SAPIENZA
UNIVERSITÀ DI ROMA

Towards Comprehensive and Efficient Information Extraction Across Languages

Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG)
Dottorato Nazionale in Intelligenza Artificiale (XXXVII cycle)

Simone Tedeschi
ID number 1762897

Advisor
Prof. Roberto Navigli

Academic Year 2023/2024

Thesis defended on 24 January 2025
in front of a Board of Examiners composed by:

Prof. Gabriella Pasi (chairman)
Prof. Stefania Costantini
Prof. Antonio Chella
Prof. Giovanni Semeraro

This thesis was reviewed by two external reviewers:

Prof. Alan Akbik
Dr. Shervin Malmasi

Towards Comprehensive and Efficient Information Extraction Across Languages
PhD thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: tedeschi@diag.uniroma1.it

Dedicated to

my family

Abstract

The exponential growth of textual data shared online has created an urgent need for methods that can effectively extract, structure, and interpret information from vast and varied texts. Information Extraction (IE), a key area within Natural Language Processing (NLP), addresses this need by transforming unstructured text into structured formats enabling automated text analytics and decision-making. However, existing IE systems face substantial challenges in scalability and generalization. These challenges include limited labeled data for low-resource languages, computational demands that restrict accessibility to only well-resourced institutions, and a predominant focus on popular entities. Additionally, most IE tasks are entity-centric tasks (e.g. Named Entity Recognition, Entity Disambiguation, and Relation Extraction), thus overlooking the broader contextual richness present in many texts.

This thesis aims at advancing the field of IE by tackling these critical issues through novel resources, methodologies, and theoretical approaches aimed at fostering a multilingual, scalable, and semantically-enriched IE framework. To bridge the multilingual gap, we leverage a combination of neural and knowledge-based approaches and create multilingual datasets for NER and Relation Extraction, ensuring that IE systems can operate effectively across diverse linguistic settings. On the computational front, we propose optimizations designed to reduce the resource requirements of IE models, especially in the context of Entity Disambiguation, enabling broader adoption of NLP technologies by reducing dependence on high-performance hardware and extensive labeled datasets.

Additionally, this work challenges traditional IE frameworks by expanding the focus beyond named entities to encompass abstract concepts, idiomatic expressions, and tail entities, which are essential for a more nuanced and comprehensive understanding of texts. Through these contributions, this research aims to establish a robust foundation for multilingual, resource-efficient IE systems that can meet the evolving demands of global text analytics across varied languages, domains, and cultural contexts. Finally, to further encourage the usage and development of multilingual IE systems, we publicly release all the artifacts – datasets and models – introduced in this thesis.

Contents

1 Introduction **1**

1.1 A Brief Introduction to Information Extraction 1

1.2 The Multilingual Gap 3

1.3 Computational Challenges 4

1.4 Named Entities Are Not Enough 5

1.5 Thesis Statement and Research Objectives 6

1.6 Publications 8

1.7 Thesis Outline 10

2 Background and Related Work **13**

2.1 Named Entity Recognition 13

2.1.1 Task Formulation 13

2.1.2 Evaluation Metrics 13

2.1.3 Current Trends 14

2.1.4 Challenges and Limitations 17

2.2 Entity Disambiguation 18

2.2.1 Task Formulation 18

2.2.2 Evaluation Metrics 19

2.2.3 Current Trends 20

2.2.4 Challenges and Limitations 21

2.3 Relation Extraction 22

2.3.1 Task Formulation 22

2.3.2 Evaluation Metrics 22

2.3.3 Current Trends . . . 23
2.3.4 Challenges and Limitations . . . 24
2.4 Other Tasks . . . 25
2.4.1 Concept Recognition . . . 25
2.4.2 Idiomatic Expression Identification . . . 28

3 Multilingual Named Entity Recognition 31
3.1 Silver-Data Creation . . . 31
3.1.1 Preprocessing Wikipedia . . . 32
3.1.2 Identifying Entity Mentions in Wikipedia . . . 33
3.1.3 Tagging Named Entity Links Through Synsets . . . 35
3.1.4 Named Entity Tag Propagation . . . 36
3.1.5 Confirming and Augmenting Annotations . . . 37
3.1.6 Experimental Setup . . . 37
3.1.7 Results . . . 40
3.1.8 Ablation Study . . . 41
3.2 Domain Adaptation . . . 43
3.2.1 Category selection . . . 43
3.2.2 Domain embedding computation and domain extraction . . . 45
3.2.3 Experimental Setup . . . 46
3.2.4 Results . . . 47
3.3 Fine-Grained NER . . . 47
3.3.1 Our New Set of Classes . . . 48
3.3.2 Fine-grained Silver Data Creation with Self-Training . . . 49
3.3.3 Experimental Setup . . . 51
3.3.4 Results . . . 53
3.4 Open-Source Data and Models . . . 55
3.5 Summary . . . 55

4 Low-Resource Entity Disambiguation 57
4.1 Budget-Constrained Entity Disambiguation . . . 57
4.1.1 Baseline System . . . 58

4.1.2 Fine-Grained Classes for NER . . . 59
4.1.3 NER-Enhanced Entity Representation . . . 59
4.1.4 NER-Enhanced Candidate Generation . . . 60
4.1.5 NER-based Negative Sampling . . . 60
4.1.6 NER-Constrained Decoding . . . 62
4.1.7 Combinations of NER Contributions . . . 62
4.1.8 Experimental Setup . . . 63
4.1.9 Results . . . 64
4.1.10 Analysis . . . 66
4.2 Overshadowed Entities . . . 68
4.2.1 The NICE method . . . 70
4.2.2 Candidate filtering . . . 70
4.2.3 Candidate pre-scoring . . . 71
4.2.4 Collective disambiguation . . . 71
4.2.5 Parameters of the method . . . 73
4.2.6 Implementation details . . . 75
4.2.7 Experimental setup . . . 75
4.2.8 Results and discussion . . . 77
4.3 Open-Source Data and Models . . . 79
4.4 Summary . . . 79

5 Multilingual Relation Extraction 81
5.1 Resource Creation . . . 81
5.1.1 Data Extraction . . . 82
5.1.2 Manual Annotation . . . 83
5.1.3 Triplet Critic . . . 88
5.1.4 Entity Typing . . . 89
5.1.5 SRED^{FM} . . . 91
5.2 mREBEL . . . 92
5.3 Experimental Setup . . . 94
5.4 Results . . . 95
5.5 Error Analysis . . . 97

5.6 Open Source Data and Models 99
5.7 Summary 100

6 Beyond Named Entities **101**
6.1 Concepts 101
6.1.1 The CNER task 101
6.1.2 Categories 103
6.1.3 Dataset 105
6.1.4 CNER$_{silver}$ 105
6.1.5 CNER$_{gold}$ 108
6.1.6 Dataset Statistics 109
6.1.7 Experimental Setup 110
6.1.8 Results 112
6.1.9 Qualitative Analysis 114
6.2 Idiomatic Expressions 116
6.2.1 Silver-Standard Data Creation 118
6.2.2 Gold-Standard Data Creation 120
6.2.3 Idiom Identification: A New Task Formulation 121
6.2.4 Our System 122
6.2.5 Experimental Setup 122
6.2.6 Results 124
6.2.7 Qualitative Analysis 127
6.2.8 Open Source Data and Models 128

7 Conclusions and Future Works **129**
7.1 Summary of Contributions 129
7.2 Future Works 131

Bibliography **133**

Chapter 1

Introduction

1.1 A Brief Introduction to Information Extraction

From news articles and legal documents to social media posts and scientific literature, texts are filled with valuable information that must be extracted, structured, and understood in order to derive meaningful insights. However, the sheer volume and complexity of these texts present a significant challenge. In this context, Information Extraction (IE) stands out as a fundamental research area within Natural Language Processing (NLP) that aims to automatically distill structured information from unstructured textual data. Simply put, IE addresses the challenge of transforming raw text into concise, structured representations by identifying and categorizing specific pieces of information (Grishman, 2015). For instance, from the sentence *"The CEO of XYZ Corporation, John Doe, announced a new product launch in Berlin on Wednesday"*, an IE system should ideally extract the following critical information units: *John Doe* as a person, *XYZ Corporation* as an organization, *new product launch* as an event, *Berlin* as a location, *Wednesday* as a time entity. Additionally, it should recognize *John Doe* as the CEO of *XYZ Corporation*.

IE is often broken down into a variety of subtasks, each targeting specific aspects of the extraction process. Named Entity Recognition (NER), for example, focuses on identifying and classifying entities in text, such as names, locations, dates, events, and organizations (Nadeau and Sekine, 2007a). Building on this, Entity Disambiguation (ED) resolves the identified entities with a unique identifier, typically within a knowledge base like Wikipedia or Wikidata. For instance, an ED system can determine whether *"Paris"*

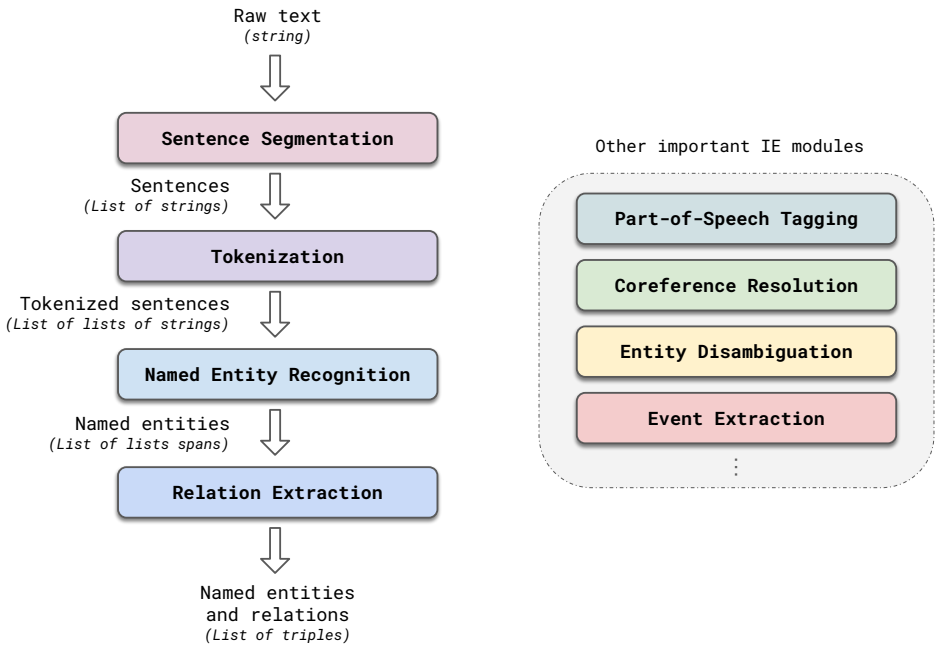

Figure 1.1. A standard Information Extraction (IE) pipeline on the left with some other important (optional) IE modules on the right.

refers to the city in France, the person *Paris Hilton*, or *Paris, Texas*, based on context (Sevgili et al., 2022). Relation Extraction (RE), on the other hand, seeks to determine relationships between these entities, such as employment, ownership, or family relations (Pawar et al., 2017b). Together with other basic NLP modules, these subtasks form the foundational elements of an information extraction pipeline (cf. Figure 1.1). This process – namely identifying and understanding entities together with their relationships – is crucial for enabling automated text analytics as well as more complex reasoning tasks. Indeed, IE systems and their sub-modules are typically leveraged in downstream applications such as question answering (Jijkoun et al., 2004; Fader et al., 2014), text summarization (Hachey, 2009; Bharti and Babu, 2017), and knowledge graph construction (Jaradeh et al., 2023), *inter alia*.

Nevertheless, to fully benefit from the potential of IE systems, several challenges still need to be addressed. The primary limitation lies in the scarcity of labeled data for many languages which prevents the large-scale usage of IE systems (Faisal et al., 2022). Additionally, even when this data is available, it is typically domain-specific or biased toward particular entities and relations, limiting the generalization capabilities across domains of the

resulting systems (Huang et al., 2024). Yet another challenge is the substantial computational resources required to effectively train IE systems, which are often accessible only to a limited group of researchers or organizations (Petroni et al., 2021). Finally, entity-centric tasks alone might not be sufficient to capture the full richness of textual information (Katz et al., 1998; Navigli, 2009). In the remainder of this chapter, we go deeper into these problems (Sections 1.2, 1.3, and 1.4), outline the thesis objectives (Section 1.5) and highlight our contributions to the field (Section 1.6). Finally, Section 1.7 provides a comprehensive overview of the structure of this thesis.

1.2 The Multilingual Gap

The number of documents published on the Web in languages other than English grows rapidly every year, underscoring the urgent need for IE systems that can operate effectively across diverse linguistic contexts. However, while high-resource languages like English benefit from extensive datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003a; Balasuriya et al., 2009; Ritter et al., 2011a; Derczynski et al., 2017; Zhang et al., 2017; Petroni et al., 2021) and well-developed tools (Botha et al., 2020; Cao et al., 2021; Huguet Cabot and Navigli, 2021), most other languages face a severe shortage of labeled corpora (Joshi et al., 2020; Faisal et al., 2022). This data scarcity hampers the performance of IE systems when applied to non-English languages, leading to a significant performance gap between high-resource and low-resource languages, consequently limiting their applicability in multilingual settings.

In response to the data scarcity in multilingual NLP, several multilingual Pretrained Language Models (PLMs) have been designed. These models, including multilingual BERT (Devlin et al., 2019, mBERT), XLM-R (Conneau et al., 2020), and newer architectures such as mT5 (Xue et al., 2021), have established significant advancements in handling a wide range of languages with a single model architecture. In particular, mBERT marked a significant milestone by pretraining the BERT transformer architecture on a wide array of languages using the Masked Language Modeling (MLM) objective. This enables the model to capture a generalized understanding of multiple languages, allowing for zero-shot and few-shot transfer between high-resource and low-resource languages. In other words,

this process allows mBERT and its successors to apply a model trained on one language to another language with some degree of effectiveness (Pires et al., 2019). Based on these findings, multiple studies tried to apply cross-lingual transfer learning techniques in the context of information extraction (Wang et al., 2013; Nag et al., 2023; Hazem et al., 2022; Taghizadeh and Faili, 2022). However, while multilingual models like mBERT and XLM-R are pretrained on generalized language tasks, they are not inherently optimized for specific tasks like IE. Additionally, depending on the language, certain cultural and contextual distinctions, such as regional entities and local references further reduce the efficacy of transfer learning strategies (Williams et al., 2020). To excel in specialized tasks, PLMs require fine-tuning on language-, domain- and task-specific data. Essentially, fine-tuning is needed to successfully address tasks such as the recognition of entities or the extraction of relationships from text, which general pretraining objectives do not fully cover.

Addressing the multilingual gap in IE therefore requires concerted efforts to produce high-quality, annotated datasets across languages and domains allowing models to adapt to specific tasks in several languages. Investing in these areas is pivotal, as effective multilingual IE systems are essential not only for enabling global information access, but also for making NLP technologies inclusive and accessible across linguistic and cultural boundaries.

1.3 Computational Challenges

The advent of transformer architectures has revolutionized the NLP field, offering unprecedented performance in various tasks, including information extraction. However, these gains have come at a significant computational cost, with the increased complexity of transformer models, such as BERT (Devlin et al., 2019), BART (Lewis et al., 2019), GPT (Radford et al.), and T5 (Raffel et al., 2023), necessitating substantial computational resources for both training and inference. Additionally, the rapid progression to models like GPT-3 (Brown et al., 2020), which has 175 billion parameters compared to approximately 110M parameters of BERT, exemplifies the ongoing exponential growth in model size. Therefore, training state-of-the-art transformers requires access to high-performance hardware, such as GPUs or TPUs, and vast memory resources, as well as extended time frames for training. Such

computational requirements present a significant barrier to entry for many research teams and organizations, limiting access to institutions with sufficient resources. Moreover, the environmental impact of training large-scale models, often measured by carbon emissions, adds another layer of concern, as model size and training demands continue to escalate (Strubell et al., 2019).

In the context of IE systems, which are typically composed of separate modules, e.g. NER, ED, and Relation Extraction (cf. Figure 1.1), the need to train and maintain separate models for each module exacerbates these computational demands. Additionally, among IE tasks, entity disambiguation is particularly computationally demanding due to its knowledge-intensive nature. Specifically, ED systems require associating textual mentions with entries in external KBs, such as Wikidata or Wikipedia, necessitating both an extensive understanding of contextual cues and access to vast stores of encyclopedic knowledge. Indeed, ED has been included in the Knowledge-Intensive Language Task (KILT) benchmark encompassing tasks that rely heavily on external information sources (Petroni et al., 2021).

Therefore, to ensure large-scale adoption and practical application of IE systems, it is critical to address the above-mentioned computational challenges. Moving forward, a balance must be struck between performance improvements and resource constraints. This involves reducing training times by requiring less labeled data or developing lighter models. This resource-efficient approach to IE aims at prioritizing scalability, inclusivity, and environmental responsibility over pure performance. Resource-aware model design is essential in enabling the practical application of IE systems across industries, regions, and resource settings.

1.4 Named Entities Are Not Enough

As discussed in the previous sections, a substantial portion of the important information in texts is conveyed through named entities – specific references to people, organizations, places, events, and other prominent entities. These named entities serve as the key building blocks for understanding the context and content of the text, as they represent central actors, locations, or objects within a narrative (De Cao, 2024). Moreover, the relationships between

these entities add another layer of meaning, illustrating how entities interact or are connected to each other. For this reason, most of the tasks in IE are entity-centric tasks, i.e. named entity recognition, entity disambiguation, coreference resolution, and relation extraction.

However, we argue that these tasks alone are not sufficient to capture the full richness of textual information. Indeed, texts contain much more than named entities and their relations; they also convey abstract ideas or nominal concepts (Navigli, 2009) and often employ figurative language such as metaphors, idiomatic expressions, or euphemisms (Katz et al., 1998). For instance, concepts like *"climate change"* or *"machine learning"* may not correspond to specific named entities, but they are central to understanding scientific, technical, or ethical discussions. Figurative language, on the other hand, involves expressions where the literal meaning differs from the intended meaning, such as the phrase *"the project is a sinking ship"*, which indicates failure rather than a literal boat. For these reasons, we believe that understanding concepts and interpreting figurative expressions is essential for a deeper and more nuanced understanding of texts.

1.5 Thesis Statement and Research Objectives

The work presented in this thesis can be summarized by the following statement:

> *We aim to address the intra- and inter-linguistic challenges in Information Extraction (IE) by developing novel multilingual resources and methods; our focus includes reducing training times and expanding the reach of IE to handle complex concepts, figurative language, and tail entities, thus paving the way for scalable, widespread adoption of IE systems across diverse linguistic contexts.*

That is, the focus of this thesis is to investigate novel methodologies for IE to achieve performance parity across various languages while enhancing the efficiency and coverage of IE systems in well-resourced languages. The journey toward our overarching objective is marked by numerous challenges, including the creation of multilingual IE datasets, the development of robust models, the proposal of comprehensive evaluation frameworks, and the identification of gaps in current systems that leave critical aspects unaddressed. Moving beyond traditional, entity-centric approaches, we aim to reframe information extraction with a broader, more nuanced understanding of textual content. In summary, we present the

primary objectives that this thesis aims to accomplish:

1. **Creating resources for the diverse IE subtasks across languages.** While IE encompasses a range of subtasks – such as named entity recognition and relation extraction – existing datasets are often limited in both scope and language diversity, especially for low-resource languages. Our first objective is to create multilingual datasets specifically designed to support these subtasks.

2. **Building robust multilingual IE models.** Current multilingual IE models frequently exhibit substantial variability in performance across languages and domains, particularly between high- and low-resource contexts. Our second objective is to develop models that are not only capable of handling linguistic diversity but also demonstrate consistent, high-quality performance across a wide spectrum of languages, thus reducing disparities.

3. **Reducing the computational requirements of IE models.** Modern IE models often require significant computational power for training and inference, which limits their accessibility for researchers and practitioners with limited resources and can impede large-scale deployment. Our third objective is to use lightweight model architectures and optimize training processes to reduce computational demands, making IE systems more efficient and cost-effective to train and deploy without sacrificing performance.

4. **Enhancing IE capabilities for tail and rare entities.** IE models are typically optimized for popular entities, but tail and rare entities are often overlooked, leading to a gap in model understanding and accuracy for less frequently occurring or context-specific terms. Our fourth objective is therefore to improve model performance on less common entities by exploiting contextually relevant information.

