrdf	http://www.w3.org/2005/11/its/rdf#.	133
mdaas	http://ont.thomsonreuters.com/mdaas/.	123, 128
mo	http://purl.org/ontology/mo/.	133, 136
nif	http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#.	133
niw	http://aksw.org/notInWiki.	134
og	https://ogp.me/ns#.	123, 128, 129
owl	http://www.w3.org/2002/07/owl#.	123, 133, 137
permid	http://permid.org/.	123
permidOrg	http://permid.org/ontology/organization/.	123, 127

Notation	Description	Page List
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#.	55, 56, 109, 123, 133
rdfs	http://www.w3.org/2000/01/rdf-schema#.	55, 103, 109, 123, 133
schema	http://schema.org/.	129, 133, 137
scms	http://ns.aksw.org/scms/.	109, 117
scmsann	http://ns.aksw.org/scms/annotations/.	55, 61, 109, 117, 118
sioc	http://rdfs.org/sioc/ns#.	109
vcard	http://www.w3.org/2006/vcard/ns#.	123, 127, 128
xsd	http://www.w3.org/2001/XMLSchema#.	109, 123, 133
yago	http://yago-knowledge.org/resource/.	55, 60

Notation	Description	Page List
¶	The publication information for a chapter.	33, 41, 55, 67, 93, 109, 123, 133
$2^{\mathfrak{z}}$	The set of all contiguous subsequences of a sentence $\mathfrak{z}$.	35, 68, 94, 98
$\leq$	A binary tree relation as defined in Section 6.2.5.	95, 96
A	A finite set of vertex attributes.	95, 96
$\mathscr{A}$	A system in the domain of discourse.	36, 172
$\mathcal{B}$	A basic classifier.	42, 43
β	The amount of F-measure points a system achieves per second for a given amount of documents.	xx, 137, 138, 140–143
$\mathscr{C}$	A corpus is a finite set of sentences.	35, 68, 97, 98, 102
$\mathcal{C}$	A classifier.	42, 47–49, 51, 52
χ	A mapping function that decides if two entities are related to a given relation or not based on the extracted features.	94, 95
$\hat{c}$	An ordered sequence of child vertices.	95
D	A set of documents d.	138
d	A document, a sequence of i sentences $\mathfrak{z}_i$.	134, 135, 138, 169
E	A finite set of edges of a graph.	95, 96, 100
$\mathscr{E}$	An ensemble learning algorithm.	42–44
$\mathcal{E}$	A set of entity descriptions.	35, 36, 68, 94, 98
Er	The Error Rate as defined in Equation (2.5).	37, 38, 46–51, 169
Er_{mac}	The macro Error Rate as defined in Equation (2.15).	38
Er_{mic}	The micro Error Rate as defined in Equation (2.10).	37

Notation	Description	Page List
ε	An entity.	68, 94
F-score	The F-measure as defined in Equation (2.3).	xix, xxi, xxii, 31, 36–38, 46–53, 63–65, 104, 109, 121, 122, 131, 138, 141, 143, 144, 160, 170
$F\text{-}score_{\text{mac}}$	The macro F-measure as defined in Equation (2.13).	xxii, 38, 63, 64, 104, 121, 122, 160
$F\text{-}score_{\text{mic}}$	The micro F-measure as defined in Equation (2.8).	xix, xxii, 37, 53, 63–65, 121, 122, 141, 143, 144, 160
FN	False negatives are relevant entities incorrectly classified by a system as irrelevant.	36–38, 46, 138
FP	False positives are irrelevant entities incorrectly classified by a system as relevant.	29, 36–38, 46, 93, 94, 105, 138
$\mathcal{G}$	A Knowledge Graph (KG).	35, 98, 102
γ	A function that maps each resource and predicate to a word.	36, 98
Γ	Features extracted from a sentence $\mathfrak{z}$.	94
$\varkappa$	The number of documents that cause errors as defined by Gerbil.	126, 138, 139, 141, 143, 144
M	A matrix.	43
$\mathscr{M}$	A set of entity mentions.	68, 82
Mcc	The Matthews Correlation Coefficient as defined in Equation (2.4).	37, 38, 46–51, 170
Mcc_{mac}	The macro Matthews Correlation Coefficient as defined in Equation (2.14).	38
Mcc_{mic}	The micro Matthews Correlation Coefficient as defined in Equation (2.9).	37
μ	The arithmetic mean.	97, 103
m	An entity mention.	68, 82, 94, 95
$\mathfrak{n}$	The number of datasets in an evaluation.	37, 38, 46, 138
$\mathfrak{O}$	Set of all object labels.	98
Ω	A set of candidate sentences.	98

Notation	Description	Page List
Φ	A vertex labeling function family.	95
Pr	The Precision as defined in Equation (2.1).	36–38, 46–51, 62–64, 103, 104, 121, 131, 141, 143, 144, 171
Pr_{mac}	The macro Precision value as defined in Equation (2.11).	38, 62–64, 104, 121
Pr_{mic}	The micro Precision value as defined in Equation (2.6).	37, 62–64, 121, 141, 143, 144
ϕ	A vertex labeling function.	95, 96, 101
ψ	An edge labeling function.	95, 96, 100
Re	The Recall as defined in Equation (2.2).	36–38, 47–51, 63, 64, 104, 121, 131, 141, 143, 144, 171
Re_{mac}	The macro Recall value as defined in Equation (2.12).	38, 63, 64, 104, 121
$\mathcal{R}$	A set of relations.	35, 36, 82, 94, 97, 98
Re_{mic}	The micro Recall value as defined in Equation (2.7).	37, 63, 64, 121, 141, 143, 144
r	A target relation or predicate.	36, 82, 94, 95, 97–99
$\hat{r}$	The root vertex of a tree.	95, 96, 100, 101
$\mathcal{S}$	A set of statements.	35, 36, 98
Σ	An alphabet, the set of symbols of a language.	35, 171, 172
Σ^*	The Kleene closure, the set of all finite sequences over Σ.	35, 36, 171
Σ_a^*	Σ^* over the symbols set of the language a.	95
Σ_E^*	Σ^* over the symbols set of the language E.	95
$\mathcal{S}$	Set of all subject labels.	98
$\mathfrak{s}$	A sentence, a finite sequence of words.	35, 68, 94, 95, 98, 134, 169, 170
σ	The standard deviation.	97, 103
$\hat{s}$	The root vertex of a dependency parse tree, i. e., the semantic head of a sentence.	95, 96, 101
$\mathcal{T}$	A task in the domain of discourse.	36, 172

Notation	Description	Page List
TN	True negatives are irrelevant entities correctly classified by a system as irrelevant.	36–38, 46, 138
TP	True positives are relevant entities correctly classified by a system as relevant.	36–38, 46, 105, 106, 138
T	A tree.	95, 96, 98–101
$\mathcal{T}$	A set of trees.	98, 99
$\mathscr{T}$	A finite set of entity types $\{\tau_1, \ldots, \tau_i\}$ with a size of i.	45, 46, 121, 172
t	The time.	138, 141, 143, 144
τ	An entity type $\tau \in \mathscr{T}$, e. g., `PERSON`.	42, 43, 46, 52, 61, 67, 121, 122, 129, 172
$\mathcal{U}$	The domain of discourse.	36, 172
$\mathcal{U}_{\mathcal{T}^-}$	The set of irrelevant entities for a given task $\mathcal{T}$.	36
$\mathcal{U}_{\mathcal{T}^+}$	The set of relevant entities for a given task $\mathcal{T}$.	36
$\mathcal{U}_{\mathscr{A}}$	The set of retrieved entities by a system $\mathscr{A}$.	36
V	A finite set of vertices of a graph.	95, 100, 101
ω	A word, a finite sequence over an alphabet Σ.	35, 42, 68, 94
ζ	A function for extended labels of any resource.	98

BIBLIOGRAPHY

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. "Analyzing user modeling on twitter for personalized news recommendations." In: *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg, 2011, pp. 1–12.