5. **Expanding the scope of IE beyond named entities to include concepts and figurative language.** Most existing IE systems are centered on named entities, focusing on proper nouns like names, locations, and organizations, leaving out complex forms of meaning such as abstract concepts and figurative expressions. Our fifth objective is to broaden IE capabilities to capture these complex language forms, allowing for a nuanced and comprehensive extraction of meaning from diverse texts.

Our overall objective is therefore to break down the obstacles that still prevent the effective adoption and integration of IE systems, especially in multilingual settings.

1.6 Publications

The research presented in this thesis has been jointly conducted at Sapienza NLP – the research group directed by Professor Roberto Navigli at Sapienza University of Rome – and Babelscape – a deep tech company focused on multilingual NLP – over the three years of the Ph.D. program. The majority of this thesis has already been published in top-tier NLP and AI conferences, including ACL, NAACL and EMNLP, among others. In this thesis, we discuss in-depth only those publications that are thematically close to the aforementioned research objectives. In what follows, we provide a reference to these publications in chronological order:

1. **Tedeschi, S.**, Maiorca, V., Campolungo, N., Cecconi, F., & Navigli, R. (2021, November). WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. *In Findings of the association for computational linguistics: EMNLP 2021 (pp. 2521-2533).*

2. **Tedeschi, S.**, Conia, S., Cecconi, F., & Navigli, R. (2021, November). Named Entity Recognition for Entity Linking: What works and what's next. *In Findings of the Association for Computational Linguistics: EMNLP 2021 (pp. 2584-2596).*

3. **Tedeschi, S.**, & Navigli, R. (2022, July). MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). *In Findings of the Association for Computational Linguistics: NAACL 2022 (pp. 801-812).*

4. **Tedeschi, S.**, Martelli, F., & Navigli, R. (2022, July). ID10M: Idiom identification in 10 languages. In Findings of the Association for Computational Linguistics: NAACL 2022 (pp. 2715-2726).

5. **Tedeschi, S.**, & Navigli, R. (2022, July). NER4ID at SemEval-2022 Task 2: Named Entity Recognition for Idiomaticity Detection. *In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) (pp. 204-210).*

6. Provatorova, V., **Tedeschi, S.**, Vakulenko, S., Navigli, R., & Kanoulas, E. (2022). Focusing on Context is NICE: Improving Overshadowed Entity Disambiguation. *arXiv preprint arXiv:2210.06164.*

7. Cabot, P. L. H., **Tedeschi, S.**, Ngomo, A. C. N., & Navigli, R. (2023, July). REDFM: a Filtered and Multilingual Relation Extraction Dataset. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 4326-4343).*

8. Martinelli, G., Molfese, F., **Tedeschi, S.**, Fernández-Castro, A., & Navigli, R. (2024, June). CNER: Concept and Named Entity Recognition. *In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 8329-8344).*

In addition, we will briefly discuss key aspects, insights, and results from other efforts we carried out throughout the Ph.D. program. More specifically, we will reference material from the following publications:

9. Keh, S. S., Bharadwaj, R., Liu, E., **Tedeschi, S.**, Gangal, V., & Navigli, R. (2022, December). EUREKA: EUphemism Recognition Enhanced through Knn-based methods and Augmentation. *In Proceedings of the 3rd Workshop on Figurative Language Processing (FLP) (pp. 111-117).*

10. Barba, E., Campolungo, N., **Tedeschi, S.**, & Navigli, R. (2023, Mar). On the Intra- and Inter-linguistic Challenges of Multilingual Silver-Data Creation and Disambiguation Biases. *UniDive First General Meeting (Research Communication).*

11. **Tedeschi, S.**, Bos, J., Declerck, T., Hajic, J., Hershcovich, D., Hovy, E., Koller, A., Krek, S., Schockaert, S., Sennrich, R., Shutova, E., & Navigli, R. (2023, July). What's the Meaning of Superhuman Performance in Today's NLU?. *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 12471-12491).*

12. Molfese, F., Bejgu, A. S., **Tedeschi, S.**, Conia, S., & Navigli, R. (2024). CroCoAlign: A Cross-Lingual, Context-Aware and Fully-Neural Sentence Alignment System for

Long Texts. *In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 2209-2220). Association for Computational Linguistics.*

13. Nakamura, T.*, Mishra, M.*, **Tedeschi, S.***, Chai, Y., Stillerman, J. T., Friedrich, F., ... & Pyysalo, S. (2024). Aurora-m: The first open source multilingual language model red-teamed according to the us executive order. *arXiv preprint arXiv:2404.00399.*

14. **Tedeschi, S.**, Friedrich, F., Schramowski, P., Kersting, K., Navigli, R., Nguyen, H., & Li, B. (2024). ALERT: A Comprehensive Benchmark for Assessing Large Language Models' Safety through Red Teaming. *arXiv preprint arXiv:2404.08676.*

15. Ghinassi, I., **Tedeschi, S.**, Marongiu, P., Navigli, R., & McGillivray, B. (2024, May). Language Pivoting from Parallel Corpora for Word Sense Disambiguation of Historical Languages: A Case Study on Latin. *In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 10073-10084).*

16. Proietti, L., Perrella, S., **Tedeschi, S.**, Vulpis, G., Lavalle, L., Sanchietti, A., ... & Navigli, R. (2024, May). Analyzing Homonymy Disambiguation Capabilities of Pretrained Language Models. *In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 924-938).*

17. Teglia, S., **Tedeschi, S.**, & Navigli, R. (2024, Sep). How Much Do Pretrained Language Models Know About Word Senses? *Under review at NAACL 2025.*

18. Scirè, A., Bejgu, S.,**Tedeschi, S.**, Ghonim, K., Martelli, F., & Navigli, R. (2024, Sep). Truth or Mirage? Towards End-To-End Factuality Evaluation with LLM-OASIS *Under review at Computational Linguistics.*

1.7 Thesis Outline

The thesis is structured into four main chapters, with each chapter focusing on one or more of our publications. These chapters are not arranged in the chronological order of the

corresponding publications, but rather follow the standard flow of an information extraction pipeline, i.e. NER, entity disambiguation, and relation extraction, subsequently followed by other newly-introduced tasks such as Concept Recognition (CR) and figurative language processing. We believe that this organization allows readers to examine each challenge individually and gain a clearer understanding of the purpose behind each study.

Before introducing these four core chapters, Chapter 2 provides a formal definition of the primary IE tasks and a comprehensive review of the relevant literature, covering both foundational and recent studies. This chapter contextualizes our research objectives by discussing essential resources and key methodological approaches that have shaped current practices in IE.

In Chapter 3, we focus on enabling multilingual named entity recognition and introduce our novel language-agnostic methodology to produce high-quality training data for the task. Subsequently, to enhance the recognition capabilities of NER models in specific domains (such as *sport*, *politics*, or *finance*), we introduce a new domain adaptation strategy that enables the creation of domain-specific datasets. Moreover, to further enhance the applicability of NER systems in specific domains, we expand our methodologies to create datasets annotated with fine-grained NER categories. Finally, based on the data introduced in this chapter, we train and release accurate and efficient models for multilingual NER, each containing less than 200M parameters.

In Chapter 4, instead, in contrast with recent trends, we explore the challenges of low-resource entity disambiguation, a task that relies heavily on external knowledge to accurately identify and differentiate entities. Specifically, we start by proposing strategies aimed at enhancing the performance of an ED system in settings with budget constraints, considerably reducing the amount of resources needed to successfully train a system. Following this, we introduce an innovative technique designed to boost disambiguation performance specifically for overshadowed and tail entities, which are often underrepresented in the training datasets.

In Chapter 5, we proceed with relation extraction and present our methodology for automatically creating multilingual training data for the task together with a new evaluation benchmark. Additionally, we release a new family of small specialized models compared to recent LLMs (i.e. with an overall number of parameters comprised between 0.4B and 0.7B) capable of extracting hundreds of relation types in 18 languages.

In Chapter 6, we expand the coverage of information extraction systems by moving beyond traditional entity-centric tasks. We begin by introducing the Concept Recognition (CR) task and integrating it with the NER task. Additionally, we propose a novel methodology for automatically generating training data for the idiom identification task, and frame it it as a sequence labeling task. Utilizing the datasets developed for these two tasks, we train and release efficient and accurate systems. Notably, the incorporation of these new modules into the information extraction pipeline enables the identification and classification of a broader range of information from text.

Finally, Chapter 7 presents the conclusions of this dissertation, summarizing our key findings and outlining potential directions for future research.

Chapter 2

Background and Related Work

2.1 Named Entity Recognition

2.1.1 Task Formulation

Named Entity Recognition (NER) is a subtask of Natural Language Processing (NLP) that focuses on identifying and classifying named entities within a text (Nadeau and Sekine, 2007a). These entities refer to specific real-world objects such as people, organizations, locations, dates, quantities, or domain-specific terms like genes, chemicals or diseases in biomedical texts (Weber et al., 2021; Sänger et al., 2024). The goal of NER is therefore to detect and categorize these entities into predefined semantic types.

Formally, given a sequence of tokens denoted as $T = (t_1, t_2, \ldots, t_N)$, the task of NER involves generating a set of tuples (s, e, ℓ), where s and e are integers within the range $[1, N]$ representing the starting and ending indices of a named entity, respectively, while ℓ refers to the entity type from a predefined set of categories. For instance, in the sentence "Elon Mask is the CEO of SpaceX", a NER system would identify *Elon Musk* as a person and *SpaceX* as a location, as shown in Figure 2.1.

2.1.2 Evaluation Metrics

NER systems are evaluated using precision, recall, and F1-score, calculated at both the micro and macro levels. The micro-F1 score aggregates true positives, false positives, and false negatives across all entities, providing a metric that emphasizes the system's performance

across all mentions. Meanwhile, the macro-F1 score calculates the F1-score for each entity type separately and then averages these scores, which highlights the model's performance across different entity categories. The formulas for precision, recall, and F1-score are:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Importantly, NER typically uses a strict span-based evaluation, where a predicted entity span must match exactly with the annotated span in both boundaries and type to be counted as a true positive. This precise evaluation ensures accurate assessment of the model's ability to identify and categorize entities correctly, although in some cases, relaxed boundaries are allowed.

2.1.3 Current Trends

Methods. Early NER systems relied heavily on handcrafted, domain-specific rules and features. For example, (Borkowski and Watson, 1967) proposed an algorithm that uses rule-based lists and indicators, such as capital letters, to identify company names. Lexical markers are elements that accompany named entities, helping to indicate their presence (for example, titles like Mr. or Ms. before names). Approaches in this category typically do not require annotated data and primarily rely on rules and dictionaries based on lexical resources and domain-specific knowledge. However, these methods depend on domain experts to generate numerous rules using lexical markers (Zhang and Elhadad, 2013) or entity dictionaries (Sekine and Nobata, 2004; Etzioni et al., 2005). Furthermore, within this group of methods, precision tends to be high, while recall is often low due to the specificity of domain and language rules, as well as incomplete dictionaries. Additionally, the process of adding new rules and dictionaries can be expensive. To mitigate these drawbacks, Fetahu et al. (2022) proposed a flexible approach to jointly integrate both textual and gazetteer information dynamically, i.e. the external knowledge is accessed only when the textual knowledge is insufficient for performing the NER task.

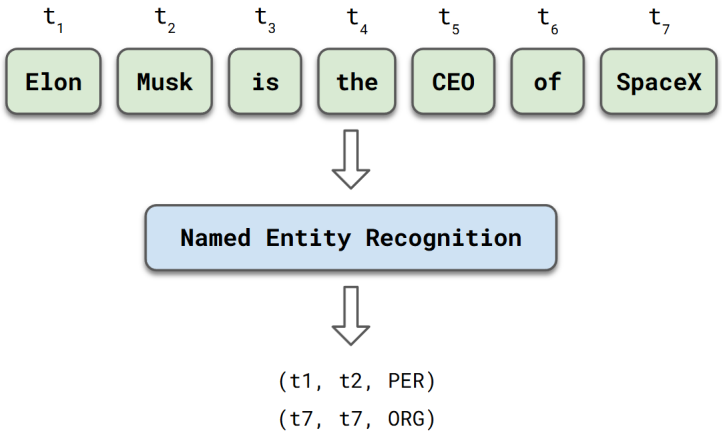

Figure 2.1. Given a sequence of tokens, Named Entity Recognition (NER) identifies the boundaries of named entities and categorizes them by type.

However, to fully address the limitations of rule-based systems, purely data-driven neural methods became the *de facto* standard in NER. Specifically, starting from (Collobert et al., 2011), approaches requiring minimal feature engineering and manual intervention became increasingly popular (Li et al., 2018). Additionally, with the advent of pre-trained language models such as BERT (Devlin et al., 2019), once a sufficient amount of training data is available for the task of interest, fine-tuning is often employed to address the task successfully. In general, most modern NER systems are trained in a supervised learning fashion, thus requiring hundreds or thousands of training examples for learning the task. Notably, since the first shared task on NER organized by Grishman and Sundheim (1996), several datasets have been proposed. In the remainder of this section, we will briefly outline the most popular resources used for the NER task.

Gold-Standard Resources. The CoNLL-2002 and 2003 datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003a)[1] were created in four different languages (Spanish, Dutch, English, and German) from newswire articles and focused on 4 entity types: PER (Person), ORG (Organization), LOC (Location), and MISC (Miscellaneous, i.e., all other entity types). Several other NER shared tasks were organized in the years which followed, covering further languages such as Indic (Rajeev Sangal and Singh, 2008) and Balto-Slavic languages (Piskorski et al., 2017), or domains (Balasuriya et al., 2009; Ritter et al., 2011b).

[1]Since prior works found significant numbers of annotation errors, incompleteness, and inconsistencies in the CoNLL-2003 dataset, Rücker and Akbik (2023) introduced CleanCoNLL.

For instance, Balasuriya et al. (2009) claimed that NER was needed in many domains beyond newswire texts, and introduced WikiGold, a manually-annotated dataset derived from Wikipedia articles, while (Ritter et al., 2011b) introduced a dataset for English tweets using 10 NER classes. However, these datasets were limited in size and considered only the English language.

Another notable resource was proposed by Weischedel et al. (2013), who introduced OntoNotes 5.0. This dataset covered several categories, multiple genres (e.g. newswire and weblogs), and multiple languages (English, Chinese, and Arabic). Thanks to its high quality, it is one of the most widely used datasets for training and benchmarking NER systems.

Finally, another notable dataset was proposed for the WNUT 2017 shared task on emerging and rare entities (Derczynski et al., 2017), covering different textual genres (tweets, YouTube comments, Reddit and StackExchange posts). However, only the English language and 6 categories were considered.

Silver-Standard Resources. Although the above datasets constitute a valuable resource for training and evaluating NER systems, their applicability is limited by their size and the few languages they cover. Indeed, in the last decade, with the aim of reducing time and costs associated with the creation of labeled data for NER, and consequently scaling NER to a wider set of languages, several strategies have been proposed.

For instance, Iovine et al. (2022) proposed an unsupervised approach based on cycle-consistency training, which requires only pairs of unlabeled sentences and a small independent sample of named entities to generate training data for NER across diverse languages and domains. This method significantly reduces the costs associated with creating NER datasets, offering a cost-effective alternative to the labor-intensive manual annotation processes needed for traditional supervised datasets. However, the performance of systems trained with this approach still lags behind that of supervised systems, reaching approximately 70% of state-of-the-art performance.

Other works, instead, focused on automatic data creation methodologies. The first successful attempt in this direction was made by Nothman et al. (2013) who produced the WikiNER dataset. They proposed a strategy for automatically creating multilingual training data for NER by exploiting the texts of Wikipedia and its hypertext organization. In addition,

they also used redirect-base heuristics to infer more named-entity mentions. Adopting a similar strategy, Pan et al. (2017) introduced WikiANN, a language-independent framework for extracting entities from documents. Their procedure was made up of two main steps: i) classify entries in the English Wikipedia into specific entity types, and ii) propagate the annotations to other languages by applying cross-lingual transfer. This procedure yielded massive corpora consisting of 282 languages, but with lower annotation quality and a focus on persons, locations, organizations and geo-political entities. Other works relied on Freebase (Bollacker et al., 2008), a sizeable collaborative graph database, either by using its association to English Wikipedia as a training feature (Tsai et al., 2016), or by mapping its attributes to entity types, in order to identify the NER classes (Al-Rfou et al., 2015). Moreover, Al-Rfou et al. (2015) also tried to overcome the issue of missing annotations of non-anchored mentions in Wikipedia by using a simple surface string-matching heuristic, and resampled the datasets they produced to reduce the high class imbalance.

Despite covering many languages, these methods still tend to produce noisy annotations. Additionally, many of them strongly rely on overly-specific heuristics for propagating tags and overlook the benefits that auxiliary neural methods could provide in improving both the precision and recall of the annotations.

2.1.4 Challenges and Limitations

As previously mentioned, NER systems have made significant strides, but they still face notable limitations, particularly in multilingual settings. One major challenge, that affects also other IE tasks (cf. Section 1.2), is the limited availability of high-quality datasets for most languages as these datasets are expensive and time-consuming to collect. Indeed, gold-standard datasets typically focus on the English language (and a few other high-resource languages), are limited in size, and are designed for benchmarking rather than large-scale training, making them insufficient for real-world applications where more extensive and diverse data are needed. The narrow domain focus of these datasets further limits their utility across different contexts.

On the contrary, silver-standard datasets attempt to fill the data gap for low-resource languages but come with their own set of problems: they often suffer from noisy, inconsistent annotations and domain-specific biases. As a result, high-quality multilingual datasets are

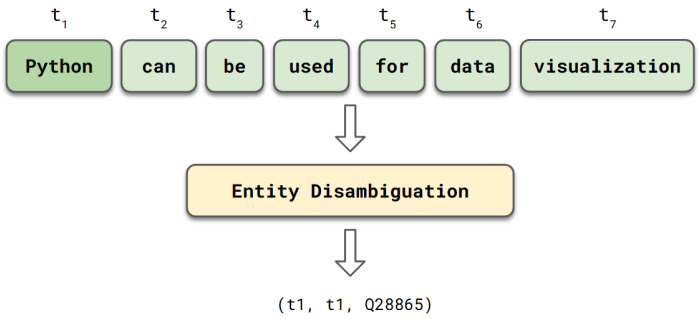

Figure 2.2. Given a sequence of tokens, Entity Disambiguation (ED) assigns a unique identifier (e.g. a Wikidata ID) to the pre-identified entity mentions.

needed to produce robust and efficient multilingual NER systems that can effectively identify and classifying named entities across several languages.

As an alternative solution, one might suggest using Large Language models (LLMs), such as GPT-4 (OpenAI et al., 2024), LLaMa (Touvron et al., 2023) or Mistral (Jiang et al., 2023) to perform NER in zero-shot or few-shot settings. However, the performance of these models in multilingual environments is poor as they are mainly pretrained with English data scraped from the web, hence overlooking low-resource languages that remain largely underrepresented in their training corpora. Additionally, these models are designed to be general-purpose task solvers, with this generality coming at the expense of specialization (Kocoń et al., 2023). Indeed, they often overgeneralize, missing the fine-grained distinctions needed for precise entity recognition, especially when dealing with ambiguous, rare, or domain-specific entities. Last but not least, even if they can achieve state-of-the-art performance after fine-tuning, LLMs are resource-intensive, making them impractical for real-time applications or for processing gigabytes or terabytes of data.

2.2 Entity Disambiguation

2.2.1 Task Formulation

Entity Disambiguation (ED), also known as Named Entity Disambiguation (NED) or Entity Linking (EL), is another core task in NLP. Unlike NER, which focuses on identifying and classifying named entities within text, ED involves linking these identified entities to specific, unique entries within a knowledge base such as Wikipedia, Wikidata, Freebase, or BabelNet

(Sevgili et al., 2022). This task is particularly important in real-world scenarios, as distinct entities often share the same or similar names – such as *Michael Jordan* which could refer to the *basketball player*, the *actor*, or the *researcher*, among others.[2] By grounding each mention in a structured knowledge base, ED enables systems to go beyond surface-level text, allowing access to additional, contextually relevant information. Specifically, this grounding is important for enriching downstream tasks such as QA, information retrieval, and recommendation systems, where accurate associations and factuality are crucial.

Formally, given a sequence of tokens $T = (t_1, t_2, \ldots, t_N)$ and a set of identified entity spans (s, e, ℓ) from NER – where s and e denote the start and end positions of an entity within the sequence, and ℓ denotes the entity type (cf. Section 2.1) – the ED task aims to map each entity span to a unique identifier I (e.g., a URI or an ID) in a predefined knowledge base $\mathcal{K}$. For example, given the sentence "Python can be used for data visualization", an entity disambiguation system must determine whether *Python* refers to the programming language commonly used for artificial intelligence and data analysis or to the snake species known for its large size, as illustrated in Figure 2.2.

2.2.2 Evaluation Metrics

ED requires linking ambiguous mentions to entities in a knowledge base (KB) and it is typically evaluated with accuracy and the inKB micro-F1 score metrics. Accuracy measures the proportion of correctly disambiguated mentions:

$$\text{Accuracy} = \frac{\text{Correctly Linked Entities}}{\text{Total Mentions}}$$

The inKB metric, instead, evaluates only those mentions that can be linked to valid entities within the KB, ignoring any mentions without valid KB entries. In other words, this score aggregates precision and recall across all mentions at the micro level, focusing on the system's accuracy in linking entities to the correct KB entries. In this evaluation setting, out-of-KB entities are treated separately to avoid skewing results.

[2] `https://en.wikipedia.org/wiki/Michael_Jordan_(disambiguation)`

2.2.3 Current Trends

The task of entity linking in its modern sense has been first introduced by Mihalcea and Csomai (2007). Since then, multiple evaluation datasets have been proposed, and numerous methods have been developed to improve the results. The first methods of entity disambiguation focused explicitly on two components: entity *commonness*, which shows how likely a particular entity is to be linked to a given mention regardless of the context, and entity *relatedness*, which shows how relevant an entity is to a given context (Milne and Witten, 2008). Three most prominent methods that used this approach are AIDA (Hoffart et al., 2011a), TagMe (Ferragina and Scaiella, 2010) and WAT (Piccinno and Ferragina, 2014). AIDA (Hoffart et al., 2011a) uses a graph-based collective disambiguation algorithm enhanced with robustness tests, building weighted edges between mentions and candidate entities as well as between different candidate entities, and removing an edge on every iteration to maximise the minimum weighted degree of the graph. TagMe (Ferragina and Scaiella, 2010), instead, uses the relatedness measure defined by Milne and Witten (2008) weighted with the commonness of a sense together with the keyphraseness measure defined by Mihalcea and Csomai (2007) to exploit the context around the target word. Finally, WAT (Piccinno and Ferragina, 2014) is a redesigned version of TagMe and includes graph-based algorithm for ranking entities in entity graph based on entity relatedness, and vote-based algorithm for local disambiguation. Another approach to entity disambiguation presented in the same year as WAT is Babelfy (Moro et al., 2014a), a method that combines entity linking with word sense disambiguation, using an algorithm based on random walks on the BabelNet multilingual graph to disambiguate all linkable mentions together with senses of words occurring in the text.

In the last few years, instead, neural approaches started to attain strong results in ED, especially thanks to the advances in contextualized word embedding and entity representation techniques (Ganea and Hofmann, 2017; Le and Titov, 2018, 2019; Yang et al., 2019). While initial work in this direction relied on static word embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) to represent mentions and entities, more recent studies (Shahbazi et al., 2019; Broscheit, 2019; Botha et al., 2020; Cao et al., 2021) have shown the benefit of employing contextualized embeddings from PLMs, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and BART (Lewis et al., 2020). For

example, Botha et al. (2020) put forward a dual-encoder architecture, composed of two separate encoders for mentions and entities, that maximizes the similarity between a mention embedding and its corresponding entity embedding, whereas Cao et al. (2021) proposed GENRE which, given a mention in context, generates its unique name (e.g. the title of its corresponding Wikipedia page) in an autoregressive fashion.

Finally, another promising body of work studies how to enrich neural models by taking advantage of relational knowledge from semantic networks. For instance, Raiman and Raiman (2018) proposed DeepType which relies on Wikidata to integrate symbolic knowledge into the reasoning process of a neural network. Similarly, Orr et al. (2020) introduced Bootleg, a system that uses the edges defined in Wikidata and YAGO to encode entity relations and entity types as input embeddings to a Transformer-based architecture.

2.2.4 Challenges and Limitations

Although state-of-the-art systems achieved impressive results on standard benchmarks, i.e. higher than 90% F1 score, in order to perform at their best, these systems require millions of training samples. For instance, GENRE is trained on KILT (Petroni et al., 2021) which is made up of 9M training instances – and this often means days-long training times. Therefore, a researcher with a limited hardware budget must decide between training on lower amounts of data at the cost of drastic drops in performance – the performance of GENRE drops by 8.8 points in F1 when trained only on the AIDA-YAGO-CoNLL training set – and long training times.

Additionally, while learning from large amounts of data allows the modern neural ED methods to achieve high performance on most of the evaluation datasets, it also puts them at risk of overfitting to the most frequently seen entities and overlooking the important edge cases. Ilievski et al. (2018) showed that entities with low frequency and high ambiguity appear to be underrepresented in standard ED evaluation datasets, despite such entities being especially challenging to disambiguate. Provatorova et al. (2021) continued this line of work, introducing the concept of entity overshadowing and releasing ShadowLink, the first dataset designed specifically to study this phenomenon. Both studies conclude that the problem of entity disambiguation is still far from solved, and call for ED systems to consider more complex cases instead of only optimizing for the standard evaluation frameworks.

Finally, while there is clear evidence that integrating relational knowledge into ED approaches is beneficial, the sparsity of the relations used by Raiman and Raiman (2018) and Orr et al. (2020) may make them an unappealing option for low-data scenarios.

In light of these critical concerns, in this thesis we show that the clever use of NER for ED can significantly narrow the gap between systems trained on thousands as opposed to millions of instances, while retaining the benefits of shorter training times and enhancing disambiguation performance on less common entities. Indeed, thanks to its coarse-grained classes, NER is an obvious way to cluster entities and, therefore, to reduce the intrinsic sparsity of the ED task.

2.3 Relation Extraction

2.3.1 Task Formulation

Relation Extraction (RE) is a key IE task that focuses on identifying and classifying semantic relationships between entities within a text (Pawar et al., 2017b). This task is essential for building structured representations of unstructured data, where entities – such as people, organizations, locations, and dates – are connected through meaningful relations, thus enabling deeper knowledge representation. For instance, in biomedical texts, RE can capture relationships between genes and diseases, while in general texts, it can extract connections like a person's affiliation with an organization or the location of an event.

Formally, given a sequence of tokens $T = (t_1, t_2, \ldots, t_N)$ and a set of detected entities, the task of RE is to produce a set of triples (e_1, e_2, r), where e_1 and e_2 are entities within the text, and r denotes the semantic relationship between them, chosen from a predefined set of relation types. For example, in the sentence "Roger Federer has been awarded an honorary doctorate by the University of Basel" an RE system would identify a relation like `award received` (P166 in Wikidata) between *Roger Federer* and the *honorary doctorate at University of Basel* as illustrated in Figure 2.3.

2.3.2 Evaluation Metrics

RE systems are evaluated based on relational triples, which consist of two entities and the relationship between them. The evaluation metrics commonly used are precision, recall,

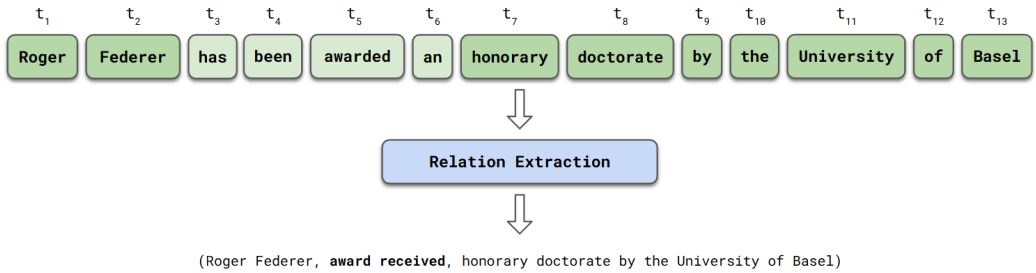

Figure 2.3. Given a sequence of tokens together with pre-identified entities, a Relation Extraction (RE) system predicts relationships between pairs of entities.

and the micro-F1 score. A relational triple is considered correct only if both entities and the relationship type match the ground truth exactly. However, with the emergence of joint models that simultaneously address entity recognition and relation classification, various evaluation metrics have been introduced to accommodate different scenarios. Specifically, Bekoulis et al. (2018) proposed the following three distinct evaluation settings:

- In the *Strict* setting, both the entity boundaries and the relationship type must match the ground truth precisely;
- The *Boundaries* setting relaxes one requirement by allowing for correct entity boundaries while disregarding the need for an exact match on the entity types;
- The *Relaxed* setting further eases the criteria by permitting partial matches of entities, such as subtokens.

It is important to note that regardless of the evaluation setting used, the relationship type must always be accurately identified for a relational triple to be considered correct.

2.3.3 Current Trends

In relation extraction, the goal is to identify all triplets, composed of a subject, an object, and a relation between them, within a given text. Early approaches to RE split the task into two different sub-tasks: Named Entity Recognition (NER) (Nadeau and Sekine, 2007b), which identifies all entities, and Relation Classification (Bassignana and Plank, 2022), which classifies the relationship, or lack thereof, between them. In this pipeline approach, NER is used to identify all entities in the text, and then, two entities are given to the RC system which classifies the relationship between them. However, errors from the NER system may

be propagated to the subsequent module, leaving the shared information in the interaction of both tasks unexplored.

To address this issue, recent works have tackled RE in an end-to-end fashion, seeking to overcome these problems by using different abstractions of the task. Miwa and Sasaki (2014) introduced a table representation and reframed RE as a table-filling task. This idea was further explored and extended by Pawar et al. (2017a) and Wang and Lu (2020). However, these systems still had some restrictions, such as assuming that only one relation exists between each pair. Instead by framing the task as a sequence of triplets to be decoded, seq2seq approaches (Paolini et al., 2021; Huguet Cabot and Navigli, 2021) provided more flexibility to the RE task and lifted some of these restrictions.

2.3.4 Challenges and Limitations

Recent language models, especially seq2seq models, are notoriously data-hungry, hence vast amounts of data are needed to enable them to learn the task with satisfactory scores. However, manually annotating RE data is a costly and time-consuming process, thus leading to a lack of training data, especially in multilingual settings, as for NER (cf. Section 2.1.4).

To fill this gap, many RE datasets have been created using distant supervision methods, such as NYT (Riedel et al., 2010), T-REx (Elsahar et al., 2018) or DocRED (Yao et al., 2019). Despite their widespread use in the RE community, these datasets have limitations. For instance, automatically generated datasets often contain noisy labels, leading to unfair or misleading evaluations. Additionally, there has been a long-standing focus on monolingual relation extraction systems, particularly in English. The ACE05[3] benchmark presented some of the first relation extraction datasets in three languages, Arabic, Chinese, and English. However, ACE05 covers only six relation types, and requires a paid license for its use. Additionally, the focus on Arabic and Chinese quickly faded away while resources for English continued to grow. Subsequently, The SMiLER dataset (Seganti et al., 2021) has been introduced. Despite being still based on distant supervision, it uses Wikipedia and Wikidata to create a large multilingual relation extraction dataset covering a broader set of RE types. However, besides being automatic, it does not contain human-annotated samples that permit reliable evaluation and limits annotations to one triple per sentence leading to

[3] https://catalog.ldc.upenn.edu/LDC2006T06

sparse annotations not suitable for training end-to-end RE systems.

In this thesis, we aim at overcoming the limitations of existing datasets by providing a new multilingual dataset that includes manual annotations and enables RE with a wide coverage and higher quality despite being based on automatic annotation.

2.4 Other Tasks

The IE tasks described so far, such as NER (Section 2.1), ED (Section 2.2), and RE (Section 2.3), are essential for structuring and interpreting unstructured data. In addition to these, coreference resolution (Sukthanker et al., 2020) can identify when different expressions in a text refer to the same entity, thereby helping to maintain coherence and clarity in the analysis. Furthermore, event extraction (Xiang and Wang, 2019) involves identifying actions or events described in the text, as well as the participants and attributes associated with those events, thus allowing for a deeper comprehension of the context in which entities operate and interact.

However, while these techniques excel at identifying specific entities, linking them to knowledge bases, and finding relationships between them, they fall short of capturing the full spectrum of semantic nuances present in language. Indeed, natural language is rich with other kinds of nominal expressions as well as figurative expressions that often extend beyond the literal interpretation of words. Consequently, relying solely on entity-centric modules can lead to an incomplete understanding of the text, as subtleties and layers of meaning may be overlooked. Therefore, to fully grasp the complexity of human language, it is essential to incorporate additional analytical frameworks that address these nuances, enabling more comprehensive information extraction and interpretation. To fill this gap, as anticipated in Section 1.4, in this thesis we study concepts and figurative language (with a particular focus on idiomatic expressions), and propose novel task formulations that enable the seamless integration of these new modules in a standard IE pipeline.

2.4.1 Concept Recognition

In the literature, when looking for a task aimed at identifying and classifying concepts, there is no direct counterpart for concepts to what NER is for named entities. Indeed, the

vast majority of works that focus on concepts exclusively deal with their disambiguation. Specifically, the closest task to what we have called Concept Recognition (CR) is Word Sense Disambiguation (Navigli, 2009, WSD), the main difference being that in WSD a word is tagged with one of the senses it can denote in a given inventory – e.g. WordNet (Miller, 1995) – whereas in CR a word needs to be tagged with a more general category that is independent of the word itself.

However, assigning fine-grained senses to words has been found to be extremely difficult (Izquierdo et al., 2015; Maru et al., 2022), which has led to coarse-grained disambiguation approaches aimed at reducing the number of categories to choose from. Vial et al. (2019) leverage hypernymy relations to reduce WordNet granularity by exploiting its taxonomy graph, releasing a coarser-grained inventory with 39K labels. Similarly, Proietti et al. (2024) exploited homonymy relationships to cluster word senses. Despite the efforts, the still high number of proposed categories is too specific to be considered as a good candidate categorization for CR.[4] Izquierdo Beviá et al. (2007), instead, exploit WordNet relations to automatically extract a set of fundamental senses, called Basic Level Concepts, to which all other senses are mapped. Nevertheless, depending on a threshold, this approach outputs hundreds or thousands of Basic Level Concepts, which are again too fine-grained. WordNet lexicographer files, instead, organize WordNet synsets into 45 lexicographer files[5] based on syntactic categories and logical groupings. Here, each category contains synsets with the same PoS tag and a general semantic type (e.g. NOUN.FEELING). However, to the best of our knowledge, no works trained neural systems to classify concepts based on this categorization. Additionally, despite WordNet lexicographer files can be used as coarse categories for nominal concepts, we argue that a unified formulation with NER could bring synergistic benefits to the two tasks.

For this reason, we move a step further and explore comprehensive categorizations for NER. Notably, while initial datasets were primarily constructed to classify named entities into a limited set of categories, namely PER, ORG, LOC and MISC (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003a), recent approaches have proposed unwrapping the MISC category – containing miscellaneous entities – into fine-grained categories to explicitly

[4] Choi et al. (2018) showed that with 10k+ categories a state-of-the-art model struggles to obtain reasonable classification accuracy.

[5] https://wordnet.princeton.edu/documentation/lexnames5wn

identify more specific entity types. Among these, OntoNotes stands out as a well-established resource that identifies 18 categories for named entities (Pradhan et al., 2012a). However, its annotations do not cover instances of several classes, including food, animals, or celestial bodies, among others. This was instead addressed in subsequent works, i.e. Few-NERD (Ding et al., 2021) and MultiCoNER II (Fetahu et al., 2023), which proposed named entity categorizations of 66 and 36 categories, respectively. Although these categorizations are more comprehensive and fine-grained, they tend to focus on excessively detailed semantic types for our purpose. For instance, the category ISLAND is well-suited for classifying named entities like *Lesbos*, but it proves less suitable for nominal concepts, as only a few, such as *islet*, fall within its scope. Other examples of categories that are not suitable for the classification of concepts are LIBRARY, AIRPORT and HOTEL, *inter alia*.

Even more related to our objective, a categorization that spans both concepts and entities has been proposed in the context of the ultra-fine entity typing[6] task (Choi et al., 2018, UFET). Here, the goal is to produce a three-level annotation by selecting tags from coarse, fine, and ultra-fine sets of labels consisting of 9, 121 and $\sim$10K types, respectively. Nevertheless, the extreme granularity of the ultra-fine labels makes it difficult for systems to output the right categories, as reported by the authors, while the coarse-grained types are too generic to be considered a good candidate for our fine-grained objective. The fine set of categories comprising 121 tags, instead, not only contains categories that are too specific (e.g. DOCTOR, COACH), but also fails to categorize commonly used concepts, such as relations (e.g. *inferiority*, *brotherhood*), date/times (e.g. *November*, *September 3rd*), feelings (e.g. *love*, *anger*), and psychological features (e.g. *cognition*, *thought*), *inter alia*. Finally, we remark that the proposed ultra-fine entity typing task disregards the identification part. This is in marked contrast to CR and NER, where each and every span of text denoting concepts or entities has to be identified and tagged.

Ultimately, concerning concepts, with WordNet lexicographer files having strong psycholinguistic grounding, we deem them as highly reliable and use them as a starting point for our CR and NER joint categorization. Similarly, we believe that OntoNotes strikes a good balance in terms of the level of granularity of its categories and blends well with a complementary set of abstract semantic types needed for concepts, which includes

[6]Entity Typing is often referred to as the second step of NER, aiming at classifying pre-identified entity mentions.

relations, properties, and feelings, among others. Therefore, based on these resources (as later described in Section 6.1.2), in this thesis, we aim to introduce a new comprehensive categorization that enables IE systems to produce denser and richer semantic annotations by identifying and classifying both concepts and named entities in a joint fashion.

2.4.2 Idiomatic Expression Identification

Besides named entities and concepts, over the course of the last few years, several attempts have been made to classify other kinds of Multiword Expressions (MWEs) based on specific dimensions such as polylexicality, fixedness, compositionality, and idiomaticity (Sailer and Markantonatou, 2018). Among MWEs, idioms are of particular interest in that their meaning cannot be obtained by compositionally interpreting their word constituents. These include non-compositional phrases, e.g. *kick the bucket*, and partially-compositional phrases, e.g. *rain cats and dogs* (Nunberg et al., 1994). Given their complex nature, idioms are hard to be automatically identified and pose a crucial challenge to the entire field of Natural Language Understanding (NLU). Indeed, several approaches have been put forward to address the idiom identification task. To this end, two main properties of idioms have been leveraged, namely their syntactic and semantic idiosyncrasies. While the former refers to the peculiar syntactic behaviour of idioms, the latter indicates the linguistic property in which the meaning of an idiomatic expression cannot be completely derived from the meaning of its individual components.

Initial studies regarding idiom identification focused on syntactic idiosyncrasy, concentrating on verb/noun idioms, e.g. *shoot the breeze* (Fazly and Stevenson, 2006; Cook et al., 2007; Diab and Bhutada, 2009), on verb/particle idioms, e.g. *call off* (Ramisch et al., 2008) or on idioms satisfying specific restrictions, i.e. subject/verb such as *tension mounted* and verb/direct-object, e.g. *break the ice* (Shutova et al., 2010).

Subsequent approaches exploited semantic idiosyncrasies. This property implies that idiomatic expressions often occur in contexts typically unrelated to the meaning of their individual constituents, thus providing a key feature to be exploited in an automatic approach. In particular, Muzny and Zettlemoyer (2013) introduced new lexical and graph-based features that use WordNet[7] and Wiktionary[8], and proposed a simple yet efficient binary Perceptron

[7] https://wordnet.princeton.edu/
[8] https://www.wiktionary.org/

classifier to distinguish between idiomatic and non-idiomatic expressions by exploiting their components and dictionary definitions. A similar, but unsupervised approach was adopted by Verma and Vuppuluri (2015) which relied on the dictionary definitions of each component of a given idiom. These latter methods have more recently been superseded by approaches making use of distributional similarity in the form of both static and contextualized word embeddings (Gharbieh et al., 2016; Ehren, 2017; Senaldi et al., 2019; Hashempour and Villavicencio, 2020; Fakharian, 2021; Garcia et al., 2021; Nedumpozhimana and Kelleher, 2021), while keeping the underlying assumption unchanged: the vector representation of the component words should be distant from the vector representation of the context or of the expression as a whole.