[2] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. "Semantic enrichment of twitter posts for user profile construction on the social web." In: *The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer, 2011, pp. 375–389.

[3] Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. *RDFa in XHTML: Syntax and Processing*. Oct. 2008.

[4] Benjamin Adrian, Jörn Hees, Ivan Herman, Michael Sintek, and Andreas Dengel. "Epiphany: Adaptable RDFa Generation Linking the Web of Documents to the Web of Data." In: *EKAW*. 2010, pp. 178–192.

[5] Eugene Agichtein and Luis Gravano. "Snowball: Extracting Relations from Large Plain-Text Collections." In: *In ACM DL*. 2000, pp. 85–94.

[6] Ifeyinwa Angela Ajah and Henry Friday Nweke. "Big Data and Business Analytics: Trends, Platforms, Success Factors and Applications." In: *Big Data and Cognitive Computing* 3.2 (2019).

[7] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers." In: *J. Mach. Learn. Res.* 1 (Sept. 2001), pp. 113–141.

[8] R. Amsler. "Research Towards the Development of a Lexical Knowledge Base for Natural Language Processing." In: *SIGIR Forum* 23 (1989), pp. 1–2.

[9] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "DBpedia: A Nucleus for a Web of Open Data." In: *Proceedings of the 6th International Semantic Web Conference (ISWC)*. Vol. 4825. Lecture Notes in Computer Science. Springer, 2007, pp. 722–735.

[10] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. "LODStats – An Extensible Framework for High-Performance Dataset Analytics." In: *Knowledge Engineering and Knowledge Management*. Ed. by Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 353–362.

[11] Sören Auer, Sebastian Dietzold, and Thomas Riechert. "OntoWiki - A Tool for Social, Semantic Collaboration." In: *ISWC 2006*. Vol. 4273. LNCS. Springer, 2006, pp. 736–749.

[12] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. "Relation extraction from the web using distant supervision." In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2014, pp. 26–41.

[13] Nguyen Bach and Sameer Badaskar. "A review of relation extraction." In: *Literature review for Language and Statistics II* 2 (2007), pp. 1–15.

[14] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. *Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering*. 2023. arXiv: 2306.04136 [cs.CL].

[15] Jason Baldridge. *The opennlp project*. https://opennlp.apache.org/. Accessed: 2017-04-06. 2005.

[16] S. D. Bay and S. Hettich. *The UCI KDD Archive [http://kdd.ics.uci.edu]*. 1999.

[17] Tim Berners-Lee, Roy Thomas Fielding, and Larry Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. Internet RFC 2396. Aug. 1998.

[18] Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web." In: *Scientific American* 284.5 (May 2001), pp. 34–43.

[19] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge." In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. Vancouver, Canada: ACM, 2008, pp. 1247–1250.

[20] Kalina Bontcheva and Dominic Paul Rout. "Making sense of social media streams through semantics: A survey." In: *Semantic Web* 5.5 (2014), pp. 373–403.

[21] Leo Breiman. "Bagging predictors." In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140.

[22] Leo Breiman. "Random Forests." In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32.

[23] Dan Brickley and Ramanathan Guha. "RDF vocabulary description language 1.0: RDF schema." In: *W3C Recommendation* (Jan. 2004).

[24] Sergey Brin. "Extracting Patterns and Relations from the World Wide Web." In: *WebDB*. London, UK: Springer-Verlag, 1999, pp. 172–183.

[25] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. "Class-Based *n*-gram Models of Natural Language." In: *Computational Linguistics* 18.4 (1992), pp. 467–480.

[26] Lorenz Bühmann, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "ASSESS — Automatic Self-Assessment Using Linked Data." In: *International Semantic Web Conference (ISWC)*. 2015.

[27] Razvan Bunescu and Raymond Mooney. "Learning to Extract Relations from the Web using Minimal Supervision." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 576–583.

[28] Eric Burgener and John Rydning. *High Data Growth and Modern Applications Drive New Storage Requirements in Digital Transformed Enterprises*. `https://www.delltechnologies.com/asset/en-us/products/storage/industry-market/h19267-wp-idc-storage-reqs-digital-enterprise.pdf`. IDC white paper (Doc. #US49359722) sponsored by Dell Technologies and NVIDIA (Accessed: 2024-03-18). July 2022.

[29] S. le Cessie and J.C. van Houwelingen. "Ridge Estimators in Logistic Regression." In: *Applied Statistics* 41.1 (1992), pp. 191–201.

[30] Mohamed Chabchoub, Michel Gagnon, and Amal Zouaq. "Collective Disambiguation and Semantic Annotation for Entity Linking and Typing." In: *Semantic Web Challenges*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 33–47.

[31] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM - A Library for Support Vector Machines*. The Weka classifier works with version 2.82 of LIBSVM. 2001.

[32] Hui Chen, Baogang Wei, Yonghuai Liu, Yiming Li, Jifang Yu, and Wenhao Zhu. "Bilinear joint learning of word and entity embeddings for Entity Linking." In: *Neurocomputing* 294 (2018), pp. 12–18.

[33] Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. "Improving Entity Linking by Modeling Latent Entity Type Information." In: *arXiv preprint arXiv:2001.01447* (2020).

[34] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation." In: *BMC genomics* 21 (2020), pp. 1–13.

[35] Wen-Haw Chong, Ee-Peng Lim, and William Cohen. "Collective Entity Linking in Tweets Over Space and Time." In: *European Conference on Information Retrieval*. Springer, 2017, pp. 82–94.

[36] Chinmay Choudhay and Colm O'Riordan. "A graph-based collective linking approach with Group Co-existence Strength." In: *Proceedings for the 26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*. CEUR-WS, 2018, pp. 267–278.

[37] Kendall Grant Clark, Lee Feigenbaum, and Elias Torres. *SPARQL Protocol for RDF*. W3C Recommendation. W3C, Jan. 2008.

[38] Sam Coates-Stephens. "The Analysis and Acquisition of Proper Names for the Understanding of Free Text." In: *Computers and the Humanities* 26 (5 1992). 10.1007/BF00136985, pp. 441–456.

[39] Diego Collarana, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, and Sören Auer. "Semantic data integration for knowledge graph construction at query time." In: *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. IEEE. 2017, pp. 109–116.

[40] Sergio Consoli and DiegoReforgiato Recupero. "Using FRED for Named Entity Resolution, Linking and Typing for Knowledge Base Population." English. In: *Semantic Web Evaluation Challenges*. Ed. by Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann. Vol. 548. Communications in Computer and Information Science. Springer International Publishing, 2015, pp. 40–50.

[41] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. "A framework for benchmarking entity-annotation systems." In: *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2013, pp. 249–260.

[42] P. Cui, X. Wang, J. Pei, and W. Zhu. "A Survey on Network Embedding." In: *IEEE Transactions on Knowledge & Data Engineering* PP.01 (2018), pp. 1–1.

[43] James R. Curran and Stephen Clark. "Language independent NER using a maximum entropy tagger." In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*. Edmonton, Canada, 2003, pp. 164–167.

[44] James R. Curran, Tara Murphy, and Bernhard Scholz. "Minimising semantic drift with mutual exclusion bootstrapping." In: *In Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*. 2007, pp. 172–180.

[45] Richard Cyganiak, David Wood, Markus Lanthaler, Graham Klyne, Jeremy J Carroll, and Brian McBride. "RDF 1.1 concepts and abstract syntax." In: *W3C recommendation* 25.02 (2014), pp. 1–22.

[46] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. "Improving Efficiency and Accuracy in Multilingual Entity Extraction." In: *Proceedings of the 9th International Conference on Semantic Systems*. Association for Computing Machinery, 2013, pp. 121–124.

[47] Luciano Del Corro and Rainer Gemulla. "ClausIE: Clause-based Open Information Extraction." In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, 2013, pp. 355–366.