Notwithstanding the recent improvements, most existing datasets are limited in size. For instance, Cook et al. (2008) and Sporleder et al. (2010) manually selected a small number of idioms, and then extracted sentences containing such idioms from the British National Corpus (BNC, Consortium et al., 2007). Similarly, Sporleder and Li (2009) extracted a dataset from the Gigaword corpus (Graff and Christopher, 2003). Street et al. (2010), instead, used multiple annotators to validate sentences from the American National Corpus (ANC, Ide and Macleod, 2001). Furthermore, Korkontzelos et al. (2013) introduced Task 5b at SemEval-2013 regarding the detection of semantic compositionality in context. The authors selected idioms from Wiktionary, and extracted instances from the ukWaC corpus (Ferraresi et al., 2008). Schneider et al. (2016), instead, proposed the DiMSUM dataset for Task 10 at SemEval-2016, and extracted annotations from reviews, tweets and TED talks. However, this work did not categorise MWEs into subtypes, making it difficult to quantify the number of idioms in the corpus. Gong et al. (2017), instead, introduced a small-scale dataset derived from Google Books[9] for English and Chinese. Finally, Madabushi et al. (2021) proposed a SemEval-2022 task on idiom identification. Nevertheless, their datasets are limited in size and they only cover three languages, namely English, Portuguese and Galician.

Notwithstanding the recent improvements, to the best of our knowledge, the identification of idiomatic expressions in multiple languages is still under-investigated.

[9] https://books.google.com/

Chapter 3

Multilingual Named Entity Recognition

We start our journey towards efficient and comprehensive information extraction across languages by tackling the existing challenges in NER. Specifically, as described in Section 2.1.4, NER datasets are available for a narrow set of languages, strongly limiting the applicability of NER systems in real-world scenarios. Additionally, many of these datasets are too general in nature and focus primarily on coarse-grained categories, further hampering the effectiveness of the resulting systems when applied to specialized or complex tasks.

In this chapter, aiming at filling these gaps, we first introduce a novel automated approach to producing multilingual training corpora for NER (Section 3.1). Then, we present a new strategy that, combined with our automated approach, enables the construction of NER datasets tailored to specific domains (e.g. sports, politics, or finance) further enhancing the performance of NER systems in specialized settings (Section 3.2). Finally, we design a new comprehensive set of categories for NER and expand our study by producing fine-grained data for the task (Section 3.3).

3.1 Silver-Data Creation

The production of high-quality multilingual data for NER is essential to enable the widespread and effective deployment of NER systems on a global scale. However, generating such data presents considerable challenges, particularly in a multilingual context where the in-

volvement of expert annotators from a wide range of linguistic and cultural backgrounds is required. Additionally, given the nature of the task, where individual tokens must be carefully annotated, the time and costs of producing multilingual NER data are further amplified.

Motivated by these challenges, here we introduce WikiNEuRal, our novel automated methodology for generating high-quality multilingual NER data. The key novelty of WikiNEuRal resides in the effective combination of neural and knowledge-based approaches. Specifically, it takes as input a small set of entity annotations and proceeds by tagging all the entities in Wikipedia following the semantic relations in the underlying knowledge graph. Then, these annotations are processed by an advanced neural model that further increases the annotation quality. We leverage Wikipedia as our primary data source due to its vast, diverse, and manually-curated content. Moreover, the multilingual nature of Wikipedia enables the creation of datasets that support a wide range of languages, further enhancing the scalability and applicability of our approach. Therefore, by using Wikipedia, we ensure that our NER data is both domain-rich and linguistically diverse, allowing for more robust and versatile NER models.

In the remainder of this section, we describe each step involved in the WikiNEuRal annotation pipeline (Figure 3.1), and then proceed by validating the quality of the produced annotations by comparing the WikiNEuRal data against the data produced with other automated data-creation methodologies as well as manually-created datasets (Sections 3.1.6 and 3.1.7).

3.1.1 Preprocessing Wikipedia

In order to obtain clear and informative texts, we start by cleaning up the text of Wikipedia articles by removing the sections with the 10 most frequent titles, which usually list related resources (e.g., bibliography, references, see also). We also remove all other elements that tend to introduce noise, such as lists, tables, templates, formulas, images, infoboxes, footnotes, etc.

After this process, besides the raw text, the only remaining elements are *wikilinks*,[1] which provide potential entity mentions. These links typically show up either with the

[1] A wikilink (or internal link) is a link from one page to another page within the English Wikipedia, or, more generally, within the same Wikipedia (e.g. within the French Wikipedia).

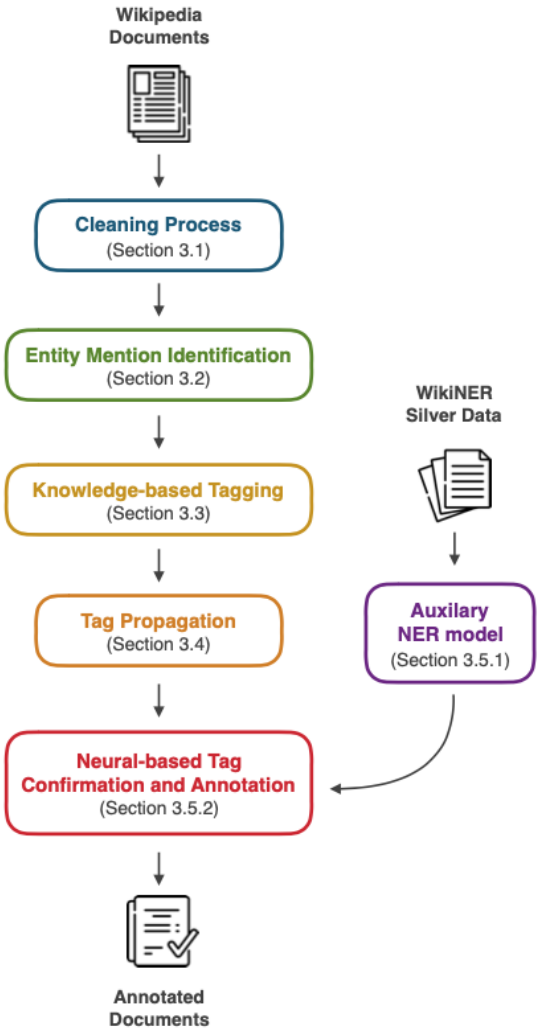

Figure 3.1. WikiNEuRal annotation pipeline.

links-only format (e.g., `[[apartment]]`), where the link and the displayed surface form are identical, or with both a link and a surface form (e.g., `[[apartment|flats]]`). We opt to discard all occurrences of this latter case because such wikilinks might introduce errors (e.g., in `[[Royal Dutch Shell|oil industry company]]`, the surface text *oil industry company*, that do not correspond to a named entity, would be tagged as ORG).

3.1.2 Identifying Entity Mentions in Wikipedia

Although wikilinks offer valuable information, not all links point to a named entity (e.g., see `[[apartment]]` above, which denotes a concept, instead). We therefore exploit the

one-to-one correspondence between Wikipedia articles and BabelNet synsets to classify each synset into either an abstract concept (C) or a named entity (NE), which we will refer to as its *type*. Although BabelNet provides this information for most of its synsets, upon manual inspection we find that many of these *type annotations* are noisy[2], so we opt to train a model to perform binary classification, effectively replacing the annotations provided by BabelNet.

Specifically, the dataset for this task is constructed exploiting the multilingual nature of BabelNet: given a synset s and a language L, we have access to a set of glosses $G_L(s)$. For every such gloss $g \in G_L(s)$ we generate the following training sample $I(s, g, L) =$ [CLS] $l_L(s) \mid g$ [SEP] with the type as label, where $l_L(s)$ is the main lemma for synset s in language L or, if it does not exist, a special token [NOLEMMA] shared among languages. We rank BabelNet synsets according to the number of glosses they contain and take the top 500k synsets to build our training and validation splits (450k and 50k respectively).

We then use this dataset to train a simple yet efficient Sequence Classification architecture (Devlin et al., 2019) with Multilingual BERT as encoder and a linear layer to perform binary classification on top of the [CLS] token. As this system represents only one step of the entire WikiNEuRal pipeline, we measure its quality in vivo (see Section 3.1.8), as a test set generated with the same distribution would yield inconclusive results.

Once trained, we use the model to tag each BabelNet synset (and therefore each Wikipedia article) as either a concept or a named entity. In particular, in order to classify any particular synset, we gather all its possible inputs generated by the function I (i.e., we take into account all of its glosses in the considered languages $\hat{L}$[3]) $I(s) = \{I(s, L, g) | L \in \hat{L},\ g \in G_L(s)\}$ and label them independently using the trained model. Finally, we select, for synset s, the label with the maximum average confidence[4] – with this label becoming the new *type* for synset s and, therefore, for its linked Wikipedia page, regardless of the language.

Concerning our pipeline, we use these new labels to discard links to Wikipedia articles corresponding to concept types, as we are only interested in named entities.

[2]Based on the majority case of the title occurrence within its Wikipedia article.

[3]We only take glosses in $\hat{L} = \{$English, Italian, German, Spanish, French$\}$, as they are the best supported languages between BabelNet and Multilingual BERT.

[4]We ignore predictions for which the averaged confidence falls below 0.6.

3.1.3 Tagging Named Entity Links Through Synsets

After identifying all the wikilinks corresponding to named entities in a given Wikipedia article, the next step involves tagging each wikilink with a category $c \in C$, where C is the set of the common NER classes: $C = \{\text{PER}, \text{ORG}, \text{LOC}, \text{MISC}\}$.

To achieve this, we create a mapping between Wikipedia articles and their respective NER classes that is subsequently used in our pipeline to annotate individual wikilinks. To construct this mapping, we introduce a novel *semantic classifier* that exploits the one-to-one correspondence between Wikipedia pages and BabelNet synsets. Specifically, we start by manually annotating 200 synsets to cover as many high-order concepts of the WordNet[5] (Miller, 1995) nominal taxonomy – which is a subset of the BabelNet taxonomy – as possible. For instance, we label the following high-level synsets as follows:

- `human being (bn:00044576n)` → PER;
- `company (bn:00021286n)` → ORG;
- `association (bn:00006539n)` → ORG;
- `community (bn:00021248n)` → ORG;
- `town (bn:00077773n)` → LOC;
- `body of water (bn:00011766n)` → LOC;
- `country (bn:00023235n)` → LOC;
- `food (bn:00035650n)` → MISC;
- `work of art (bn:00081581n)` → MISC;
- `event (bn:02131709n)` → MISC;

Then, we proceed by propagating these annotations to all other synsets in WordNet (i.e. the high-quality subset of BabelNet with manually-curated relations) by descending through its taxonomy. More technically, we follow *hyponymy* and *has-instance* relationships (i.e. parent-to-child relations) yielding 40k high-quality annotations. For example, all the

[5]We start from WordNet synsets because they are manually curated.

children of `company (bn:00004222n)`, e.g. `Amazon (bn:03796855n)`, `Apple Inc.(bn:03739345n)`, and `Google (bn:00212508n)` inherit the ORG tag.

Finally, to annotate all the remaining BabelNet synsets, we again traverse the BabelNet taxonomy but this time following hypernymy relationships (child-to-parent relationships) until one of the 40k high-order synsets in the expanded set is reached. The annotation of the first hypernymous synset reached through this Breadth-First Search (BFS) traversal with a max depth equal to 2 is inherited (if any).

This procedure yields a classification of more than 7.5 million synsets corresponding to named entities, and we use the resulting mapping to tag each wikilink in a given Wikipedia article. For clarity, we inform the reader that the mapping is constructed only once and subsequently accessed by the pipeline to tag named entities.

3.1.4 Named Entity Tag Propagation

As a result of the above steps, for each Wikipedia article we know which anchored strings (i.e. wikilinks) correspond to named entities and which NER classes they belong to. However, one of the major drawbacks of using Wikipedia texts is that only small portions of text are anchored. This includes the fact that, according to Wikipedia guidelines, only the first mention to the article has to be linked. For example, given the following portion of text:

> Steve Jobs was the CEO and co-founder of Apple Inc.. *Steve Jobs* is widely recognized as a pioneer of the personal computer revolution of the 1970s and 1980s, along with *Apple* co-founder Steve Wozniak.

we notice that "Steve Jobs" is anchored only the first time it appears, and that, the second time "Apple Inc." is referred to, "Inc." is missing.

To address this issue, since every anchored string a corresponds to a BabelNet synset s, we gather the synonyms in s (including the anchored text itself) in the language of the article, and employ an exact matching heuristic to propagate the named entity tag of a to any occurrence of synonyms of s throughout the whole article. For instance, "Apple" is among the synonyms of "Apple Inc.", hence all its occurrences within a text in which the latter link occurs are tagged as ORG, leading to denser annotations.

3.1.5 Confirming and Augmenting Annotations

We are now left with two potential weaknesses: first, even though aiming for high precision, our annotations might still include some mistakes; second, our tagged sentences might still contain unannotated entities due to unlinked or unmatched entity mentions. To address both issues, we introduce an NER classifier in the pipeline, and apply it to both confirm our annotations and augment the sentences with additional tags.

Our Auxiliary NER model

Our NER model is a variant of the BERT-based neural model of Mueller et al. (2020): following recent literature, rather than representing a word with the first contextualized subword representation as provided by multilingual BERT, we take the mean of its subwords (Ács et al., 2021). The resulting vectors are passed through a multi-layer sentence-level BiLSTM network, whose logits are then fed into a CRF model (Lafferty et al., 2001), trained to maximize the log-likelihood of the span-based gold label sequences.

Improving Precision and Recall

To address the above-mentioned weaknesses, we train our NER model with the WikiNER dataset (Nothman et al., 2013) and use it to confirm and augment our annotations. More formally, for every sentence x composed of n tokens $x_1, \ldots, x_n$, we compare the annotation (i.e., a named entity tag) y_i produced by our approach for each token x_i with the one produced by the neural model $\hat{y}_i$, and keep the sentence if i) there is at least one annotation $y_i \neq O$, and ii) every $y_i \neq O$ has the same annotation of the corresponding $\hat{y}_i$. This results in an improved precision of our annotations, as they are confirmed through an ensemble approach. Finally, we output as annotations for sentence x the labels produced by the neural model $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_n]$, therefore accounting for previously undiscovered entities and improving recall.

3.1.6 Experimental Setup

In the previous sections, we described our methodology for automatically creating training data for multilingual NER. In this section, we aim to assess the accuracy of the proposed

methodology compared to existing alternatives.

Architecture. For our experiments, we use the same NER architecture introduced in Section 3.1.5. We train this model on both the dataset produced with our methodology, the datasets created using existing methodologies, and popular gold-standard datasets. Each model variant is trained with early stopping set with a patience parameter of 10; we use Adam (Kingma and Ba, 2015a) optimizer with a learning rate fixed at 10^{-3} and a cross-entropy loss criterion. The full list of hyperparameter values is shown in Table 6.8. We repeat each training on 10 different seeds, fixed across experiments, and report the mean and standard deviation of their span F1 score; we compare experiments by means of Student's t-test (Student, 1908).

We highlight that the experiment aims to understand the impact of the dataset used for training the NER model (ours vs. existing silver- and gold-standard datasets), hence the architecture, hardware, and training parameters are kept identical across runs.

Training Data. We train our architecture with four different silver-standard datasets:

- **WikiNEuRal**: the dataset created using our methodology described in Section 3.1 from Wikipedia. It covers 9 languages: Dutch, English, French, German, Italian, Polish, Portuguese, Russian and Spanish. Data statistics are shown in Table 3.2.
- **WikiNER** (Nothman et al., 2013): the current best-performing approach for NER silver data creation. It covers the same languages as WikiNEuRal.
- **WikiANN**[6] (Pan et al., 2017): a multilingual NER dataset consisting of Wikipedia articles annotated in 282 languages.

We also train our reference model for every available, manually-annotated gold-standard training set from the **CoNLL-2002** NER Shared Task (Tjong Kim Sang, 2002) for Spanish and Dutch, the **CoNLL-2003** NER Shared Task (Tjong Kim Sang and De Meulder, 2003a) for English and German, and the **OntoNotes 5.0** dataset for English. All silver-

[6]The version used corresponds to the balanced train, dev, and test splits of Rahimi et al. (2019), which supports 176 of the 282 languages from the original WikiANN corpus, available at `https://huggingface.co/datasets/wikiann`.

Hyperparameter name	Value
number of Bi-LSTM layers	2
LSTM hidden size	512
batch size	128
learning rate	0.001
dropout	0.5
gradient clipping	1.0
adam β_1	0.9
adam β_2	0.999
adam ϵ	1e-8

Table 3.1. Hyperparameter values of the reference model used for our experiments.

and gold-standard datasets are tagged with the four standard entity types (PER, ORG, LOC, MISC), except for WikiANN which does not contain the MISC label.

All model training is carried out on an NVIDIA GeForce RTX 3090 and 2080 Ti architecture. It required $\sim$45 s/epoch on the CoNLL and WikiANN datasets, whereas it required $\sim$6 min/epoch on the WikiNER and WikiNEuRal datasets.

Test Data. We use five different test sets in our experiments:

- **CoNLL-2002** NER Shared Task dataset (Tjong Kim Sang, 2002): a popular collection of newswire articles for Spanish and Dutch.
- **CoNLL-2003** NER Shared Task dataset (Tjong Kim Sang and De Meulder, 2003a): a well-known collection of newswire articles for English and German taken from the Reuters Corpus and the ECI Multilingual Text Corpus, respectively.
- **WikiGold** (Balasuriya et al., 2009): a small set of English Wikipedia articles manually annotated with CoNLL named entity classes.
- **OntoNotes 5.0** (Pradhan et al., 2012b): this includes texts from five different text genres: broadcast conversation, broadcast news, magazine, newswire, and web data. We use it as an additional test set for English.
- **BSNLP-2017** (Piskorski et al., 2017): this consists of articles in various Slavic languages and we use it to evaluate Russian and Polish performances. Two test sets are provided: one contains articles about a specific politician, the other one about the European Commission.

Language	Articles	Sentences	Tokens	Avg. length	Avg. NEs	PER	ORG	LOC	MISC	O
English	50k	116k	2.73M	23.53	1.67	51k	31k	67k	45k	2.40M
Spanish	50k	95k	2.33M	24.46	1.61	43k	17k	68k	25k	2.04M
Dutch	65k	107k	1.91M	17.91	1.43	46k	22k	61k	24k	1.64M
German	50k	124k	2.19M	17.66	1.42	60k	32k	59k	25k	1.87M
Russian	105k	123k	2.39M	19.49	1.47	40k	26k	89k	25k	2.13M
Italian	50k	111k	2.99M	26.85	1.90	67k	22k	97k	26k	2.62M
French	50k	127k	3.24M	25.47	1.80	76k	25k	101k	29k	2.83M
Polish	105k	141k	2.29M	16.21	1.65	59k	34k	118k	22k	1.91M
Portuguese	80k	106k	2.53M	23.99	1.88	44k	17k	112k	25k	2.20M

Table 3.2. WikiNEuRal data statistics. "Avg. length" is the average sentence length and "Avg. NEs" is the average number of named entities per sentence.

All the datasets employ the CoNLL entity types (PER, ORG, LOC and MISC), except OntoNotes, which is annotated with 18 fine-grained entity types, which we manually map to the CoNLL tag set. For CoNLL, OntoNotes, and BSNLP, which are often used to benchmark NER, we take the official splits for validation and test sets. For WikiGold, which is much smaller, we use the full dataset as test material. Following the literature, we evaluate performances by means of the F1 score, i.e., the harmonic mean between Precision and Recall, using the official conlleval script. We convert all datasets to the popular BIO format.

3.1.7 Results

We assess the quality of the WikiNEuRal datasets extensively, comparing the performance obtained training the model presented in Section 3.1.5 both on WikiNEuRal and on the other silver datasets listed in Section 3.3.3. The results are reported in Tables 3.3 and 3.4. We observe consistent improvements of WikiNEuRal over the WikiNER and WikiANN alternatives on almost all tested languages and datasets. In particular, on the CoNLL test sets (Table 3.3), we notice a large average improvement, computed over the scores on the four languages, of 21.4 and 2.3 span-based F1 points over WikiANN and WikiNER, respectively. Three out of four results are also statistically significant. This improvement is even more marked if we look at the token-level F1 performance of the models trained on the various datasets. Morever, in the remaining test sets (i.e., WikiGold, OntoNotes and BSNLP), our approach achieves results that are again consistently better than the results of the two competitors, again in a statistically significant way (cf. Table 3.4).