[48] Caglar Demir, Julian Lienen, and Axel-Cyrille Ngonga Ngomo. "Kronecker decomposition for knowledge graph embeddings." In: *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*. 2022, pp. 1–10.

[49] Caglar Demir and Axel-Cyrille Ngonga Ngomo. "Convolutional complex knowledge graph embeddings." In: *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*. Springer. 2021, pp. 409–424.

[50] Janez Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets." In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 1–30.

[51] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. "Analysis of named entity recognition and linking for tweets." In: *Information Processing & Management* 51.2 (2015), pp. 32–49.

[52] Kartik Detroja, C.K. Bhensdadia, and Brijesh S. Bhatt. "A survey on Relation Extraction." In: *Intelligent Systems with Applications* 19 (2023), p. 200244.

[53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "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)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[54] Dennis Diefenbach, José Giménez-García, Andreas Both, Kamal Singh, and Pierre Maret. "QAnswer KG: Designing a Portable Question Answering System over RDF Data." In: *The Semantic Web*. Ed. by Andreas Harth, Sabrina Kirrane, Axel-Cyrille

Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez. Cham: Springer International Publishing, 2020, pp. 429–445.

[55] Reinhard Diestel, Alexander Schrijver, and Paul Seymour. "Graph theory." In: *Oberwolfach Reports* 7.1 (2010), pp. 521–580.

[56] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. "Solving the multiple instance problem with axis-parallel rectangles." In: *Artificial intelligence* 89.1-2 (1997), pp. 31–71.

[57] Thomas G. Dietterich. "Ensemble Methods in Machine Learning." In: *Proceedings of the First International Workshop on Multiple Classifier Systems*. MCS '00. London, UK: Springer-Verlag, 2000, pp. 1–15.

[58] Eleftherios Dimitrakis, Konstantinos Sgontzos, and Yannis Tzitzikas. "A survey on question answering systems over linked data and documents." In: *Journal of intelligent information systems* 55 (2020), pp. 233–259.

[59] Francesco Draicchio, Aldo Gangemi, Valentina Presutti, and Andrea Giovanni Nuzzolese. "FRED: From Natural Language Text to RDF and OWL in One Click." In: *The Semantic Web: ESWC 2013 Satellite Events*. Ed. by Philipp Cimiano, Miriam Fernández, Vanessa Lopez, Stefan Schlobach, and Johanna Völker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 263–267.

[60] Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Hoffner, and Jens Lehmann. "AskNow: A Framework for Natural Language Query Formalization in SPARQL." In: *Proc. of the Extended Semantic Web Conference 2016*. 2016.

[61] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. "Transition-based dependency parsing with stack long short-term memory." 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)*. Association for Computational Linguistics, 2015, pp. 334–343.

[62] Lisa Ehrlinger and Wolfram Wöß. "Towards a Definition of Knowledge Graphs." In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48 (2016).

[63] Cheikh Brahim El Vaigh, François Goasdoué, Guillaume Gravier, and Pascale Sébillot. "Using Knowledge Base Semantics in Context-Aware Entity Linking." In: *Proceedings of the ACM Symposium on Document Engineering 2019*. 2019, pp. 1–10.

[64] Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. "LODStats: The Data Web Census Dataset." In: *The Semantic Web – ISWC 2016*. Ed. by Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil. Cham: Springer International Publishing, 2016, pp. 38–46.

[65] Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. "Named Entity Disambiguation for Noisy Text." In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, 2017, pp. 58–68.

[66] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. "Unsupervised named-entity extraction from the web: an experimental study." In: *Artif. Intell.* 165 (1 June 2005), pp. 91–134.

[67] MS Fabian, K Gjergji, and W Gerhard. "Yago: A core of semantic knowledge unifying wordnet and wikipedia." In: *Proceedings of the 16th International Conference on World Wide Web*. Association for Computing Machinery, 2007, pp. 697–706.

[68] Anthony Fader, Stephen Soderland, and Oren Etzioni. "Identifying relations for Open Information Extraction." In: *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. 2011, pp. 1535–1545.

[69] Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. "Entity disambiguation by knowledge and text jointly embedding." In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016, pp. 260–269.

[70] Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. "Joint Entity Linking with Deep Reinforcement Learning." In: *The World Wide Web Conference*. Association for Computing Machinery, 2019, pp. 438–447.

[71] Stefano Faralli and Simone Paolo Ponzetto. "DWS at the 2016 Open Knowledge Extraction Challenge: A Hearst-Like Pattern-Based Approach to Hypernym Extraction and Class Induction." In: *Semantic Web Challenges*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 48–60.

[72] Tom Fawcett. "An Introduction to ROC Analysis." In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874.

[73] Javier D. Fernández, Wouter Beek, Miguel A. Martínez-Prieto, and Mario Arias. "LOD-a-lot." In: *The Semantic Web – ISWC 2017*. Ed. by Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange, and Jeff Heflin. Cham: Springer International Publishing, 2017, pp. 75–83.

[74] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616)*. Ed. by Internet Engineering Task Force (IETF). Request For Comments. available at `http://www.ietf.org/rfc/rfc2616.txt`, accessed 7 July 2006. 1999.

[75] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 363–370.

[76] George Forman and Martin Scholz. "Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement." In: *SIGKDD Explor. Newsl.* 12.1 (Nov. 2010), pp. 49–57.

[77] G David Forney. "The viterbi algorithm." In: *Proceedings of the IEEE* 61.3 (1973), pp. 268–278.

[78] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. "Domain-Specific Keyphrase Extraction." In: *IJCAI*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 668–673.

[79] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm." In: *International Conference on Machine Learning*. 1996, pp. 148–156.

[80] J. Friedman, T. Hastie, and R. Tibshirani. *Additive Logistic Regression: a Statistical View of Boosting*. Tech. rep. Stanford University, 1998.

[81] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine." In: *Annals of statistics* 29.5 (2001), pp. 1189–1232.

[82] Yun Fu and Yunqian Ma. *Graph embedding for pattern analysis*. Springer Science & Business Media, 2012.

[83] Tim Furche, Georg Gottlob, Giovanni Grasso, Christian Schallhart, and Andrew Sellers. "OXPath: A Language for Scalable Data Extraction, Automation, and Crawling on the Deep Web." In: *The VLDB Journal* 22.1 (Feb. 2013), pp. 47–72.

[84] Joao Gama. "Functional Trees." In: 55.3 (2004), pp. 219–250.

[85] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. "Probabilistic bag-of-hyperlinks model for entity linking." In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conference Steering Committee, 2016, pp. 927–938.

[86] Octavian-Eugen Ganea and Thomas Hofmann. "Deep Joint Entity Disambiguation with Local Neural Attention." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 2619–2629.

[87] Jie Gao and Suvodeep Mazumdar. "Exploiting Linked Open Data to Uncover Entity Types." English. In: *Semantic Web Evaluation Challenges*. Ed. by Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann. Vol. 548. Communications in Computer and Information Science. Springer International Publishing, 2015, pp. 51–62.

[88] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and **René Speck**. "DeFacto - Temporal and Multilingual Deep Fact Validation." In: *Web Semantics: Science, Services and Agents on the World Wide Web* (2015).

[89] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. "Bootstrapping the Linked Data Web." In: *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*. 2011.

[90] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. "From RDF to Natural Language and Back." In: *Towards the Multilingual Semantic Web*. Springer, 2014.

[91] Michael Glass and Alfio Gliozzo. "A Dataset for Web-Scale Knowledge Base Population." In: *The Semantic Web*. Ed. by Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam. Cham: Springer International Publishing, 2018, pp. 256–271.

[92] Michael Glass, Alfio Gliozzo, Oktie Hassanzadeh, Nandana Mihindukulasooriya, and Gaetano Rossiello. "Inducing Implicit Relations from Text Using Distantly Supervised Deep Nets." In: *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*. Ed. by Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl. Vol. 11136. Lecture Notes in Computer Science. Springer, 2018, pp. 38–55.