Finally, in Table 3.5, we also show how the quality of the WikiNEuRal data compares with that of human-produced datasets. Specifically, models trained on our data perform

Training set \ Test set	CoNLL			
	English	Spanish	Dutch	German
WikiANN	56.85 ± 2.18	53.55 ± 3.10	55.76 ± 2.19	44.39 ± 0.95
WikiNER	73.05 ± 1.20	75.07 ± 0.96	74.75 ± 0.59	**64.03 ± 1.86**
WikiNEuRal (Ours)	**76.94 ± 0.75**	**77.87 ± 0.85**	**77.40 ± 0.57**	64.02 ± 0.54
WikiANN	63.90 ± 1.86	63.18 ± 2.80	64.32 ± 1.38	55.49 ± 0.97
WikiNER	71.15 ± 0.76	72.90 ± 0.97	75.90 ± 0.63	65.72 ± 1.03
WikiNEuRal (Ours)	**77.02 ± 0.66**	**77.98 ± 0.49**	**79.20 ± 0.40**	**67.50 ± 0.31**

Table 3.3. Span-based micro (top) and token-level macro (bottom) F1 scores on the CoNLL benchmark. Statistical significance is computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance scores are computed between a system and its next-best scoring competitor (e.g., WikiNEuRal vs WikiNER).

Training set \ Test set	WikiGold	OntoNotes	BSNLP	
	English	English	Russian	Polish
WikiANN	57.05 ± 2.76	36.43 ± 4.54	51.85 ± 1.80	53.50 ± 1.63
WikiNER	81.98 ± 0.28	71.16 ± 0.72	65.99 ± 0.94	62.31 ± 0.95
WikiNEuRal (Ours)	**82.42 ± 0.33****	**71.98 ± 0.55***	**66.50 ± 0.67**	**62.44 ± 1.00**
WikiANN	68.31 ± 2.14	53.58 ± 3.24	54.00 ± 1.57	60.55 ± 1.45
WikiNER	84.30 ± 0.29	74.83 ± 0.76	59.71 ± 1.20	64.87 ± 2.00
WikiNEuRal (Ours)	**84.69 ± 0.25***	**75.25 ± 0.44**	**61.08 ± 0.74***	**66.00 ± 1.33**

Table 3.4. Span-based micro (top) and token-level macro (bottom) F1 scores on the WikiGold, OntoNotes and BSNLP benchmark. Statistical significance is computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance scores are computed between a system and its next best scoring competitor (e.g., WikiNEuRal vs WikiNER).

8.2 F1 points better than CoNLL-trained models when evaluated on the WikiGold test set (in-domain distribution). Interestingly, models trained on our data also perform almost 1 point better when tested on a neutral test set, namely a corpus whose source and domain are different from both the WikiNEuRal and CoNLL ones (i.e., OntoNotes). Similarly, WikiNEuRal-based models perform 10.6 F1 points better than OntoNotes-trained models on the WikiGold test set, and almost 4 points better when tested on a neutral test set, i.e., CoNLL.

3.1.8 Ablation Study

In order to show the effectiveness of the steps detailed in Section 3.1, we disassembled our NER data creation pipeline. We conducted these experiments on the English WikiNEuRal

Training set \ Test set	CoNLL	WikiGold	OntoNotes
	English	English	English
WikiNEuRal (Ours)	76.94 ± 0.75	**82.42 ± 0.33**	71.98 ± 0.55
CoNLL	**90.07 ± 0.33**	74.22 ± 0.45	71.03 ± 0.47
OntoNotes	73.39 ± 0.60	71.59 ± 0.42	**89.39 ± 0.39**

Table 3.5. Span-based micro F1 scores comparing the quality of our silver-standard dataset against two well-known gold-standard datasets on the English Language, i.e. CoNLL and OntoNotes. Statistical significance is computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance scores are computed between a system and its next best scoring competitor.

corpus; results are shown in Table 6.12. We started by removing the *Concept vs. Named Entity* module of Section 3.1.2 and simply relied on the entity typing provided by BabelNet. The result is a drop in performance, confirming the need for this kind of validation. Then, we removed the *named entity augmentation* module presented in Section 3.1.5, i.e., named entities that are neither anchored in Wikipedia articles nor identified as synonyms of anchored ones, are no longer caught by the neural model. This removal causes a reduction of annotations, which results in an average decrease – over the three test sets – of 7.53 F1-score points. Subsequently, we also removed the *named entity confirmation* module, which used the BERT-based model to corroborate the annotations produced by the knowledge-based approach (Section 3.1.3). These annotations were obtained through an automatic approach, and our intuition suggested using a neural model to discard potentially imprecise sentences, which is confirmed by the further average decrease in performances of 4.29 points when removing it. At this stage, the neural model is only employed as a discriminator, i.e. it just outputs *NE* or *not NE* for each token. Hence, for each sentence, if there are tokens annotated as PER, ORG, LOC or MISC by the knowledge-based approach, but labeled as *not NE* by the model, the sentence is discarded. The removal of this *named entity discrimination* block again results in a worsening of performances by 5.91 points, on average. Finally, we also ablated the tag propagations of Section 3.1.4, i.e., we just left the tags associated with preexisting links in the article. Both tag propagation methods were used to increase the density of annotations, crucial for obtaining high-quality annotated sentences: in fact, their exclusion leads to a further deterioration in performance.

We can summarize this ablation study by pointing out an average gap of approximately 20 F1-score points between the final WikiNEuRal version detailed in Section 3.1 and the

Version	CoNLL	WikiGold	OntoNotes
WikiNEuRal	**76.94 ± 0.75**	**82.42 ± 0.33****	**71.98 ± 0.55**
- Concept vs. NE	76.24 ± 0.35	81.66 ± 0.22	71.69 ± 0.24
- NE Augmentation	68.34 ± 0.64**	75.83 ± 0.36	62.84 ± 0.40
- NE Confirmation	64.60 ± 0.56**	70.98 ± 0.42	58.57 ± 0.21
- NE Discrimination	59.19 ± 0.63	64.46 ± 1.21	52.76 ± 1.28
- Tag Propagation	57.15 ± 1.53	63.36 ± 1.55	51.77 ± 1.25

Table 3.6. Span-based micro F1 scores of WikiNEuRal versions on the three English CoNLL, WikiGold, and OntoNotes test sets. Statistical significance is computed using Student's t-test: ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance is expressed with respect to the row immediately below.

basic one, which only uses strings anchored in Wikipedia articles.

For completeness, we also constructed a baseline version of the WikiNEuRal dataset using just the neural model employed in Section 3.1.5 to annotate Wikipedia articles. The system trained on the resulting dataset achieved 69.46 ± 0.50 on CoNLL, 77.46 ± 0.43 on WikiGold and 64.53 ± 0.41 on OntoNotes, showing how the combination of neural and knowledge-based approaches adopted by the final version of WikiNEuRal is essential in order to achieve higher performances.

3.2 Domain Adaptation

Thanks to the use of Wikipedia, our automatically-created datasets cover a wide range of domains. However, in many cases tests are performed on a limited set of domains. To address this issue and enable the production of domain-fitting NER datasets, here we introduce our methodology for performing domain adaptation. This consists of a domain extraction technique which, later combined with the approach presented in Section 3.1, enables the creation of domain-adapted training data for NER systems when given domain-specific texts.

3.2.1 Category selection

We first select a general subset of Wikipedia categories, under the assumption that they reflect all domains. Let us start by considering the directed category graph $G = (C, E)$ of (English) Wikipedia, where a node $c \in C$ represents a Wikipedia category and $(c_1, c_2) \in E$

is an edge representing that c_1 is a parent category of c_2. First, since $|C| \approx 1.6$M (as of September 2020), we need to find a subset $\hat{C}$ of C such that the coverage of knowledge fields is maximized, whereas the word vector dimensionality is minimized. We compute $\hat{C}$ by taking all nodes in G with depth from the root node ≤ 2, yielding a total of ~ 1200 categories.

Since the majority of Wikipedia articles do not belong to any of the selected $\hat{C}$ categories, we need to compute a distribution over $\hat{C}$ for every category $c \in C \setminus \hat{C}$. We follow the intuition that the number of random walks from c to $\hat{c} \in \hat{C}$ is proportional to $P(\hat{c}|c)$; hence, we compute:

$$P(\hat{c}|c) = \frac{1}{k} \cdot \sum_{i=1}^{k} RW(c, \hat{c})$$

where k is the number of random walks performed and $RW(c, \hat{c}) = 1$ if a random walk starting from node c and walking only to parent nodes ends up on category $\hat{c}$.

Now, given an article w and its associated set of categories C_w,

$$P(\hat{c}|w) = 1 - \prod_{c \in C_w} (1 - P(\hat{c}|c)) \; \forall \; \hat{c} \in \hat{C}$$

can be interpreted as the probability of associating $\hat{c}$ with one of the categories of w; we discard this association in the case that $P(\hat{c}|w) < \sigma$.[7] The next natural step is to prune $\hat{C}$ so as to further reduce categories that are either too general or too specific. For all $\hat{c}$, we compute the unconditioned probability

$$P(\hat{c}) = \frac{1}{|W|} \cdot \sum_{w \in W} P(\hat{c}|w),$$

where W is the set of all Wikipedia articles, compute the median value m_v of all $P(\hat{c}) \; \forall \; \hat{c} \in \hat{C}$ and take the 600 elements of $\hat{C}$ closest to m_v, yielding the final set of supercategories $\mathcal{S} \subset \hat{C}$, which cover the general knowledge encoded in Wikipedia in a concise way.

[7] A manually-tuned threshold, set at $3 \cdot 10^{-4}$.

3.2.2 Domain embedding computation and domain extraction

We now use the above category selection to produce both interpretable and domain-aware embeddings, which we then use to select the best-fitting Wikipedia articles for producing our Named Entity tagged dataset. Let us now consider all Wikipedia articles W, a token t occurring in any of its articles, a supercategory $s \in \mathcal{S}$, the set of articles W_s associated with s, and a function f which takes as input a token t' and a collection D of Wikipedia articles, and returns the number of the occurrences of t' within the documents of D; we compute the relevance score of token t in supercategory s as:

$$P(t|s) = \frac{P(s|t)P(t)}{P(s)}$$

where:

$$P(t) = \frac{f(t, W)}{\sum_{t' \in W} f(t', W)}$$
$$P(s) = \frac{\sum_{t \in W_s} f(t, W_s)}{\sum_{t' \in W} f(t', W)}$$
$$P(s|t) = \frac{f(t, W_s)}{f(t, W)}.$$

By repeating the above computation for every token $t \in W$ (excluding stopwords) and every supercategory $s \in \mathcal{S}$, we obtain a large matrix $\mathbb{E}^{n \times m}$,[8] i.e., our category embeddings for the selected language.

The procedure for exploiting the above-mentioned embeddings in order to extract the main categories from a corpus of documents is formally described in Algorithm 1. The algorithm's core is a hierarchical aggregation of the probability distribution of tokens over categories: first, it averages the token-level distributions M to obtain a document-level distribution Z. Then it proceeds by taking the main categories C across the whole set of document distributions. Finally, only categories that appear at least δ times are considered as categories of that corpus. These extracted categories will be used to select the Wikipedia pages for silver-standard training data production which best fit the input document domains.

[8] n is the size of the Wikipedia vocabulary, $m = |\mathcal{S}| = 600$.

Algorithm 1 Domain extraction procedure

Inputs: Corpus D, Category embeddings $\mathbb{E}^{n\times m}$, Counting function f
Parameters: Threshold δ, Token-level top-k k_t, Document-level top-k k_d
Output: Main categories in D

$$\gamma(\mathbf{x}, k) \rightarrow [\hat{x}_1, \cdots, \hat{x}_n], \hat{x}_i = \begin{cases} 1 & \text{if } \hat{x}_i \in \text{top}_k(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}$$

$C \in \mathbb{N}^m \mid c_i = 0 \; \forall i \in [1, m]$
for $d \in D$ **do**
  $\mathbf{v} \in \mathbb{N}^n, \mathbf{v}_i = f(w_i, d) \; \forall w_i \in \mathbb{E}$
  $M \leftarrow \mathbf{v}^T \cdot \mathbb{E}$
  $R \in \mathbb{R}^{n\times m}, R_i = M_i^T \cdot \gamma(M_i, k_t)$
  $Z \in \mathbb{R}^m, Z_i = \frac{1}{n} \cdot \sum_{i=1}^{n} (R^T)_i \; \forall i \in [1, m]$
  $C \leftarrow C + \gamma(Z, k_d)$
end for
return $\{i \mid C_i \geq \delta \; \forall i \in [1, |\mathcal{S}|]\}$

Language	Articles	Sentences	Tokens	Avg. length	Avg. NEs	PER	ORG	LOC	MISC	O
English DA (CoNLL)	20k	29k	759k	21.41	1.55	12k	23k	6k	3k	0.54M
Dutch DA (CoNLL)	25k	34k	598k	17.69	1.44	17k	8k	18k	6k	0.51M
German DA (CoNLL)	20k	41k	706k	17.29	1.37	17k	12k	23k	3k	0.61M
English DA (OntoNotes)	35k	48k	1.18M	24.31	1.70	20k	13k	38k	12k	1.02M

Table 3.7. Statistics on the produced data on a fixed number of articles after applying the Domain Adaptation (DA) strategy to fit the domain distribution of the target dataset. "Avg. length" is the average sentence length and "Avg. NEs" is the average number of named entities per sentence.

3.2.3 Experimental Setup

We apply our domain adaptation technique (Section 3.2) to filter the Wikipedia articles used to create our training data and fit them to the domains of the test data. To this end, we use the CoNLL and OntoNotes test sets[9] where, except for the Spanish CoNLL test set, this kind of document split is provided. Specifically, we perform domain extraction as described in Algorithm 1 with parameters $\delta = 5$, $k_t = 50$, $k_d = 5$ to the test set documents; thus, we provide WikiNEuRal with articles whose domains, i.e., categories, match the ones extracted for the targeted corpus. Statistics are shown in Table 3.7.

[9]We strongly emphasize that we do not use anything from the test sets except their raw text.

Training set \ Test set	CoNLL			OntoNotes
	English	Dutch	German	English
WikiNEuRal	76.94 ± 0.75	77.40 ± 0.57	64.02 ± 0.54	71.98 ± 0.55
WikiNEuRal DA	**79.07 ± 0.51**	**79.07 ± 0.52**	**68.33 ± 0.46**	**74.38 ± 0.30**
WikiNEuRal	77.02 ± 0.66	79.20 ± 0.40	67.50 ± 0.31	75.25 ± 0.44
WikiNEuRal DA	**78.22 ± 0.57****	**81.27 ± 0.94**	**69.69 ± 0.57**	**77.58 ± 0.36**

Table 3.8. Span-based micro (top) and token-level macro (bottom) F1 scores on common NER benchmarks. DA stands for Domain Adaptation. Statistical significance is computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$.

3.2.4 Results

The results reported in Table 3.8 show that Domain Adaptation consistently improves performances over already state-of-the-art results, while requiring much less training data compared to the standard WikiNEuRal version (cf. Tables 3.2 and 3.7). Specifically, domain-adapted datasets attain large and consistent improvement over the general-purpose version of the WikiNEuRal dataset. This implies that the domain-adapted datasets are strongly specialized towards the domains targeted by the adaptation technique – as expected – therefore proving the effectiveness of the proposed strategy.

3.3 Fine-Grained NER

NER systems are traditionally trained to recognize broad, coarse-grained types of entities involving names of people, companies, or places. However, these general classifications often fall short in capturing the complexity of real-world entities, limiting the systems' effectiveness in specialized contexts where finer distinctions between entity subtypes are necessary. In contrast, fine-grained entity categorization enables more precise and nuanced identification of entities within a text, providing a deeper understanding of their roles and relationships. Consequently, besides applying domain adaptation strategies (cf. Section 3.2), advancing towards fine-grained entity recognition is crucial for improving the performance and applicability of NER systems in domain-specific tasks. To fill this gap, in this section, we first introduce a new comprehensive categorization for named entities (Section 3.3.1), and then extend the WikiNEuRal methodology introduced in Section 3.1 to automatically generate high-quality and fine-grained annotations for multilingual NER.

NER Tag	NER Class	# Wikipedia articles
PER	Person	1,886K
ORG	Organization	439K
LOC	Location	1,228K
ANIM	Animal	330K
BIO	Biology	16K
CEL	Celestial Body	13K
DIS	Disease	9K
EVE	Event	249K
FOOD	Food	15K
INST	Instrument	52K
MEDIA	Media	703K
MON	Monetary	2K
NUM	Number	1K
PHY	Physical Phen.	2K
PLANT	Plant	51K
SUPER	Supernatural	6K
TIME	Time	9K
VEHI	Vehicle	78K
Total	—	5,089K

Table 3.9. Our new set of finer-grained classes for NER and the number of Wikipedia articles, and therefore entities, that we map to each class.

3.3.1 Our New Set of Classes

In its standard formulation, NER distinguishes between four classes of entities: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). Although NER systems that use these four classes have been found to be beneficial in downstream tasks, we argue that they might be too coarse-grained and, therefore, not provide a sufficiently exhaustive coverage for entities. For these reasons, we introduce a new set of finer-grained NER classes, namely, Person (PER), Location (LOC), Organization (ORG), Animal (ANIM), Biology (BIO), Celestial Body (CEL), Disease (DIS), Event (EVE), Food (FOOD), Instrument (INST), Media (MEDIA), Monetary (MON), Number (NUM), Physical Phenomenon (PHYS), Plant (PLANT), Supernatural (SUPER), Time (TIME) and Vehicle (VEHI).

We design our set of classes starting from the 18 fine-grained classes used for OntoNotes 5.0 (Pradhan et al., 2012b), splitting and merging them to better fit the ED task. Specifically, first, we split the PRODUCT class of OntoNotes into three separate classes,

Algorithm 2 Self-Improvement Algorithm

Inputs: Corpus of raw Wikipedia and Wikinews documents W
Parameters: Integer n, Integer t
Output: MultiNERD Dataset D

$A \leftarrow \{w_1, \ldots, w_n\}, w_i \in W$
$D \leftarrow annotate(A)$
$M_D \leftarrow train(D)$
for i $\leftarrow 1, \ldots, t$ **do**
  $\hat{A} \leftarrow \{w'_1, \ldots, w'_n\}, w'_i \in W, A \cap \hat{A} = \varnothing$
  $A = A \cup \hat{A}$
  $D = annotate(A, M_D)$
  $M_D = train(D)$
end for
return D

namely, FOOD, INST and VEHI. These three classes are i) very different from each other, ii) very frequent in texts, and, iii) a NER classifier will easily predict the correct one, so keeping them separated helps in better clustering entities. Second, while they use 3 different tags (LOC, FAC and GPE) to represent locations, we use a single LOC class as, in this case, the 3 classes are very similar, and a NER classifier could easily get confused. Similarly, they use 4 classes QUANTITY, ORDINAL, CARDINAL, and PERCENT to express numeric values, while we use a single NUM class. Finally, we introduce 6 new classes, crucial for better distinguishing entities, that do not have a corresponding class in the OntoNotes categorization. Notably, animals, plants and diseases were not covered by the OntoNotes categorization.

Finally, in order to use the newly-introduced NER classes, we label each Wikipedia entity with one of them by means of the semantic classifier introduced in Section 3.1.3 with the initial seed of annotations expanded to cover the additional entity types. Table 3.9 provides an overview of the number of Wikipedia articles for each NER class.

3.3.2 Fine-grained Silver Data Creation with Self-Training

Based on this new fine-grained classification of Wikipedia pages, we now apply the same steps described in Section 3.1 to produce silver-standard training data for the NER task. Specifically, we again clear the content of Wikipedia pages (Section 3.1.1), identify and classify named entities in text (Sections 3.1.2 and 3.1.3), and propagate the annotations to

the other occurrences in the articles (Section 3.1.4).