[93] Palash Goyal and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey." In: *Knowledge-Based Systems* 151 (2018), pp. 78–94.

[94] R. Grishman and R. Yangarber. "Nyu: Description of the Proteus/Pet system as used for MUC-7 ST." In: *MUC-7*. Morgan Kaufmann, 1998.

[95] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks." In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. Association for Computing Machinery, 2016, pp. 855–864.

[96] Zhaochen Guo and Denilson Barbosa. "Entity linking with a unified semantic representation." In: *Proceedings of the 23rd International Conference on World Wide Web*. Association for Computing Machinery, 2014, pp. 1305–1310.

[97] Zhaochen Guo and Denilson Barbosa. "Robust named entity disambiguation with random walks." In: *Semantic Web* 9.4 (2018), pp. 459–479.

[98] Lara Haidar-Ahmad, Ludovic Font, Amal Zouaq, and Michel Gagnon. "Entity Typing and Linking Using SPARQL Patterns and DBpedia." In: *Semantic Web Challenges*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 61–75.

[99] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA Data Mining Software: An Update." In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), pp. 10–18.

[100] Xianpei Han, Le Sun, and Jun Zhao. "Collective entity linking in web text: a graph-based method." In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. Association for Computing Machinery, 2011, pp. 765–774.

[101] Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. "More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction." In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Ed. by Kam-Fai Wong, Kevin Knight, and Hua Wu. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 745–758.

[102] Sanda Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. "Shallow semantics for relation extraction." In: *IJCAI*. Edinburgh, Scotland, 2005, pp. 1061–1066.

[103] Mofeed M. Hassan, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "Using Caching for Local Link Discovery on Large Data Sets." English. In: *Engineering the Web in the Big Data Era*. Vol. 9114. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 344–354.

[104] Trevor Hastie and Robert Tibshirani. "Classification by Pairwise Coupling." In: *Advances in Neural Information Processing Systems*. Ed. by Michael I. Jordan, Michael J. Kearns, and Sara A. Solla. Vol. 10. MIT Press, 1998.

[105] Marti A. Hearst. "Automatic Acquisition of Hyponyms from Large Text Corpora." In: *COLING*. COLING '92. Nantes, France, 1992, pp. 539–545.

[106] Stefan Heindorf. "Vandalism Detection in Crowdsourced Knowledge Bases." PhD in Computer Science. PhD thesis. Paderborn University, 2019.

[107] Nicolas Heist, Sven Hertling, Daniel Ringler, and Heiko Paulheim. "Knowledge Graphs on the Web - An Overview." In: *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*. Ed. by Ilaria Tiddi, Freddy Lécué, and Pascal Hitzler. Vol. 47. Studies on the Semantic Web. IOS Press, 2020, pp. 3–22.

[108] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. "Integrating NLP using Linked Data." In: *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*. 2013.

[109] Pascal Hitzler, Krzysztof Janowicz, and Freddy Lecue. "On the Role of Knowledge Graphs in Explainable AI." In: *Semant. Web* 11.1 (Jan. 2020), pp. 41–51.

[110] Jerry Hobbs. "Pronoun resolution." In: *Lingua* 44 (1978), pp. 339–352.

[111] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. "Robust disambiguation of named entities in text." In: *EMNLP*. Association for Computational Linguistics. 2011, pp. 782–792.

[112] Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. "Survey on Challenges of Question Answering in the Semantic Web." In: *Semantic Web Journal* 8.6 (2017).

[113] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and

Antoine Zimmermann. *Knowledge Graphs*. 2020. arXiv: 2003.02320 [cs.AI].

[114] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. "Knowledge Graphs." In: *ACM Comput. Surv.* 54.4 (July 2021).

[115] Ian Horrocks, Peter F. Patel-Schneider, Sean Bechhofer, and Dmitry Tsarkov. "OWL rules: A proposal and prototype implementation." In: *Journal of Web Semantics* 3.1 (2005). Rules Systems, pp. 23–40.

[116] Wen Hua, Kai Zheng, and Xiaofang Zhou. "Microblog Entity Linking with Social Temporal Context." In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, 2015, pp. 1761–1775.

[117] Elwin Huaman and Dieter Fensel. "Knowledge graph curation: A practical framework." In: *The 10th International Joint Conference on Knowledge Graphs*. 2021, pp. 166–171.

[118] Elwin Huaman, Amar Tauqeer, and Anna Fensel. "Towards Knowledge Graphs Validation Through Weighted Knowledge Sources." In: *Knowledge Graphs and Semantic Web*. Ed. by Boris Villazón-Terrazas, Fernando Ortiz-Rodríguez, Sanju Tiwari, Ayush Goyal, and MA Jabbar. Cham: Springer International Publishing, 2021, pp. 47–60.

[119] Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. "Collective Tweet Wikification based on Semi-supervised Graph Regularization." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 380–390.

[120] David Huynh, Stefano Mazzocchi, and David R. Karger. "Piggy Bank: Experience the Semantic Web Inside Your Web Browser." In: *ISWC*. Vol. 3729. Lecture Notes in Computer Science. Springer, 2005, pp. 413–430.

[121] Rricha Jalota, Nikit Srivastava, Daniel Vollmers, **René Speck**, Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "Finding datasets in publications: The University of Paderborn approach." In: *Rich Search and Discovery for Research Datasets*. Ed. by Julia I. Lane, Ian Mulvany, and Paco Nathan. SAGE Publications Ltd, 2020, pp. 129–141.

[122] Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, Andreas Both, and Sören Auer. "Information extraction pipelines for knowledge graphs." In: *Knowledge and Information Systems* 65 (Jan. 2023).

[123] George H. John and Pat Langley. "Estimating Continuous Distributions in Bayesian Classifiers." In: *Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: Morgan Kaufmann, 1995, pp. 338–345.

[124] Armand Joulin, Edouard Grave, Piotr Bojanowski, Maximilian Nickel, and Tomas Mikolov. "Fast Linear Model for Knowledge Graph Embeddings." In: *arXiv preprint arXiv:1710.10881* (2017).

[125] Giuseppe Rizzo Julien Plu and Raphaël Troncy. "An Hybrid Approach for Entity Recognition and Linking." In: *Proceedings of the OKE Challenge 2015 co-located with the 12th Extended Semantic Web Conference (ESWC 2015)*. 2015.

[126] Dan Jurafsky and James H. Martin. *Speech and Language Processing*. 3rd (Draft). Last time accessed, July 20th, 2021. Dec. 2020.

[127] Fahd Kalloubi and Omar El Nfaoui El Habib Beqaali. "Microblog semantic context retrieval system based on linked open data and graph-based theory." In: *Expert Systems with Applications* 53.C (2016), pp. 138–148.

[128] Aleksandar Kaplar, Milan Stošović, Aleksandra Kaplar, Voin Brković, Radomir Naumović, and Aleksandar Kovačević. "Evaluation of clinical named entity recognition methods for Serbian electronic health records." In: *International Journal of Medical Informatics* 164 (2022), p. 104805.

[129] Mahboob Alam Khalid, Valentin Jijkoun, and Maarten De Rijke. "The Impact of Named Entity Normalization on Information Retrieval for Question Answering." In: *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*. Springer-Verlag, 2008, pp. 705–710.

[130] Ali Khalili, Sören Auer, and Axel-Cyrille Ngonga Ngomo. "conTEXT – Lightweight Text Analytics using Linked Data." In: *Extended Semantic Web Conference (ESWC 2014)*. 2014.

[131] Su Nam Kim and Min-Yen Kan. "Re-examining automatic keyphrase extraction approaches in scientific articles." In: *MWE '09*. 2009, pp. 9–16.

[132] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. "SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles." In: *SemEval*. 2010, pp. 21–26.

[133] J. Kittler, M. Hatef, R. P W Duin, and J. Matas. "On combining classifiers." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.3 (Mar. 1998), pp. 226–239.