Despite these steps enable multilingual and fine-grained annotations to be created, the resulting annotations are derived automatically and, therefore, may contain errors. For this reason, in Section 3.1.5 , we improved the quality of the annotations by combining them with the predictions of a Transformer-based neural classifier (mBERT + Bi-LSTM + CRF, Mueller et al., 2020). However, unfortunately, this strategy requires pre-existing annotated data in the same set of languages and with the same NER tags in order to train the NER classifier, and these are not available in our case. To cope with this issue, we employ the same Transformer-based architecture but drop the requirement of pre-existing training data by introducing a general, straightforward iterative strategy to jointly improve both the performance of the neural model and the quality of the data produced. Algorithm 2 illustrates the procedure. Essentially, it starts by taking a set A of n articles from W (line 1), and annotating them with the steps described in Sections 3.1.1-3.1.3 (line 2). Then, it uses the obtained annotated dataset D to train a neural model M_D (line 3). Here the iterative step begins: i) another (disjoint) set $\hat{A}$ of n articles is taken from W (line 5), ii) a larger set A is obtained by concatenating the new set $\hat{A}$ with the previous set A (line 6), and consequently iii) a larger dataset D is obtained, but this time using also the model M_D to validate the annotations produced by the steps in Sections 3.1.1-3.1.3 (line 7), and finally iv) a better model M_D is trained on the new (larger and more accurate) dataset D (line 8). Steps (i)-(iv) are repeated t times, where t is used to regulate the size and the quality of the final dataset D. In the $annotate(A, M_D)$ function, the neural model M_D is used to refine the annotations and reduce noise. Specifically, if the NER class of an entity predicted by the neural model is different from the one assigned by the semantic classifier (cf. Section 3.1.3), the corresponding sentence is discarded.

Finally, we apply this strategy to generate training data for Named Entity Recognition across 10 languages (details provided below). To incorporate a new textual genre—namely news—we extend this approach to WikiNews as well, given that the input format for both resources is identical. By merging the data from these two sources on a per-language basis, we create a comprehensive multilingual and multigenre dataset for NER, which we call MultiNERD.

		DE	EN	ES	FR	IT	NL	PL	PT	RU	ZH
Statistics	PER	79.2K	75.8K	70.9K	89.6K	75.3K	56.9K	66.5K	54.0K	43.4K	47.7K
	ORG	31.2K	33.7K	20.6K	28.2K	19.3K	21.4K	29.2K	13.2K	21.5K	22.2K
	LOC	72.8K	78.5K	90.2K	90.9K	98.5K	78.7K	100.0K	124.8K	75.2K	70.4K
	ANIM	11.5K	15.5K	10.5K	11.4K	8.8K	34.4K	19.7K	14.7K	7.3K	6.9K
	BIO	0.1K	0.2K	0.3K	0.1K	0.1K	0.1K	0.1K	0.1K	0.1K	0.1K
	CEL	1.4K	2.8K	2.4K	2.3K	5.2K	2.1K	3.3K	4.2K	1.2K	1.4K
	DIS	5.2K	11.2K	8.6K	3.1K	6.5K	6.1K	6.5K	6.8K	1.9K	2.2K
	EVE	4.0K	3.2K	6.8K	7.4K	5.8K	4.7K	6.7K	5.9K	2.8K	2.5K
	FOOD	3.6K	11.0K	7.8K	3.2K	5.8K	5.6K	3.3K	5.4K	3.2K	2.9K
	INST	0.1K	0.4K	0.6K	0.7K	0.8K	0.2K	0.6K	0.6K	1.1K	0.5K
	MEDIA	2.8K	7.5K	8.0K	8.0K	8.6K	3.8K	4.9K	9.1K	11.3K	6.9K
	MYTH	0.8K	0.7K	1.6K	2.0K	1.8K	1.3K	1.3K	1.6K	0.6K	0.7K
	PLANT	7.8K	9.5K	7.6K	4.4K	5.1K	6.3K	6.6K	9.2K	4.8K	5.2K
	TIME	3.3K	3.2K	45.3K	27.4K	71.2K	31.0K	44.1K	48.6K	22.8K	27.4K
	VEHI	0.5K	0.5K	0.3K	0.6K	0.6K	0.4K	0.7K	0.3K	0.5K	0.4K
	O (OTHER)	2.4M	3.1M	3.8M	3.8M	4.2M	2.7M	2.5M	3.4M	2.0M	2.1K
	Sentences	156.8K	164.1K	173.2K	176.2K	181.9K	171.7K	195.0K	177.6K	129.0K	115.0K
	Tokens	2.7M	3.6M	4.3M	4.3M	4.7M	3.0M	3.0M	3.9M	2.3M	2.4K
	Avg. sentence length	17.7	21.7	24.6	24.5	25.7	17.7	15.3	21.9	16.7	21.8
	Avg. NEs per sentence	1.4	1.5	1.6	1.6	1.7	1.5	1.5	1.7	1.3	1.7

Table 3.10. Statistics concerning the data produced.

3.3.3 Experimental Setup

In our experiments, we evaluate the quality of our data in two different settings:

1. In order to compare our dataset against previous silver datasets, we map our fine-grained annotations to the coarse-grained classes used by these datasets. Then, we train the model introduced in Section 3.1.6 on both our dataset and the other above-mentioned datasets. Finally, we compare the performance of the corresponding systems on gold-standard benchmarks for NER;

2. To measure the quality of our fine-grained annotations, we manually annotate a random sample of 1K English sentences, and compare the annotations produced by our methodology with the corresponding ground truths.

We implement our model with PyTorch using the Transformers library (Wolf et al., 2019) to load the weights of `BERT-base-multilingual-cased` (mBERT), and train each model configuration with an early stopping strategy using a patience value of 5. We use Adam optimizer (Kingma and Ba, 2015a) with learning rate of 10^{-3} and a cross-entropy loss criterion. We repeat each training on 10 different seeds, fixed across experiments, and report the mean of their span F1 scores computed with the official conlleval script.

Train \ Test	CoNLL				WikiGold	OntoNotes	BSNLP	
	EN	ES	NL	DE	EN	EN	RU	PL
WikiANN	56.85	53.55	55.76	44.39	57.05	36.43	51.85	53.50
WikiNER	73.05	75.07	74.75	64.03	81.98	71.16	65.99	62.31
WikiNEuRal	76.94	77.87	77.40	64.02	82.42	71.98	66.50	62.44
MultiNERD w/o Self-Improvement	69.75	71.04	70.58	57.99	76.88	65.12	60.41	58.43
MultiNERD	**77.11**	**78.20**	**77.84**	**65.22**	**83.11**	**72.45**	**67.39**	**62.94**

Table 3.11. Span-based micro F1 scores obtained by training a reference NER system on different automatically-created datasets (i.e. WikiANN, WikiNER, WikiNEuRal, MultiNERD) and testing on common NER benchmarks.

Training Data. We start by training our reference model on **MultiNERD**, the resource created using the steps described in this section using Wikipedia and Wikinews articles, with $n = 30K$ and $t = 8$. Our dataset covers 10 languages, namely Chinese, Dutch, English, French, German, Italian, Polish, Portuguese, Russian and Spanish (statistics are shown in Table 3.2). Then, we train the same model on **WikiNEuRal** (Tedeschi et al., 2021c), the approach for NER silver data creation described in Section 3.1, **WikiNER** (Nothman et al., 2013) and **WikiANN** (Pan et al., 2017), and compare the scores of the resulting systems. In the first setting, all datasets are tagged with the four standard entity types (PER, ORG, LOC, MISC), except for WikiANN – which does not contain the MISC label – and MultiNERD. Indeed, when evaluating WikiANN, only the PER, ORG and LOC classes are evaluated, while for MultiNERD we convert the fine-grained classes to the coarse-grained ones.

Test Data. For our first setting (Section 3.3.3), we use the same common gold-standard test sets as in Section 3.1.6, namely the **CoNLL-2002 and 2003** benchmarks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003a), **WikiGold** (Balasuriya et al., 2009), **OntoNotes 5.0** (Pradhan et al., 2012b) and **BSNLP-2017** (Piskorski et al., 2017).

For our second experimental setting (Section 3.3.3), instead, due to the absence of NER benchmarks that use our set of categories (Section 3.3.1), we conduct a manual evaluation to assess the quality of our dataset. Specifically, we randomly select[10] a sample of 1K English sentences, pre-annotated with the NER tags produced using our methodology, and confirm or replace the annotations associated with each token in the dataset. The resulting gold-standard dataset is used to analyze the quality of our silver-standard data.

[10]We ensure that the dataset contains a sufficient number n of instances for each NER class. We set n = 20. Statistics are provided in the "Support" column of Table 3.12.

Class	P	R	F1	Support
ANIM	0.72	0.52	0.60	60
BIO	0.78	0.66	0.71	32
CEL	0.96	0.96	0.96	25
DIS	0.99	0.88	0.93	78
EVE	0.96	0.93	0.94	27
FOOD	0.83	0.44	0.58	122
INST	0.90	0.53	0.67	34
LOC	0.99	0.99	0.99	262
MEDIA	0.95	0.95	0.95	41
O	0.99	1.00	0.99	11823
ORG	0.97	0.95	0.96	59
PER	0.99	0.99	0.99	217
PLANT	0.73	0.40	0.52	47
MYTH	0.88	0.96	0.92	23
TIME	0.99	0.96	0.98	171
VEHI	0.82	0.85	0.84	27
ALL	0.90	0.81	0.85	13048

Table 3.12. Fine-grained evaluation on the MultiNERD English test set.

3.3.4 Results

Coarse-Grained Evaluation. In our first setting, we measure the effectiveness of our methodology by comparing the quality of the data produced against that of other datasets created using previous state-of-the-art strategies for NER silver-data creation (i.e. datasets listed in Section 3.3.3). Since past approaches focused on coarse-grained entities, we can compare the quality only for such entity types. The results obtained are shown in Table 6.10. Although our dataset covers a wider range of categories than its competitors, it nevertheless outperforms all of them on all tested datasets and languages. We attribute this advancement mainly to the self-improvement algorithm introduced in Section 3.3.2, which iteratively refines the annotations using a better model at each iteration. To demonstrate the impact of our algorithm, we construct baseline versions of MultiNERD for DE, EN, ES, NL, PL and RU with the same sizes as the corresponding refined versions, but without using our enhancement procedure. As can be observed from Table 6.10, the refined versions provide an average improvement of almost 7 F1 points. In addition, the wider number of textual genres covered by MultiNERD leads to more robust systems.

Fine-Grained Evaluation Although the coarse-grained evaluation conducted in the previous section showed that our MULTINERD methodology creates high-quality annotations, independently of the language, it is not sufficient to understand how our annotation pipeline

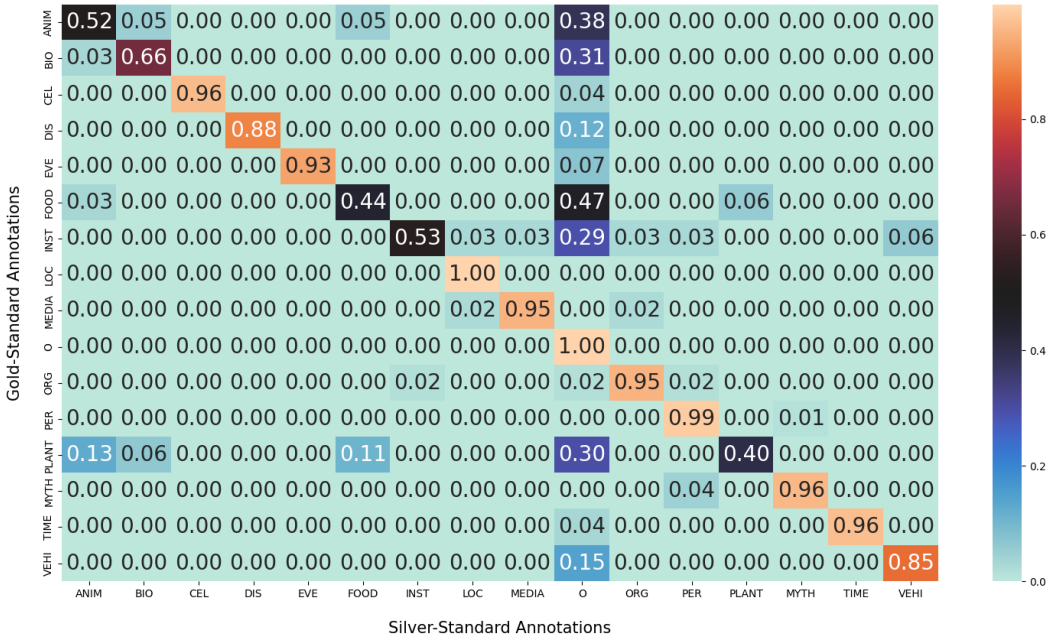

Figure 3.2. Confusion matrix of our silver-standard annotations compared to the corresponding ground truths.

performs on fine-grained classes. Indeed, to measure this, we use the sample of 1K English sentences manually-annotated with fine-grained entities, and report the results in Table 3.12. As expected, the PER, ORG and LOC classes are among the best-performing classes. Similarly, celestial bodies, diseases, events and media also have very high performance, thanks to their occurrences being almost always linked in Wikipedia and Wikinews articles (*high recall*) and easily distinguishable (*high precision*). In contrast, animals, biological entities, foods and plants have a high-degree of confusion (*lower precision*), and are very often not linked (*low recall*). To better explain this, we report in Figure 6.10 the confusion matrix of the silver-standard annotations produced by our approach compared to gold-standard ones. As an example, it can be observed that animals and plants are often confused with each other, mainly because their scientific names are morphologically very close. Similarly, animals and plants are also confused with foods (e.g. *Alaskan salmon* and *Quinoa*), and vice versa.

Even though the quality of the annotations produced by our approach for any particular one of the 10 languages covered herein is strongly dependent on the quality of the corresponding Wikipedia and Wikinews dumps, we expect comparable performance on all other languages, as suggested by the statistics in Table 3.2 which show strong consistency across languages.

3.4 Open-Source Data and Models

As a result of the above-described methodologies, we obtained high-quality, large-scale, multilingual datasets for Named Entity Recognition covering several languages languages (i.e. de, en, es, fr, it, nl, pl, pt, ru, zh) and domains, and consisting of millions of annotated sentences. To encourage future studies on multilingual NER, we publicly release both the WikiNEuRal[11] and MultiNERD[12] datasets on Hugging Face. Additionally, to further encourage the use of multilingual NER models, we fine-tune a multilingual language model (mBERT) for 3 epochs on our WikiNEuRal dataset on all 9 languages jointly and publicly release it on Hugging Face.[13] The limited model size (177M parameters) and its high performance enable NLP researchers and practitioners to efficiently and accurately process documents in multiple languages. Similarly, a multilingual NER model (Aarsen) trained on the MultiNERD dataset is made available on Hugging Face.[14] Further information and software details can be found on the Github repositories of the WikiNEuRal[15] and MultiNERD[16] projects.

3.5 Summary

Motivated by the lack of multilingual corpora for the NER task, we presented WikiNEuRal, an automatic, language-agnostic approach for generating labeled datasets for NER in multiple languages (Section 3.1). While we follow other silver-data creation approaches and exploit the hyperlinked texts of Wikipedia articles, we depart from past works in three fundamental aspects which integrate knowledge-based and neural approaches: i) we automatically type tags by utilizing the structure of a multilingual lexical-semantic knowledge base, BabelNet, ii) we exploit neural BERT-based models to discern entity from non-entity tags and iii) as a complementary approach to confirm and augment sentences with entity tags. Additionally, we put forward a domain adaptation technique which can produce NER training data for arbitrary domains (Section 3.2). Finally, we extend our strategy to a finer set

[11] https://github.com/Babelscape/wikineural
[12] https://github.com/Babelscape/multinerd
[13] https://huggingface.co/Babelscape/wikineural-multilingual-ner
[14] https://huggingface.co/tomaarsen/span-marker-mbert-base-multinerd
[15] https://github.com/Babelscape/wikineural
[16] https://github.com/Babelscape/multinerd

of categories and produce MultiNERD, the first multilingual, multi-genre and fine-grained dataset for NER (Section 3.3). To encourage future studies on NER and to enable the large-scale usage of NER systems, all the produced artifacts (i.e. both datasets and models), have been made publicly available to the research community (cf. Section 3.4).

Chapter 4

Low-Resource Entity Disambiguation

After NER (Chapter 3), the next key step in the information extraction pipeline is entity disambiguation, which ensures that identified entities are correctly matched to their real-world counterparts in a reference knowledge-base. However, as described in Section 2.2.4, one common issue with current disambiguation approaches is that they require massive amounts of training data – often millions of labeled items – in order to perform at their best, making the development of a high-performance ED system viable only to a limited audience. Indeed, differently from NER – where small, pre-defined set of categories are used to classify named entities – here there are millions of classes corresponding to the number of entities in the adopted KB. Additionally, when presented with an ambiguous entity mention, the models are much more likely to rank a more frequent yet less contextually relevant entity at the top, due to the high-popularity of some entities.

Aiming at addressing these issues, in this chapter, we first study whether it is possible to narrow the performance gap between systems trained on limited (i.e. thousands) and large amounts (i.e. millions) of labeled data (Section 4.1). Subsequently, we study the problem of overshadowed entities, and introduce an iterative approach that uses entity type information to leverage context and avoid over-relying on the frequency-based prior (Section 4.2).

4.1 Budget-Constrained Entity Disambiguation

While the first successful approaches to ED often relied on non-neural graph-based techniques (Hoffart et al., 2011b; Rao et al., 2013; Moro et al., 2014b), there is a growing body of

work that studies how to enrich neural models by taking advantage of relational knowledge from semantic networks such as Wikidata, YAGO (Suchanek et al., 2007), WordNet (Miller, 1995) and BabelNet (Navigli and Ponzetto, 2012a; Navigli et al., 2021), *inter alia*. As anticipated in Section 2.2, Raiman and Raiman (2018) proposed DeepType to integrate symbolic knowledge into a neural network and constrain the behavior of an entity prediction model with respect to the symbolic structure defined by types. Additionally, Bootleg (Orr et al., 2020), uses the Wikidata and YAGO edges to encode entity relations and entity types as input embeddings to a Transformer-based architecture. However, the sparsity of such types and relations may make them an unappealing option for low-data scenarios.

Inspired by previous studies, and based on the findings introduced in Chapter 3, we aim at exploiting NER to improve a strong entity disambiguation baseline in a low-resource, budget-constrained setting without requiring any additional data. Specifically, thanks to its coarse-grained classes, we look at NER as an obvious way to cluster entities and, therefore, to reduce the intrinsic sparsity of the entity disambiguation task. With this as our aim, in this section, we first describe a simple yet strong baseline into which we will plug our NER-focused contributions (Section 4.1.1). Then, we introduce a set of finer-grained NER classes that we will use for our experiments (Section 4.1.2) and use them to inject NER information into entity representations (Section 4.1.3), devise a NER-enhanced candidate generation module (Section 4.1.4), better select negative samples during training (Section 4.1.5), and introduce a NER-constrained decoding technique (Section 4.1.6).

4.1.1 Baseline System

Our baseline system for ED is composed of two main modules: a candidate generation module and a mention disambiguation module. Given an input sentence with pre-identified mentions, the former of the two modules is responsible for i) retrieving a set of candidate entities of any given mention from an alias table, and ii) reducing the size of this set by taking the top-k candidates according to their frequency in Wikipedia. The latter module is, instead, a neural architecture which features two Transformer-based encoders – one to represent a mention in context, the other to represent candidate entities – whose output states are used to assign the most appropriate entity to the considered mention.

More formally, let ϕ and ψ be the mention and entity encoders of the disambiguation

module. The disambiguation module uses ϕ and ψ to compute the cosine similarity score of each mention-candidate pair (m, c_i) for each $i \in \{1, \ldots, k\}$ and selects the highest-scoring entity ε as follows:

$$\varepsilon = \underset{i \in \{1,\ldots,k\}}{\operatorname{argmax}} \frac{\phi(m)^T \psi(c_i)}{\|\phi(m)\| \|\psi(c_i)\|} \tag{4.1}$$

Following Botha et al. (2020), the mention encoder ϕ takes as input a sequence of tokens in which the start and the end of the mention m is identified by special tokens ([E] and [/E]) and surrounded by left and right contexts of at most 64 tokens, whereas the entity encoder ψ models each entity by taking as input the first 128 tokens of the corresponding Wikipedia article.

4.1.2 Fine-Grained Classes for NER

NER typically distinguishes between the PER, LOC, ORG, and MISC classes. However, we argue that this categories might be too coarse-grained and, at the same time, not provide a sufficiently exhaustive coverage to also benefit ED, as many different entities would fall within the same MISC class. For these reasons, in our study we rely on the new set of finer-grained NER classes introduced in Section 3.3.1.

4.1.3 NER-Enhanced Entity Representation

In the baseline ED system we presented in Section 4.1.1, the aim of the mention encoder ϕ is to produce a dense mention representation that is as similar as possible to the representation produced by the entity encoder ψ for its most appropriate entity. The better and richer the representations for the candidate entities are, the easier it will be for the system to disambiguate the corresponding mentions. One way to enrich the representation of each candidate entity is to make the entity encoder aware of class information. In particular, together with the textual description of an entity, we propose also providing the NER class as an additional input to the entity encoder. More specifically, we prepend the NER tag of a Wikipedia entity to its textual description and feed this enhanced string to the entity encoder. Not only does this feature help the entity encoder to better distinguish between entities that belong to different NER classes, but it also leads the mention encoder to consider such classes indirectly when producing the dense representation of a mention.