[134] Dan Klein and Christopher D Manning. "Fast exact inference with a factored model for natural language parsing." In: *Advances in neural information processing systems* 15 (2002).

[135] Dan Klein and Christopher D Manning. "Accurate unlexicalized parsing." In: *Proceedings of the 41st annual meeting of the association for computational linguistics*. 2003, pp. 423–430.

[136] Graham Klyne and Jeremy J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C. 2004. URL: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[137] Ron Kohavi. "The Power of Decision Tables." In: *8th European Conference on Machine Learning*. Springer, 1995, pp. 174–189.

[138] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. "End-to-End Neural Entity Linking." In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2018, pp. 519–529.

[139] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. "Learning concept lengths accelerates concept learning in ALC." In: *The Semantic Web: 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29–June 2, 2022, Proceedings*. Springer. 2022, pp. 236–252.

[140] Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. "Large-scale learning of relation-extraction rules with distant supervision from the web." In: *International Semantic Web Conference*. Springer. 2012, pp. 263–278.

[141] Joseph B Kruskal. "On the shortest spanning subtree of a graph and the traveling salesman problem." In: *Proceedings of the American Mathematical society* 7.1 (1956), pp. 48–50.

[142] Alberto HF Laender, Berthier A Ribeiro-Neto, Altigran S da Silva, and Juliana S Teixeira. "A brief survey of web data extraction tools." In: *ACM Sigmod Record* 31.2 (2002), pp. 84–93.

[143] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural architectures for named entity recognition." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 260–270.

[144] Niels Landwehr, Mark Hall, and Eibe Frank. "Logistic Model Trees." In: *Machine Learning* 95.1-2 (2005), pp. 161–205.

[145] P. Le and I. Titov. "Distant learning for entity linking with automatic noise detection." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 4081–4090.

[146] Phong Le and Ivan Titov. "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)*. Association for Computational Linguistics, 2018, pp. 1595–1604.

[147] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents." In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. JMLR.org, 2014, pp. 1188–1196.

[148] Freddy Lecue. "On the role of knowledge graphs in explainable AI." In: *Semantic Web* 11.1 (2020), pp. 41–51.

[149] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. "Deterministic Coreference Resolution Based on Entity-centric, Precision-ranked Rules." In: *Computational Linguistics* 39.4 (Dec. 2013), pp. 885–916.

[150] Jens Lehmann, Chris Bizer, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. "DBpedia - A Crystallization Point for the Web of Data." In: *Journal of Web Semantics* 7.3 (2009), pp. 154–165.

[151] Jens Lehmann and Lorenz Bühmann. "AutoSPARQL: Let Users Query Your Knowledge Base." In: *The Semantic Web: Research and Applications*. Ed. by Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Pan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 63–79.

[152] Omer Levy and Yoav Goldberg. "Dependency-based word embeddings." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2014, pp. 302–308.

[153] Jiayi Li, Sheetal Satheesh, Stefan Heindorf, Diego Moussallem, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "AutoCL: AutoML for Concept Learning." In: *The 2nd World Conference on eXplainable Artificial Intelligence (xAI-2024)*. 2024.

[154] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. "A survey on deep learning for named entity recognition." In: *IEEE transactions on knowledge and data engineering* 34.1 (2020), pp. 50–70.

[155] Qi-na Li and Tinghui li. "Research on the application of Naive Bayes and Support Vector Machine algorithm on exercises Classification." In: *Journal of Physics: Conference Series* 1437 (Jan. 2020), p. 012071.

[156] Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. "Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 2664–2669.

[157] Yang Li, Shulong Tan, Huan Sun, Jiawei Han, Dan Roth, and Xifeng Yan. "Entity disambiguation with linkless knowledge bases." In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conference Steering Committee, 2016, pp. 1261–1270.

[158] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. "Mining evidences for named entity disambiguation." In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 1070–1078.

[159] Xiao Ling and Daniel Weld. "Fine-grained entity recognition." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 26. 1. 2012, pp. 94–100.

[160] Chen Liu, Feng Li, Xian Sun, and Hongzhe Han. "Attention-Based Joint Entity Linking with Entity Embedding." In: *Information* 10.2 (2019), p. 46.

[161] Gang Luo, Xiaojiang Huang, Chin-yew Lin, and Zaiqing Nie. "Joint Named Entity Recognition and Disambiguation." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 879–888.

[162] Klaus Lyko, Konrad Höffner, **René Speck**, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. "SAIM—One Step Closer to Zero-Configuration Link Discovery." In: *Proc. of the Extended Semantic Web Conference Posters & Demos*. 2013.

[163] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. "YAGO3: A knowledge base from multilingual Wikipedias." In: *CIDR*. CIDR 2015. 2014.

[164] J Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. "Information Extraction meets the Semantic Web: A Survey." In: *Semantic Web journal* (2018).

[165] Jose L. Martinez-Rodriguez, Ivan Lopez-Arevalo, and Ana B. Rios-Alvarado. "OpenIE-based approach for Knowledge Graph construction from text." In: *Expert Systems with Applications* 113 (2018), pp. 339–355.

[166] Pedro Henrique Martins, Zita Marinho, and André FT Martins. "Joint Learning of Named Entity Recognition and Entity Linking." In: *arXiv preprint arXiv:1907.08243* (2019).

[167] Y. Matsuo and M. Ishizuka. "Keyword Extraction From A Single Document Using Word Co-Occurrence Statistical Information." In: *International Journal on Artificial Intelligence Tools* 13.1 (2004), pp. 157–169.

[168] B. W. Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." In: *Biochim. Biophys. Acta* 405 (1975), pp. 442–451.

[169] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. "Open Language Learning for Information Extraction." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 523–534.

[170] Deborah L McGuinness, Frank Van Harmelen, et al. "OWL web ontology language overview." In: *W3C recommendation* 10.10 (2004), p. 2004.

[171] Edgar Meij, Wouter Weerkamp, and Maarten De Rijke. "Adding semantics to microblog posts." In: *Wsdm 2012* (2012), p. 563.

[172] Igor Aleksandrovic Mel'cuk et al. *Dependency syntax: theory and practice*. SUNY press, 1988.

[173] Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. "DBpedia Spotlight: Shedding Light on the Web of Documents." In: *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*. 2011.

[174] Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandecic, Paul Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble, eds. *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*. Vol. 8796. Lecture Notes in Computer Science. Springer, 2014.

[175] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space." In: *CoRR* (2013). arXiv: 1301.3781.

[176] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., 2013, pp. 3111–3119.

[177] David Milne and Ian H Witten. "Learning to link with wikipedia." In: *Proceedings of the 17th ACM conference on Information and knowledge management*. Association for Computing Machinery, 2008, pp. 509–518.

[178] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. "Distant Supervision for Relation Extraction Without Labeled Data." In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. ACL '09. Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 1003–1011.

[179] Jose G Moreno, Romaric Besançon, Romain Beaumont, Eva D'hondt, Anne-Laure Ligozat, Sophie Rosset, Xavier Tannier, and Brigitte Grau. "Combining Word and Entity Embeddings for Entity Linking." In: *The Semantic Web*. Springer International Publishing, 2017, pp. 337–352.

[180] Andrea Moro, Alessandro Raganato, and Roberto Navigli. "Entity linking meets word sense disambiguation: a unified approach." In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 231–244.

[181] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. "Named entity extraction for knowledge graphs: A literature overview." In: *IEEE Access* 8 (2020), pp. 32862–32881.

[182] Diego Moussallem. "Knowledge Graphs for Multilingual Language Translation and Generation." PhD in Computer Science. PhD thesis. Paderborn University, 2020.

[183] Diego Moussallem, Paramjot Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, Michael Röder, **René Speck**, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. "A General Benchmarking Framework for Text Generation." In: *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Dublin, Ireland (Virtual): Association for Computational Linguistics, Dec. 2020, pp. 27–33.

[184] Diego Moussallem, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "Generating Explanations in Natural Language from Knowledge Graphs." In: *Knowledge Graphs for eXplainable Artificial Intelligence*. Vol. 47. Studies on the Semantic Web. 2020, pp. 213–241.

[185] Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. "MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach." In: *K-CAP 2017: Knowledge Capture Conference*. ACM. 2017, p. 8.

[186] David Mueller and Greg Durrett. "Effective use of context in noisy entity linking." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 1024–1029.

[187] David Nadeau. *Balie—baseline information extraction: Multilingual information extraction from text with machine learning and natural language techniques*. Tech. rep. Technical report, University of Ottawa, 2005.

[188] David Nadeau. "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision." PhD thesis. University of Ottawa, Nov. 2007.

[189] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification." In: *Linguisticae Investigationes* 30.1 (Jan. 2007). Publisher: John Benjamins Publishing Company, pp. 3–26.

[190] David Nadeau, Peter Turney, and Stan Matwin. "Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity." In: 2006, pp. 266–277.

[191] Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. "Fine-grained semantic typing of emerging entities." In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013, pp. 1488–1497.

[192] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. "PATTY: A Taxonomy of Relational Patterns with Semantic Types." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 1135–1145.

[193] Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. "Named entity recognition and relation extraction: State-of-the-art." In: *ACM Computing Surveys (CSUR)* 54.1 (2021), pp. 1–39.

[194] Roberto Navigli and Simone Paolo Ponzetto. "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network." In: *Artificial Intelligence* 193 (2012), pp. 217–250.

[195] Axel-Cyrille Ngonga Ngomo, Sören Auer, Jens Lehmann, and Amrapali Zaveri. "Introduction to Linked Data and Its Lifecycle on the Web." In: *Reasoning Web. Reasoning on the Web in the Big Data Era: 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*. Ed. by Manolis Koubarakis, Giorgos Stamou, Giorgos Stoilos, Ian Horrocks, Phokion Kolaitis, Georg Lausen, and Gerhard Weikum. Cham: Springer International Publishing, 2014, pp. 1–99.

[196] Axel-Cyrille Ngonga Ngomo, Michael Röder, Diego Moussallem, Ricardo Usbeck, and **René Speck**. "BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking." In: *Proceedings of the 11th International Conference on Natural*

Language Generation, Tilburg University, The Netherlands, November 5-8, 2018. Ed. by Emiel Krahmer, Albert Gatt, and Martijn Goudbeek. Association for Computational Linguistics, 2018, pp. 339–349.

[197] Axel-Cyrille Ngonga Ngomo. "Low-Bias Extraction of Domain-Specific Concepts." PhD thesis. University of Leipzig, 2009.

[198] Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, **René Speck**, and Martin Kaltenböck. "SCMS - Semantifying Content Management Systems." In: *ISWC 2011*. 2011.

[199] Axel-Cyrille Ngonga Ngomo, Norman Heino, **René Speck**, and Prodromos Malakasiotis. "A tool suite for creating Question Answering benchmarks." In: *Proceedings of LREC*. 2014.

[200] Axel-Cyrille Ngonga Ngomo and Michael Röder. "HOBBIT: Holistic Benchmarking for Big Linked Data." In: *ESWC, EU networking session*. 2016.

[201] Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. "Relation extraction from wikipedia using subtree mining." In: *AAAI*. 2007, pp. 1414–1420.

[202] Thuy Nguyen and Min-Yen Kan. "Keyphrase Extraction in Scientific Publications." In: 2007, pp. 317–326.

[203] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. "A Review of Relational Machine Learning for Knowledge Graphs." In: *Proceedings of the IEEE* 104.1 (2016), pp. 11–33.

[204] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. "Industry-Scale Knowledge Graphs: Lessons and Challenges." In: *Commun. ACM* 62.8 (July 2019), pp. 36–43.

[205] Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Robert Meusel, and Heiko Paulheim. "The Second Open Knowledge Extraction Challenge." In: *Semantic Web Challenges*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 3–16.

[206] Andrea-Giovanni Nuzzolese, AnnaLisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. "Open Knowledge Extraction Challenge." English. In: *Semantic Web Evaluation Challenges*. Ed. by Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann. Vol. 548. Communications in Computer and Information Science. Springer International Publishing, 2015, pp. 3–15.

[207] Italo L. Oliveira, Renato Fileto, **René Speck**, Luís P.F. Garcia, Diego Moussallem, and Jens Lehmann. "Towards holistic Entity Linking: Survey and directions." In: *Information Systems* 95 (2021), p. 101624.

[208] Italo Lopes Oliveira, Diego Moussallem, Luís Paulo Faina Garcia, and Renato Fileto. "OPTIC: A Deep Neural Network Approach for Entity Linking using Word and Knowledge Embeddings." In: *Proceedings of the 22th International Conference on Enterprise Information Systems*. 2020, pp. 315–326.

[209] Pedro Oliveira and Juan Rocha. "Semantic annotation tools survey." In: *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2013, pp. 301–307.

[210] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap." In: *ArXiv* abs/2306.08302 (2023).

[211] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap." In: *IEEE Transactions on Knowledge and Data Engineering* (2024), pp. 1–20.

[212] Patrick Pantel and Marco Pennacchiotti. "Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations." In: *ACL*. ACL Press, 2006, pp. 113–120.

[213] Youngja Park, Roy J Byrd, and Branimir K Boguraev. "Automatic glossary extraction: beyond terminology identification." In: *COLING '02*. Taipei, Taiwan: Association for Computational Linguistics, 2002, pp. 1–7.

[214] Terence Parr. *The Definitive ANTLR 4 Reference*. 2nd ed. Raleigh, NC: Pragmatic Bookshelf, 2013.

[215] Alberto Parravicini, Rhicheek Patra, Davide B Bartolini, and Marco D Santambrogio. "Fast and Accurate Entity Linking via Graph Embedding." In: *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. Association for Computing Machinery, 2019, p. 10.

[216] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. "Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge." In: *proceedings of the 21st national conference on Artificial intelligence - Volume 2*. Boston, Massachusetts: AAAI Press, 2006, pp. 1400–1405.

[217] Peter Patel-Schneider. "OWL web ontology language semantics and abstract syntax W3C recommendation 10 February 2004." In: *http://www. w3. org/TR/2004/REC-owl-semantics-20040210/* (2007).

[218] Heiko Paulheim. "Knowledge graph refinement: A survey of approaches and evaluation methods." English. In: *Semantic web* 8.3 (2017), pp. 489–508.

[219] Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. "Relation extraction: A survey." In: *arXiv preprint arXiv:1712.05191* (2017).

[220] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2014, pp. 701–710.

[221] Maria Pershina, Yifan He, and Ralph Grishman. "Personalized page rank for named entity disambiguation." In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 238–243.

[222] Minh C Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. "Pair-linking for collective entity disambiguation: Two could be better than all." In: *IEEE Transactions on Knowledge and Data Engineering* 31.7 (2019), pp. 1383–1396.

[223] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. "A Hybrid Approach for Entity Recognition and Linking." In: *Semantic Web Evaluation Challenges*. Ed. by Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann. Cham: Springer International Publishing, 2015, pp. 28–39.

[224] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. "A Hybrid Approach for Entity Recognition and Linking." In: *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*. Ed. by Fabien Gandon, Elena Cabrio, Milan Stankovic, and Antoine Zimmermann. Cham: Springer International Publishing, 2015, pp. 28–39.

[225] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. "Enhancing Entity Linking by Combining NER Models." In: *Semantic Web Challenges*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 17–32.

[226] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. "Enhancing Entity Linking by Combining NER Models." In: *Semantic Web Challenges: Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. Ed. by Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange. Cham: Springer International Publishing, 2016, pp. 17–32.

[227] Julien Plu, Raphaël Troncy, and Giuseppe Rizzo. "ADEL@OKE 2017: A generic method for indexing knowlege bases for entity linking." In: *ESWC 2017, 14th European Semantic Web Conference, Open Extraction Challenge, 28th May-1st June 2017, Portoroz, Slovenia*. P, May 2017.