4.1.4 NER-Enhanced Candidate Generation

In the candidate generation step, the aim is to select a suitable set of candidates for each mention in context. The desired properties for such a set of candidates are high recall – target entities should as frequently as possible be within the corresponding candidate sets – but also a small number of candidates to choose from, so as to make the disambiguation step as easy as possible. However, the majority of mentions tends to have dozens of candidates – the most common mentions also being the most ambiguous, following the Zipf's law – and, therefore, in order to satisfy the second desired property, several ED systems set an upper bound to the size of the candidate set, in this way hampering candidate recall. Moreover, selecting this upper bound adds another layer of complexity to finding the best trade-off between recall and size.

In this section, instead, we propose a strategy for considerably decreasing the size of the candidate set while also increasing its recall. Specifically, we train and employ a NER classifier to predict the NER class of an input mention in context, and then discard all the candidates whose class is different from the predicted one. For example, consider the sentence in Figure 4.1, where the mention *Tesla* would normally have a total of 18 candidate entities to choose from. If we limited the candidates to the 10 most popular entities, then the correct entity *Tesla (band)* would be left out, making it impossible for the system to disambiguate the input correctly. Instead, thanks to our proposed strategy, if the NER classifier predicts the correct NER class, namely ORG, all the candidates that do not correspond to organizations will be discarded from the candidate set. In our example, as we can see in Figure 4.1, not only does our NER-filtered candidate set include the target entity, but it is also considerably smaller (4 candidates instead of 18). Our NER classifier is a BERT-based model, which takes as input a mention in context and outputs one of the 18 classes introduced in Section 4.1.2.

4.1.5 NER-based Negative Sampling

Our baseline system learns to model the representation of a mention by comparing it with the representations of the corresponding candidate entities, both correct and wrong ones. However, some mentions are unambiguous (i.e., they have only one possible candidate entity), leading to sub-optimal learning. One common way to overcome this problem is to

Figure 4.1. Example of the NER-enhanced Candidate Generation module. The *Tesla* mention has 18 candidates, and including only the 10 most popular entities in the candidate set, the target entity *Tesla (band)* would be not included. Applying our strategy, instead: i) the correct entity is included and, ii) the dimension of the resulting set is significantly smaller.

add negative samples. In ED, negative samples are simply entities added to the candidate set of a mention with the aim of letting the model learn more accurate mention representations which are "semantically" near to the representations of the target entities and far from the ones of the negative samples (our baseline system already makes use of them).

Although adding negative samples indiscriminately has already been proven to be beneficial for ED, we propose a more refined approach in which we select specific negative samples according to their NER class. In particular, given a mention, its target entity and its NER class c, we enlarge the candidate set at training time by adding a number of negative

samples belonging to the same class c. The main motivation for using this NER-based negative sampling strategy is to make the training process more challenging and further stress the system to produce better representations. Indeed, the textual descriptions of entities belonging to the same NER class are often similar and follow recurring patterns – e.g., in Wikipedia a person is usually described by their date, place of birth and occupation – and therefore adding NER-based negative sample encourages the underlying neural network to rely on entity-specific features.

4.1.6 NER-Constrained Decoding

So far, we have introduced a few strategies to enhance an ED baseline by exploiting NER at the input level or during the training process. In this Section, instead, we propose a strategy that uses NER to improve our ED baseline at the output level by enforcing "soft" and "hard" constraints at inference time. Our intuition is that, for very ambiguous mentions, an ED system may be biased towards very frequent entities, independently of the context such mentions appear in. In order to mitigate this issue, we propose constraining our ED system to output an entity whose NER class is consistent with the prediction(s) of a NER classifier for the same input mention. In our experiments, we distinguish between "hard" and "soft" constraints, with the difference being that in the former we force the entity predicted by the ED system to be exactly the same as that predicted by the NER classifier, whereas in the latter we force the entity to belong to one of the top-k predictions of the NER classifier.

We also analyze two alternatives for building the NER classifier: i) training a separate model as we do in Section 4.1.4, or ii) training the ED system not only to assign the most appropriate entity to a mention in context but also to provide its NER class. The advantage of the second approach is that it requires just a single model and, therefore, fewer computational resources. However, performing both tasks jointly results in worse scores in NER labeling, which, in turn, decreases the benefits of our NER-constrained decoding strategy for the overall ED system.

4.1.7 Combinations of NER Contributions

Some of our contributions can be combined to bring further improvements. For example, it is possible both to enhance entity representations (Section 4.1.3) and also to apply our NER-

System	training instances	AIDA accuracy
GENRE	18K	88.6
Our ED baseline	18K	88.8
Our ED baseline **+ NER**	18K	**92.5**
GENRE	9000K	93.3

Table 4.1. The first two rows show the InKB accuracy of our baseline system and GENRE trained, validated and tested on the AIDA-YAGO-CoNLL (in-domain setting). The last row shows the in-domain accuracy of GENRE when pretrained on KILT which is made up of 9 million training instances. Our NER-enhanced ED system is almost able to bridge the gap in performance when trained on only 18 thousand instances.

constrained decoding strategy (Section 4.1.6). Similarly, it is possible to add NER-based negative samples during training to let the model produce more accurate representations (Section 4.1.5) and also apply our decoding strategy; or even to combine all three of the above-mentioned contributions. One interesting combination consists in first removing all the candidates whose NER class is different from the one predicted for the input mention (Section 4.1.4), and then increasing the size of the candidate set by adding negative samples of the same class (Section 4.1.5), making the training process more challenging.

4.1.8 Experimental Setup

Architecture. We implemented our NER classifier, our baseline ED model, and our NER-based enhancements for ED with PyTorch, using the Transformers library (Wolf et al., 2020) to load and fine-tune the weights of `BERT-large-uncased`. We trained each model configuration for 30 epochs, adopting an early stopping strategy with a patience value of 5, with Adam (Kingma and Ba, 2015b) and a learning rate of 10^{-5}, as standard when fine-tuning the weights of a pretrained language model. We use the same NER classifier for all experiments except for those that involve jointly learning NER and ED. Our NER classifier achieves 97.1% in terms of accuracy on the AIDA-YAGO-CoNLL test set. In the remainder of this section, we report the results of the best model checkpoints according to their F1 score on the validation split of the AIDA-YAGO-CoNLL dataset computed at the end of each training epoch. All model training was carried out on a NVIDIA GeForce RTX 3090. It required $\sim$2h/epoch on the AIDA-YAGO-CoNLL training set, for an average number of $\sim$20 epochs.

Datasets. In the following, we describe the datasets we use to train, validate and test our contributions:

- **AIDA-YAGO-CoNLL** (Hoffart et al., 2011b) is one of the largest manually annotated ED datasets for English as it contains 388 articles with 27,817 linkable mentions corresponding to the named entities annotated for the original CoNLL-2003 entity recognition task (Tjong Kim Sang and De Meulder, 2003b) This dataset comprises a number of newswire articles taken from the Reuters Corpus.

- **MSNBC, AQUAINT and ACE2004** are smaller evaluation sets, cleaned and updated by Guo and Barbosa (2017). MSNBC consists of 20 news articles from 10 different topics (two articles per topic) and 656 linkable mentions. AQUAINT is made up of 50 documents and 727 linkable mentions from the Xinhua News Service, the New York Times and the Associated Press. Finally, ACE2004 features a set of 35 news articles and 257 linkable mentions.

- **WNED-WIKI and WNED-CWEB** are larger, but automatically extracted, evaluation sets for ED. They were built from the ClueWeb and Wikipedia corpora by Guo and Barbosa (2017) and Gabrilovich et al. (2013). WNED-WIKI, or simply WIKI, consists of 320 documents and 11,154 mentions, while WNED-CWEB, or simply CWEB, consists of 320 documents and 6,821 mentions.

We stress that we train each of our model configurations on only the AIDA-YAGO-CoNLL training split, i.e., on only 18K labeled instances as opposed to the millions on which current state-of-the-art systems are trained, showing the benefits of NER when a scarce amount of labeled instances are available.

4.1.9 Results

In what follows, we first show the overall results of our NER-enriched approaches for ED on an in-domain evaluation, and then we focus on the individual benefits of each contribution. Finally, we show that such contributions are robust and beneficial in out-of-domain evaluations.

Model	AIDA
Our ED baseline	88.8
w/ NER-enhanced Representations (NER-R)	89.3
w/ NER-based Negative Sampling (NER-NS)	89.6
w/ NER-enhanced Candidate Generation (NER-CG)	89.4
w/ NER-constrained Decoding (NER-CD)	92.2
w/ NER-R + NER-NS	89.7
w/ NER-R + NER-CD	92.3
w/ NER-NS + NER-CG	90.0
w/ NER-NS + NER-CD	92.4
w/ NER-R + NER-NS + NER-CD	**92.5**

Table 4.2. Accuracy of our proposed NER-based contributions and their combinations on the AIDA-YAGO-CoNLL test set. Each contribution improves the performance of the baseline, and two or more NER-based approaches can be combined to further improve the results.

NER for EL. As can be seen in Table 4.1, when trained on only the 18K instances of the AIDA-YAGO-CoNLL training split, our baseline ED system obtains results that are on par – 88.8% against 88.6% in accuracy on the test split of AIDA-YAGO-CoNLL – with those of GENRE (Cao et al., 2021) which is, on average, the current best-performing system across the datasets described in Section 4.1.8. Table 4.1 also shows that GENRE benefits greatly from drastically increasing the size of the training set from 18K to 9000K labeled instances (Petroni et al., 2021), gaining almost 5 points in accuracy. As we previously discussed, this improvement comes at the cost of much longer training times and/or more expensive hardware. However, if we put together our NER-focused contributions, they allow our baseline ED model to significantly narrow this gap, improving accuracy on the AIDA-YAGO-CoNLL test set by almost 4 points, while still using the original small training set with only 18K labeled instances.

What contributes to these results? One may wonder what the most important contributions are among the NER-focused approaches we propose. Table 4.2 reports the results of each of the contributions we described in Sections 4.1.3-4.1.7. As one can see, even the smaller contribution, i.e., enriching the representations of an entity by including its NER class (see Section 4.1.3), provides an improvement of 0.5 points in accuracy, while the most beneficial individual contribution is our NER-based constrained decoding strategy (Section 4.1.6), which provides an improvement of 3.6 points in accuracy. Moreover, we also observe

Model	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg.
Our ED baseline	86.9	61.8	89.9	65.7	58.5	72.6
w/ NER-enhanced Representation (NER-R)	86.9	62.2	89.9	66.1	59.0	72.8
w/ NER-based Negative Sampling (NER-NS)	87.0	62.4	90.0	66.5	59.2	73.0
w/ NER-enhanced Candidate Gen. (NER-CG)	87.0	62.3	89.9	66.4	59.2	73.0
w/ NER-Constrained Decoding (NER-CD)	89.2	68.2	91.2	68.1	63.8	76.1
w/ NER-R + NER-NS	87.0	62.5	90.0	66.3	59.5	73.1
w/ NER-R + NER-CD	88.7	69.2	91.2	68.3	63.5	76.2
w/ NER-NS + NER-CG	87.8	65.4	90.2	66.8	60.5	74.1
w/ NER-NS + NER-CD	89.1	69.2	91.2	68.4	63.8	76.3
w/ NER-R + NER-NS + NER-CD	**89.2**	**69.5**	**91.3**	**68.5**	**64.0**	**76.5**

Table 4.3. InKB accuracy of our ED baseline system and of the NER-based contributions on the out-of-domain test sets of MSNBC, AQUAINT, ACE2004, WNED-CWEB and WNED-WIKI.

that several of our contributions are complementary, in that their combinations bring further improvements, with the best combination attaining an accuracy of 92.5% on the test set of AIDA-YAGO-CoNLL.

Out-of-domain results. Finally, Table 4.3 shows that our NER-focused contributions bring benefits on out-of-domain evaluations too. Similarly to what we observed in the in-domain setting, our individual contributions consistently improve the results across popular out-of-domain test sets, namely, MSNBC, AQUAINT, ACE2004, CWEB and WIKI. Moreover, the combination of two or more NER-based approaches brings further improvements, totaling a net gain of 3.9% of absolute improvement in average accuracy (or a 14% reduction in error rate) with respect to our already competitive ED baseline. While our main objective is not to propose a state-of-the-art model for ED, we observe that the application of NER is particularly beneficial on ACE2004, where our NER-enhanced ED system attains state-of-the-art results – 91.3% in accuracy compared to 91.2% of Fang et al. (2019) – trained only on the 18K sentences of AIDA-YAGO-CoNLL.

4.1.10 Analysis

One could argue that our NER-based contributions could still be effective with the four standard NER classes, i.e., Person, Organization, Location and Miscellaneous. However, as we can see from Table 4.4, using the standard NER classes greatly reduces the benefits of our NER-based contributions, especially due to the fact that the Miscellaneous class

Ablation – standard NER classes	AIDA
Our ED baseline	88.8
w/ NER-enhanced Representation (NER-R)	89.0
w/ NER-enhanced Candidate Gen. (NER-CG)	89.1
w/ NER-based Negative Sampling (NER-NS)	89.1
w/ NER-Constrained Decoding (NER-CD)	90.7

Table 4.4. Results of our NER contributions on the AIDA-YAGO-CoNLL test set when using the four standard NER classes, i.e., PERSON, LOCATION, ORGANIZATION and MISCELLANEOUS.

NER class	Baseline	+NER	Δ
PER	95.8	96.5	+0.7
ORG	81.7	89.3	+7.6
LOC	93.4	94.3	+0.9
ANIM	66.7	100.0	+33.3
EVE	42.4	51.8	+9.4
FOOD	0.0	66.7	+66.7
INST	100.0	100.0	+0.0
MEDIA	90.0	95.0	+5.0
MON	100.0	100.0	+0.0
NUM	100.0	100.0	+0.0
PLANT	80.0	80.0	+0.0
SUPER	64.7	70.6	+5.9
TIME	60.0	80.0	+20.0
VEHI	86.7	100.0	+13.3

Table 4.5. Per-class accuracy of the entities in the AIDA-YAGO-CoNLL test set. Our NER-based contributions increase the performance of the baseline system on each entity class, especially the most difficult ones.

conflates several heterogeneous entity types into a single cluster.

In Table 4.5, instead, we report the per-class accuracy of our best system compared to our baseline, showing where our NER-based contributions bring more improvements. In general, our contributions positively affect each class, in particular ANIM (+33.3% in accuracy), FOOD (+66.7%) and TIME (+20.0%), i.e., our contributions help to correctly classify instances that are more difficult or rare.

Finally, in Table 6.1 we provide a qualitative look at a few examples in which our NER-based contributions aid the ED system in choosing the correct named entity.

Sentence	NER class	Prediction
Japan then laid siege to the Syrian penalty area for most of the game but rarely breached the **Syrian** defence.	ORG	**Baseline:** Syria **+NER**: Syria national football team
"You will not win the war of the Polish beer market with imported international brands", van Boxmeer said, adding that **Heineken** would remain an up-market import in Poland.	FOOD	**Baseline:** Heineken N.V. **+NER**: Heineken
Zieleniec led calls for the party and its leadership to listen to more diverse opinions, a thinly-veiled criticism of Klaus who has spearheaded the country's **post-Communist** economic reforms.	TIME	**Baseline:**: Democratic Left Alliance **+NER**: Post-communism
If successful the changes could get incorporated into future **Mars** missions Spirit and Opportunity were also fitted with a new navigation system that allows them to think several steps ahead.	CEL	**Baseline:**: Mars Pathfinder **+NER**: Mars
The five breeds credited with the most incidents were chow chows, Rottweilers, German shepherds, cocker spaniels and **Dalmatians**.	ANIM	**Baseline:**: Dalmatian Action **+NER**: Dalmatian (dog)

Table 4.6. Examples of sentences where the NER contributions help avoid errors. "Baseline" and "+NER" stand for the baseline system and the NER-enhanced best performing system, respectively.

4.2 Overshadowed Entities

According to previous research, current ED systems are prone to over-relying on prior probability instead of focusing on context information, which causes them to underperform on overshadowed entities. In benchmarking experiments performed by Provatorova et al. (2021), all ED systems under evaluation appeared to have a large performance gap between Top and Shadow subsets of the ShadowLink dataset, where Top contains most frequent entities and Shadow contains their overshadowed counterparts. The results of a human evaluation experiment in the same study indicate that the challenge of entity overshadowing is unique to automated ED methods: human participants achieved equally good results at disambiguating entities sampled from Top and Shadow. These findings call for further research in the field of ED, with the goal of building a method that outperforms existing systems on overshadowed entities while still achieving competitive results on standard datasets.

To mitigate this issue, we introduce NICE (NER[1]-enhanced Iterative Combination of Entities), a combined entity disambiguation algorithm designed to tackle the challenge

[1]Named Entity Recognition (Yadav and Bethard, 2018)

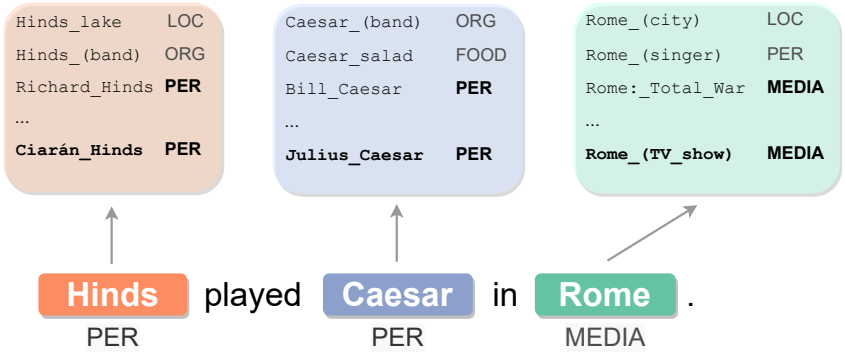

Figure 4.2. An example of filtering candidate entities by their NER types. A candidate entity is selected (highlighted in **bold**) if its NER type coincides with the predicted NER type of the corresponding entity mention. For simplicity, we assume the confidence score to be higher than the threshold, so that only the first predicted type is considered for filtering.

of entity overshadowing by focusing on three aspects of context-based information: entity types, entity-context similarity and entity coherence. The NICE pipeline includes a NER-enhanced candidate filtering module designed to improve robustness on overshadowed entities (Section 4.2.2), a pre-scoring module that calculates semantic similarity between a candidate entity and a mention in context, and an unsupervised iterative disambiguation algorithm that maximises entity coherence (Section 4.2.4), combining the relatedness scores between candidate entities with the scores of the semantic similarity module (Sections 4.2.4-4.2.5). To the best of our knowledge, this study is the first attempt to build an entity disambiguation method designed specifically to tackle the problem of entity overshadowing.

In the remainder of this section, we perform a systematic evaluation of the NICE method, and use our experimental results to answer the following research questions:

- **RQ1:** Does focusing on context information improve ED performance on overshadowed entities?

- **RQ2:** Does focusing on context information instead of relying on mention-entity priors in ED allow to maintain competitive performance on more frequent entities?

- **RQ3:** In what ways do the different aspects of context information contribute to ED performance on overshadowed entities?

4.2.1 The NICE method

Our method is based on the assumption that the main challenge in disambiguating overshadowed entities stems from over-relying on entity commonness, and therefore switching the focus to the context (entity relatedness) can improve the performance. We consider three main ways of extracting information from the context: (1) using mention-entity similarity to predict entity types and improve candidate filtering, (2) using word embeddings enhanced with entity types to measure semantic similarity between an entity and its context, and (3) using entity-entity similarity to make sure that the entity disambiguating decisions within one document are coherent (collective disambiguation).

4.2.2 Candidate filtering

Adding the step of filtering candidate entities before disambiguation brings the benefits of reduced inference time and potential improvements in accuracy. To perform this step in the NICE method, we follow the work of Tedeschi et al. (2021a) by using entity type information. Given an entity mention m surrounded by textual context $(cont_{left}, cont_{right})$ and a list of candidate entities $cands = \{e_1, \ldots, e_n\}$, we use a NER classifier to predict the top-k possible entity types of m. Then, we discard all candidate entities that have an entity type not matching any of these k classes:

$$cands_{filtered} = \{e_i : type(e_i) \in \hat{T} \mid e_i \in cands\},$$

where $\hat{T}$ is the set of top-k predicted entity types. If the confidence score of the NER classifier is above a threshold value t, only one class is used instead of k. In the current setup of the NICE method, the number of top predicted classes is $k = 3$ and the confidence threshold value is $t = 1$, which means that the classifier always outputs the top-3 entity classes. Figure 4.2 shows an example of NER-based candidate filtering.