[228] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[229] Efrén Rama-Maneiro, Juan C Vidal, and Manuel Lama. "Collective disambiguation in entity linking based on topic coherence in semantic graphs." In: *Knowledge-Based Systems* 199 (2020), p. 105967.

[230] Lev Ratinov and Dan Roth. "Design Challenges and Misconceptions in Named Entity Recognition." In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. CoNLL '09. Boulder, Colorado: Association for Computational Linguistics, 2009, pp. 147–155.

[231] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. "CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases." In: *Proceedings of the 26th International Conference on World Wide Web*. 2017, pp. 1015–1024.

[232] Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling Relations and Their Mentions without Labeled Text." In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 148–163.

[233] Martin Riedmiller. "Advanced supervised learning in multi-layer perceptrons — From backpropagation to adaptive learning algorithms." In: *Computer Standards & Interfaces* 16.3 (1994), pp. 265–278.

[234] Ellen Riloff and Rosie Jones. "Learning Dictionaries for Information Extraction by Multi-level Bootstrapping." In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*. AAAI '99/IAAI '99. Orlando, Florida, USA: American Association for Artificial Intelligence, 1999, pp. 474–479.

[235] Michael Röder. "Automating the Discovery of Linking Candidates." PhD in Computer Science. PhD thesis. Paderborn University, 2023.

[236] Michael Röder, Denis Kuchelev, and Axel-Cyrille Ngonga Ngomo. "A Topic Model for the Data Web." In: *Knowledge Graphs and Semantic Web*. Ed. by Fernando Ortiz-Rodriguez, Boris Villazón-Terrazas, Sanju Tiwari, and Carlos Bobed. Cham: Springer Nature Switzerland, 2023, pp. 183–198.

[237] Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and andreas Both. "N^3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format." In: *Proceedings of LREC'14*. 2014.

[238] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. "GERBIL–Benchmarking Named Entity Recognition and Linking Consistently." In: *Semantic Web Journal* (2018).

[239] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. *Techreport for GERBIL 1.2.2 - V1*. Tech. rep. Leipzig University, 2016.

[240] Michael Röder, Ricardo Usbeck, **René Speck**, and Axel-Cyrille Ngonga Ngomo. "CETUS – A Baseline Approach to Type Extraction." In: *1st Open Knowledge Extraction Challenge at International Semantic Web Conference*. 2015.

[241] Benjamin Roth, Nicholas Monath, David Belanger, Emma Strubell, Patrick Verga, and Andrew McCallum. "Building Knowledge Bases with Universal Schema: Cold Start and Slot-Filling Approaches." In: *Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015*. NIST, 2015.

[242] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. "Learning a health knowledge graph from electronic medical records." In: *Scientific reports* 7.1 (2017), p. 5994.

[243] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. "A survey of cross-lingual word embedding models." In: *Journal of Artificial Intelligence Research* 65.1 (2019).

[244] Vladimir Salin, Maria Slastihina, Ivan Ermilov, **René Speck**, Sören Auer, and Sergey Papshev. "Semantic Clustering of Website Based on Its Hypertext Structure." In: *Knowledge Engineering and Semantic Web*. Ed. by Pavel Klinov and Dmitry Mouromtsev. Cham: Springer International Publishing, 2015, pp. 182–194.

[245] G. Sampson. "How Fully Does a Machine-usable Dictionary Cover English Text." In: *Literary and Linguistic Computing* 4.1 (1989).

[246] Robert E. Schapire. "The Strength of Weak Learnability." In: *Mach. Learn.* 5 (2 July 1990), pp. 197–227.

[247] Özge Sevgili, Alexander Panchenko, and Chris Biemann. "Improving Neural Entity Disambiguation with Graph Embeddings." In: *Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, 2019, pp. 315–322.

[248] Wei Shen, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions." In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2015), pp. 443–460.

[249] Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. "Automating RDF Dataset Transformation and Enrichment." In: *12th Extended Semantic Web Conference, Portorož, Slovenia, 31st May - 4th June 2015*. Springer, 2015.

[250] Wei Shi, Siyuan Zhang, Zhiwei Zhang, Hong Cheng, and Jeffrey Xu Yu. "Joint Embedding in Named Entity Linking on Sentence Level." In: *arXiv preprint arXiv:2002.04936* (2020).

[251] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. "Incremental Knowledge Base Construction Using DeepDive." In: *Proc. VLDB Endow.* 8.11 (July 2015), pp. 1310–1321.

[252] Saurabh Shrivastava. "Bring rich knowledge of people, places, things and local businesses to your apps." In: *Bing blogs* (2017).

[253] Nakatani Shuyo. *Language detection library for java*. 2010.

[254] Kuldeep Singh, Isaiah Onando Mulang', Ioanna Lytra, Mohamad Yaser Jaradeh, Ahmad Sakor, Maria-Esther Vidal, Christoph Lange, and Sören Auer. "Capturing Knowledge in Semantically-typed Relational Patterns to Enhance Relation Linking." In: *Proceedings of the Knowledge Capture Conference*. K-CAP 2017. Austin, TX, USA: ACM, 2017, 31:1–31:8.

[255] Amit Singhal. *Introducing the Knowledge Graph: things, not strings*. 2020-11-13. 2012. URL: https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

[256] **René Speck**, Diego Esteves, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. "DeFacto - A Multilingual Fact Validation Interface." In: *14th International Semantic Web Conference (ISWC 2015), 11-15 October 2015, Bethlehem, Pennsylvania, USA (Semantic Web Challenge Proceedings)*. Ed. by Sean Bechhofer and Kostis Kyzirakos. Semantic Web Challenge, International Semantic Web Conference 2015. 2015.

[257] **René Speck**, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. "Twitter Network Mimicking for Data Storage Benchmarking." In: *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. 2021, pp. 298–305.

[258] **René Speck** and Axel-Cyrille Ngonga Ngomo. "On Caching for Local Graph Clustering Algorithms." In: *Australasian Conference on Artificial Intelligence*. 2013, pp. 56–67.

[259] **René Speck** and Axel-Cyrille Ngonga Ngomo. "Ensemble Learning for Named Entity Recognition." In: *The Semantic Web – ISWC 2014*. Vol. 8796. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 519–534.

[260] **René Speck** and Axel-Cyrille Ngonga Ngomo. "Named Entity Recognition using FOX." In: *International Semantic Web Conference 2014 (ISWC2014), Demos & Posters*. 2014.

[261] **René Speck** and Axel-Cyrille Ngonga Ngomo. "Ensemble Learning of Named Entity Recognition Algorithms using Multilayer Perceptron for the Multilingual Web of Data." In: *K-CAP 2017: Knowledge Capture Conference*. ACM. 2017, p. 4.

[262] **René Speck** and Axel-Cyrille Ngonga Ngomo. "On Extracting Relations using Distributional Semantics and a Tree Generalization." In: *Proceedings of The 21th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2018)*. 2018.

[263] **René Speck** and Axel-Cyrille Ngonga Ngomo. "Leopard — A baseline approach to attribute prediction and validation for knowledge graph population." In: *Journal of Web Semantics* (2019).

[264] **René Speck**, Michael Röder, Felix Conrads, Hyndavi Rebba, Catherine Camilla Romiyo, Gurudevi Salakki, Rutuja Suryawanshi, Danish Ahmed, Nikit Srivastava, Mohit Mahajan, and Axel-Cyrille Ngonga Ngomo. "Open Knowledge Extraction Challenge 2018." In: *Semantic Web Evaluation Challenge*. Springer International Publishing, 2018, pp. 39–51.

[265] **René Speck**, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. "Open Knowledge Extraction Challenge 2017." In: *Semantic Web Challenges: Fourth SemWebEval Challenge at ESWC 2017*. Communications in Computer and Information Science. Springer International Publishing, 2017.

[266] Marc Sumner, Eibe Frank, and Mark Hall. "Speeding up Logistic Model Tree Induction." In: *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, 2005, pp. 675–683.

[267] Christine Thielen. "An Approach to Proper Name Tagging for German." In: *In Proceedings of the EACL-95 SIGDAT Workshop*. 1995.

[268] Ilaria Tiddi and Stefan Schlobach. "Knowledge graphs as tools for explainable machine learning: A survey." In: *Artificial Intelligence* 302 (2022), p. 103627.

[269] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. "Fast Random Walk with Restart and Its Applications." In: *Proceedings of the Sixth International Conference on Data Mining*. IEEE Computer Society, 2006, pp. 613–622.

[270] Sebastian Tramp, Norman Heino, Sören Auer, and Philipp Frischmuth. "RDFauthor: Employing RDFa for collaborative Knowledge Engineering." In: *Proceedings of the EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses; 11th October-15th October 2010 - Lisbon, Portugal*. Ed. by P. Cimiano and H.S. Pinto. Vol. 6317. Lecture Notes in Artificial Intelligence (LNAI). Berlin / Heidelberg: Springer, Oct. 2010, pp. 90–104.

[271] Tuan Tran, Nam Khanh Tran, Teka Hadgu Asmelash, and Robert Jäschke. "Semantic Annotation for Microblog Topics Using Wikipedia Temporal Information." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 97–106.

[272] Salvatore Trani, Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. "SEL: a unified algorithm for entity linking and saliency detection." In: *Proceedings of the 2016 ACM Symposium on Document Engineering*. Association for Computing Machinery, 2016, pp. 85–94.

[273] Salvatore Trani, Claudio Lucchese, Raffaele Perego, David E Losada, Diego Ceccarelli, and Salvatore Orlando. "SEL: A unified algorithm for salient entity linking." In: *Computational Intelligence* 34.1 (2018), pp. 2–29.

[274] Peter D. Turney. "Coherent keyphrase extraction via web mining." In: *Proceedings of the 18th international joint conference on Artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 434–439.

[275] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, and Christina Unger. "HAWK - Hybrid Question Answering over Linked Data." In: *12th Extended Semantic Web Conference, Portorož, Slovenia, 31st May - 4th June 2015*. 2015.

[276] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. "7th Open Challenge on Question Answering over Linked Data (QALD-7)." In: *Semantic Web Evaluation Challenge*. Springer International Publishing, 2017, pp. 59–69.

[277] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, SandroAthaide Coelho, Sören Auer, and Andreas Both. "AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data." English. In: *The Semantic Web – ISWC 2014*. Ed. by Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble. Vol. 8796. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 457–471.

[278] Ricardo Usbeck, Michael Röder, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. "Requirements to Modern Semantic Search Engines." In: *KESW*. 2016.

[279] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, **René Speck**, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. "GERBIL – General Entity Annotation Benchmark Framework." In: *24th International Conference on World Wide Web*. 2015.

[280] Denny Vrandečić and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase." In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

[281] D. Walker and R. Amsler. "The Use of Machine-readable Dictionaries in Sublanguage Analysis." In: *Analysing Language in Restricted Domains* (1986).

[282] Gang Wang, Yong Yu, and Haiping Zhu. "PORE: Positive-Only Relation Extraction from Wikipedia Text." In: *ISWC07*. Vol. 4825. LNCS. Berlin, Heidelberg: Springer Verlag, 2007, pp. 575–588.

[283] Qianwen Wang and Mizuho Iwaihara. "Deep Neural Architectures for Joint Named Entity Recognition and Disambiguation." In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2019, pp. 1–4.

[284] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. "Knowledge graph embedding: A survey of approaches and applications." In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[285] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. "Multi-Level Recommendation Reasoning over Knowledge Graphs with Reinforcement Learning." In: *Proceedings of the ACM Web Conference 2022*. WWW '22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 2098–2108.

[286] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. "Knowledge graph and text jointly embedding." In: *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1591–1601.

[287] Feng Wei, Uyen Trang Nguyen, and Hui Jiang. "Dual-FOFE-net Neural Models for Entity Linking with PageRank." In: *arXiv preprint arXiv:1907.12697* (2019).

[288] Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. "A joint model for discovering and linking entities." In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. 2013, pp. 67–72.

[289] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. "The FAIR Guiding Principles for scientific data management and stewardship." In: *Scientific Data* 3 (Mar. 2016), pp. 160018–.

[290] Dekai Wu, Grace Ngai, and Marine Carpuat. "A Stacked, Voted, Stacked Model for Named Entity Recognition." In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. CONLL '03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 200–203.

[291] Gongqing Wu, Ying He, and Xuegang Hu. "Entity linking: an issue to extract corresponding entity with knowledge base." In: *IEEE Access* 6 (2018), pp. 6220–6231.

[292] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. "Joint learning of the embedding of words and entities for named entity disambiguation." In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016, pp. 250–259.

[293] Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. "Unsupervised relation extraction by mining Wikipedia texts using information from the web." In: *ACL*. ACL '09. 2009, pp. 1021–1029.

[294] Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, and Albert Y. Zomaya. "A Review of Ensemble Methods in Bioinformatics." In: *Current Bioinformatics* 5.4 (2010), pp. 296–308.

[295] X. Yang, X. Gu, S. Lin, S. Tang, Y. Zhuang, F. Wu, Z. Chen, G. Hu, and X. Ren. "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)*. 2019, pp. 271–281.

[296] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. "TextRunner: Open Information Extraction on the Web." In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. NAACL-Demonstrations '07. Rochester, New York: Association for Computational Linguistics, 2007, pp. 25–26.

[297] Shmuel Zaks. "Lexicographic generation of ordered trees." In: *Theoretical Computer Science* 10.1 (1980), pp. 63–82.

[298] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. "Quality Assessment for Linked Data: A Survey." In: *Semantic Web Journal* (2015).

[299] Junlang Zhan and Hai Zhao. "Span model for open information extraction on accurate corpus." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 9523–9530.

[300] Botao Zhang, Yong Feng, Lin Fu, Jinguang Gu, and Fangfang Xu. "Candidate Set Expansion for Entity and Relation Linking Based on Mutual Entity—Relation Interaction." In: *Big Data and Cognitive Computing* 7.1 (2023).

[301] Jingsong Zhang, Yinglin Wang, and Dingyu Yang. "CCSpan: Mining closed contiguous sequential patterns." In: *Knowledge-Based Systems* 89 (2015), pp. 1–13.

[302] Shiliang Zhang, Hui Jiang, Mingbin Xu, Junfeng Hou, and Lirong Dai. "The fixed-size ordinally-forgetting encoding method for neural network language models." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 495–500.

[303] Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. "Entity linking leveraging: automatically generated annotation." In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1290–1298.

[304] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. "Aligning knowledge and text embeddings by entity descriptions." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 267–272.

[305] GuoDong Zhou and Jian Su. "Named entity recognition using an HMM-based chunk tagger." In: *Proceedings of ACL*. 2002, pp. 473–480.

[306] GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. "Exploring various knowledge in relation extraction." In: *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*. 2005, pp. 427–434.

[307] Ganggao Zhu and Carlos A Iglesias. "Exploiting semantic similarity for named entity disambiguation in knowledge graphs." In: *Expert Systems with Applications* 101 (2018), pp. 8–24.

DECLARATION

Hiermit versichere ich, dass ich die vorliegende Dissertation eigenständig und ohne unzulässige Hilfe von Dritten verfasst habe. Ich habe ausschließlich die angegebenen Quellen und Hilfsmittel verwendet. Alle wörtlichen oder sinngemäßen Zitate aus veröffentlichten oder unveröffentlichten Schriften sowie mündlichen Auskünften habe ich gekennzeichnet. Darüber hinaus sind sämtliche Materialien oder Dienstleistungen, die von anderen Personen zur Verfügung gestellt wurden, ebenso gekennzeichnet.

Germany, July 12, 2024

René Speck