To obtain the entity types for the candidates, we use the Wiki2NER dictionary provided by Tedeschi et al. (2021a)[2]. Then, instead of using the NER classifier from NER4EL, which has been trained only on the AIDA training set and therefore may be biased towards frequent entities, we introduce a refined version of it, which is more robust

[2] https://github.com/Babelscape/ner4el

to overshadowing. Specifically, we filter the training set of BLINK (Wu et al., 2020)[3] by discarding the entries where the ground truth answer has the highest popularity score among all candidate entities. Then, we use the 2M remaining data entries to fine-tune the classifier. The motivation behind fine-tuning the classifier rather than training it from scratch is to achieve an improvement in recognising overshadowed entities without losing the knowledge about more popular entities. This way, we obtain a system that specialises on overshadowed entities while still performing well in a standard ED setup.

4.2.3 Candidate pre-scoring

To obtain relevance scores for the candidate entities before the collective disambiguation phase, NICE relies on semantic similarity between a candidate entity and its corresponding mention in context. To calculate these semantic similarity scores, we use NER-enhanced word embeddings produced by the NER4EL model. This model uses a dual-encoder Transformer-based architecture that, given as input a mention-candidate pair $\langle m, c \rangle$, produces a vector representation for both m and c. Then, the relatedness between the two vectors is measured as the cosine similarity score. To encode the mention, the model uses both the mention itself and its context as input. To encode the candidate entities, instead, the model uses their textual descriptions from Wikipedia.

4.2.4 Collective disambiguation

We follow the assumption that the entities within one document are coherent, i.e. there exists a coherence measure C such that $C(e_1^*, ..., e_n^*) \geq C(e_1, ..., e_n)$ where $e_1^*, ..., e_n^*$ are the correct entities for mentions $m_1, ..., m_n$, and $e_1, ..., e_n$ is any other selection of candidate entities for these mentions (with one candidate per mention). Then, we use the following heuristic to reduce the search space and speed up the disambiguation process: if the entities within one document are coherent, and different mentions have different degrees of ambiguity (different sizes of candidate sets), then using an iterative process that starts with the least ambiguous mention may reduce the chance of wrong disambiguation decisions. We thus propose ICE (for "Iteratively Combining Entities"), a collective disambiguation algorithm based on the iterative process described in Algorithm 3. The input of the algorithm

[3]BLINK is a dataset for ED consisting of 9M entries extracted from Wikipedia.

Algorithm 3 ICE Disambiguation

Input: $S_{todo} = \{ment_i : cands_i, scores_i\}_{i=1}^{n}$ where $cands_i = [cand_i^1, ..., cand_i^{k_i}]$ and $scores_i = [score_i^1, ..., score_i^{k_i}]$

Output: $S_{ans} = \{ment_i : cand_i^*\}_{i=1}^{n}$

$m_{seed} \leftarrow find\_least\_ambiguous(S_{todo})$
$S_{todo} \leftarrow S_{todo} \setminus \{m_{seed}, cands_{seed}\}$
$cand_{seed} \leftarrow disamb\_seed(m_{seed}, cands_{seed})$
$S_{ans} \leftarrow S_{ans} \cup \{m_{seed}, cand_{seed}^*\}$
while $S_{todo} \neq \emptyset$ **do**
  $m_i \leftarrow find\_least\_ambiguous(S_{todo})$
  $S_{todo} \leftarrow S_{todo} \setminus \{m_i, cands_i\}$
  $cand_i^* \leftarrow argmax\ aggr\_rel(cands_i, scores_i, S_{ans})$
  $S_{ans} \leftarrow S_{ans} \cup \{m_i, cand_i^*\}$
end while
return S_{ans}

is a set of entity mentions within one document, where every mention is matched with a pre-scored list of candidate entities. The first step of the disambiguation process consists of selecting a seed mention: the least ambigous entity mention, i.e., one that has the lowest number of candidates. The seed mention is disambiguated without using entity relatedness information, relying on the input scores instead. Then, an iterative process begins: on every step of the algorithm the least ambiguous entity mention is selected and removed from the set of unprocessed mentions. This mention is disambiguated based on coherence: for every candidate entity, the algorithm calculates an aggregated relatedness score between the entity and the set of already disambiguated entities, and chooses the candidate that maximises this score. The algorithm is terminated when the set of unprocessed entity mentions is empty.

Note that the relatedness score in Algorithm 3 can be calculated and aggregated in different ways. In the setup used in NICE, we calculate the score of each candidate as a weighted average between the entity coherence score and the original input score:
$score_{final} = \alpha score_{coherence} + (1 - \alpha) score_{input}$.
The process of choosing the parameter values is described in detail in the next subsection.

4.2.5 Parameters of the method

The NICE method has four main parameters to experiment with: confidence threshold t of the candidate filtering model, weight α of the ICE coherence score, the relatedness measure used in collective disambiguation and the relatedness aggregation method. To find the best combination of the parameters, we sampled a development set from the training data of ShadowLink and used grid search. The development set contains 100 Top and 100 Shadow entities: the two classes are represented equally because a system cannot distinguish between Top and Shadow on the inference step. For the confidence threshold value t, we considered the values from 0.5 to 0.9 with the step 0.1. For the ICE score weight α, the search space consisted of the values between 0 and 1 with the step 0.1.

To find the most suitable entity relatedness measure, we experimented with the seven measures available in the WAT relatedness API [4] (Piccinno and Ferragina, 2014), which retrieves relatedness scores by Wikipedia IDs of the corresponding entity pages. The measures available in the API are the following: PMI (Pointwise Mutual Information), Milne-Witten, LM (language model), WordVec, Jaccard, Barabasi-Albert, and conditional probability.

For aggregation of the relatedness scores in Algorithm 3, we considered three setups: *max*, *min* and *avg*. The *max* setup is a greedy approach, where the score of a candidate entity is calculated as its maximum pairwise relatedness to the already disambiguated entities. It relies on the assumption that a relevant entity does not necessarily need to be close to **all** other entities in the document: it is enough to have a high pairwise relatedness with just one other entity. The opposite setup is *min*: it follows a strict assumption that all entities within one document must be coherent, and calculates the aggregated score as the minimum pairwise relatedness to the already disambiguated entities. Finally, the *avg* setup calculates the score as a mean pairwise relatedness between a candidate and the disambiguated entities. Figure 4.3 shows the scores achieved on the development set by using the combinations of the seven relatedness measures and three aggregation setups. One point on the figure represents the score achieved with one combination of the four parameters of NICE. The

[4] https://sobigdata.d4science.org/web/tagme/wat-api

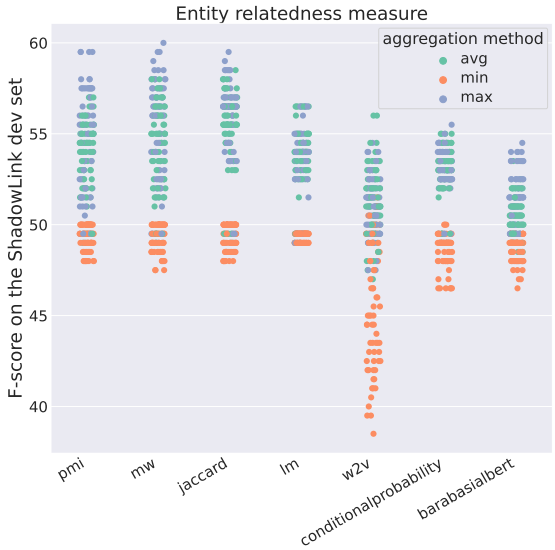

Figure 4.3. Micro F-score values achieved on the ShadowLink dev set with different combinations of relatedness measures and aggregation methods.

best relatedness measure turned out to be Milne-Witten (Milne and Witten, 2008):

$$MW(e_1, e_2) = \frac{\log \frac{\max(|L_{e_1}|,|L_{e_2}|)}{(|L_{e_1} \cap L_{e_2}|)}}{\log \frac{|W|}{\min(|L_{e_1}|,|L_{e_2}|)}},$$

where L_{e_i} is the set of Wikipedia pages that link to the page of entity e_i, and W is the entire Wikipedia. The best aggregation method according to our experiments is taking the maximum value of relatedness:

$$s(e_i) = max\{MW(e_i, \hat{e}_j) | \hat{e}_j \in S_{ans}\},$$

where S_{ans} is the set of entities already processed by the algorithm and e_i is a candidate entity.

Figure 4.4 demonstrates a heatmap of F-scores achieved on the ShadowLink development set with different combinations of candidate filtering threshold values and relatedness score weights while using the Milne-Witten relatedness measure and the *max* aggregation method. The value -1 of the filtering confidence threshold indicates not using the candidate filter at all. From the figure it can be seen that the best combination of the two remaining parameters is the confidence value of 1 (which means always predicting top-3 entity classes) and the weight 0.7 assigned to the relatedness score, which corresponds to the weight 0.3 of

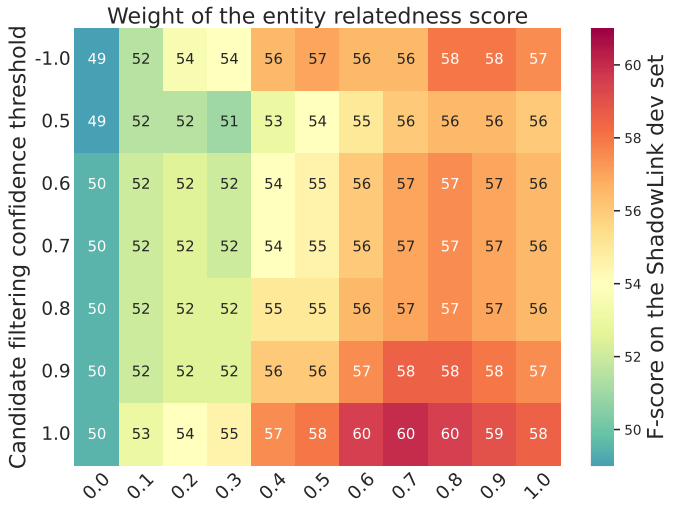

Figure 4.4. Micro F-score values achieved on the ShadowLink dev set using Milne-Witten entity similarity measure and *max* aggregation method.

the semantic similarity score. Thus, the final score of each candidate entity is calculated as:

$$s(e_i) = 0.7 \max\{MW(e_i, \hat{e}_j) | \hat{e}_j \in S_{ans}\} + 0.3 s_{input}(e_i).$$

4.2.6 Implementation details

Our contribution is focused on the task of entity disambiguation, where the mention boundaries and the candidate entities are given. We note that different methods of mention detection and candidate generation can be combined with the NICE disambiguation module to perform end-to-end entity linking. To perform collective disambiguation on ShadowLink, where only one entity span per entry is available, we used TagMe API [5] (Ferragina and Scaiella, 2010) for mention detection and REL[6] (van Hulst et al., 2020) for candidate generation.

4.2.7 Experimental setup

In this section, we describe the datasets we use to evaluate our methods (Section 6.2.5) and the baseline systems we compare with (Section 4.2.7).

[5] https://sobigdata.d4science.org/web/tagme/tagme-help
[6] https://github.com/informagi/REL

Datasets. As the NICE method is designed specifically for improving disambiguation of overshadowed entities, the main dataset for evaluating its performance is ShadowLink (Provatorova et al., 2021). ShadowLink contains ambiguous entities with short textual context gathered from the Web: it includes the Top subset which contains popular entities, and the Shadow subset which contains overshadowed entities, where every Shadow entry is matched with one Top entry by a shared entity surface form. To make sure that the main challenge is entity disambiguation and not candidate generation, we excluded all entries for which the candidate generation module of REL failed to retrieve the correct entity. The remaining dataset contains 491 Shadow and 614 Top entries.

Note that, by design, ShadowLink contains ground truth information for only one entity mention per entry. As our method relies on collective disambiguation, we extract all different entity mentions from every entry and use them in the inference – however, the evaluation is only performed on the "target mention" that has a ground truth label assigned to it. To test the collective disambiguation approach for multiple mentions within one document, as well as to make sure that our method performs competitively on standard data, we also evaluate NICE on the test subset of the AIDA-YAGO-CoNLL dataset (Hoffart et al., 2011a), the largest manually annotated ED evaluation set which consists of 388 news articles.

Baselines. We compare the NICE method with five popular ED models available in the GERBIL evaluation framework (Röder et al., 2018): AIDA (Hoffart et al., 2011a), DBpedia Spotlight (Mendes et al., 2011), AGDISTIS/MAG (Usbeck et al., 2014), Babelfy (Moro et al., 2014a) and WAT (Piccinno and Ferragina, 2014), as well as three novel models not yet available in GERBIL: REL (van Hulst et al., 2020), GENRE (De Cao et al., 2020), and NER4EL (Tedeschi et al., 2021a).

The newest of the systems under evaluation is NER4EL, a neural model that uses the information about entity types to make ED decisions based on semantic similarity between a candidate entity and its context. As NER4EL turns out to achieve high results on Shadow, we include it as the semantic similarity module in the combined NICE method. Combining NER4EL with the ICE algorithm allows to extract more information from the context than by using each of the two methods alone, as NER4EL does not include entity relatedness

Method	Shadow filtered	Top filtered	AIDA test
AIDA (Hoffart et al., 2011a)	45.4	65.0	81.8
DBpedia Spotlight (Mendes et al., 2011)	16.2	37.9	49.3
Babelfy (Moro et al., 2014a)	52.9	66.1	68.2
WAT (Piccinno and Ferragina, 2014)	32.9	59.5	60.7
AGDISTIS/MAG (Usbeck et al., 2014)	12.8	22.8	59.4
REL (van Hulst et al., 2020)	31.5	70.5	83.3
GENRE (De Cao et al., 2020)	33.8	54.2	**93.3**
NER4EL (Tedeschi et al., 2021a)	44.3	62.4	<u>92.5</u>
NER4EL + no candidate filtering	43.3	67.3	86.3
NER4EL + robust candidate filter	45.7	68.1	85.3
ICE + no candidate filtering	56.9	74.9	72.1
ICE + robust candidate filter	**59.9**	<u>76.1</u>	71.9
NICE	<u>58.3</u>	**77.2**	80.4

Table 4.7. Benchmark evaluation results in terms of micro-F_1. The top part shows the results of the baseline systems. The bottom and middle parts show the scores obtained by NICE and its ablations. We mark in **bold** the best scores and <u>underline</u> the second best.

information and ICE does not include semantic similarity.

4.2.8 Results and discussion

The results of our experiments are presented in Table 4.7. Note that the term "filtered" in the dataset description refers to filtering out the entries where the correct candidate entity was not retrieved by our candidate generation model, and not to the NER-based candidate filtering. We use the experimental data to answer the research questions introduced in Section 4.2.

RQ1: *Does focusing on context information improve ED performance on overshadowed entities?* From the evaluation results in Table 4.7 it can be seen that NICE outperforms all baseline systems on the Shadow subset by a large margin. Moreover, all variations of our method considered in the experiments achieve top results on Shadow, which shows that all ways of leveraging context information considered in our study lead to performance improvements on overshadowed entities.

RQ2: *Does focusing on context information instead of relying on mention-entity priors in ED allow to maintain competitive performance on more frequent entities?* Our experimental data shows that NICE outperforms all baselines on the Top subset of ShadowLink, where

every entry contains the most frequent entity from the corresponding entity space. This demonstrates that placing more impact on context information does not automatically decrease the performance on more frequent entities: on the contrary, it appears to be beneficial in the setup of using the short textual context of ShadowLink.

The results on the AIDA dataset, however, are different: while still outperforming 4 out of 8 baseline systems, NICE achieves results slightly lower than AIDA and REL and by a large margin lower than GENRE and NER4EL. There are different factors contributing to this performance drop. Firstly, it appears that the original NER classifier used for candidate filtering in NER4EL works much better on the AIDA dataset than our classifier, which has been trained to focus on overshadowed entities. Secondly, the input structure differed between the two dataset types: for Top and Shadow we extracted auxiliary mention spans from the context to perform collective disambiguation (as the ShadowLink dataset only contains one ground truth entity mention per entry), and for AIDA we only used the mention spans provided in the dataset. In the AIDA dataset, all the labeled mentions satisfy the strict definition of a named entity, while for ShadowLink the auxiliary mentions were noun phrases extracted using the TagMe API, which considers all Wikipedia anchors as entities. For example, in the sentence *"Michael Jordan published a paper"*, *paper* is not a named entity but a common concept that can be linked to Wikipedia and used for collective disambiguation. Interestingly, a similar idea is used by Babelfy (Moro et al., 2014a), which disambiguates named entities and common concepts simultaneously using graph information, achieving the second-highest score both on Top and Shadow datasets. This indicates that considering all linkable mentions within a short textual context is benificial for disambiguating overshadowed entities.

RQ3: *In what ways do the different aspects of context information contribute to ED performance on overshadowed entities?* NICE uses three aspects of context information: entity types, semantic similarity between a candidate entity and its mention in context, and graph-based relatedness between candidate entities. To estimate the impact of these aspects, we turn to the experimental data of two kinds: the evaluation results presented in Table 4.7 and the results achieved on the development set with different combinations of component weights, shown in Figure 4.4. From Table 4.7 it can be seen that using entity types is

beneficial both for candidate filtering and for enhancing the semantic similarity module: NER4EL achieves considerably high results on Shadow when used alone and demonstrates a further improvement when combined with the robust candidate filter. Figure 4.4 shows that the highest scores are achieved when the weight of graph-based entity coherence score is relatively high, with the best combination being $0.3score_{NER4EL} + 0.7score_{ICE}$. Moreover, the best results on the Shadow test set are achieved when using the ICE coherence score alone, without semantic similarity (middle part of Table 4.7). Thus, all aspects of context information are beneficial for disambiguating overshadowed entities, with entity coherence being the most important component.

4.3 Open-Source Data and Models

We hope that the findings discussed in the previous sections will provide a stepping stone for future studies on the interplay between Entity Disambiguation and Named Entity Recognition in the context of high-performance EL systems and disambiguation of tail entities, especially for scenarios in which only a small amount of labeled data is available. We release our artifacts – data, code and model checkpoints – on GitHub.[7].

4.4 Summary

In recent years, ED systems based on contextualized embeddings from pretrained language models have achieved unprecedented results. However, such systems require training on millions of labeled samples, making them practically inaccessible to broad audiences and users and severely hampering the development of a high-performance ED system. Additionally, despite achieving excellent results on common benchmarks, they still fall short in disambiguating less frequent or tail entities.

In this chapter, to fill these gaps, we presented various NER-based strategies which allow systems trained on limited amounts of data to narrow the performance gap with those systems trained on massive training corpora (Section 4.1). To this end, we first introduced a new fine-grained set of NER classes to better cluster entities and then used these classes to enhance a strong ED baseline with i) NER-enriched entity representations, ii) NER-enhanced

[7] https://github.com/Babelscape/ner4el

candidate selection, iii) NER-based negative sampling, and iv) NER-constrained decoding. Our results show how the integration of NER information can aid an ED system trained on less than 20K instances in narrowing the gap with systems trained on millions of samples.

Then, building on these findings, we introduced NICE, a new method based on three main components to tackle the issue of overshadowed entities (Section 4.2). Specifically, this methods consists of a NER-based candidate filtering module to reduce the complexity of the task, a collective disambiguation module that uses an unsupervised iterative algorithm to capture entity coherence, and a module that measures semantic similarity between an entity and its context using NER-enhanced word embeddings. Our experimental results show that NICE achieves substantial improvements on overshadowed entities compared to the baseline methods, while still performing competitively in a standard entity disambiguation setting. This demonstrates that encompassing explicit contextual features, such as entity types and entity coherence, improves ED performance on overshadowed entities.

Finally, we publicly released our code and software to encourage the development of new resource-constrained ED systems capable of disambiguating both popular and rare entities (cf. Section 4.3).

Chapter 5

Multilingual Relation Extraction

As discussed in Section 2.3.4, existing Relation Extraction (RE) datasets are often outdated, locked behind paywalls, suffer from design limitations, or are restricted to the English language. While some multilingual datasets, such as ACE05[1] and SMiLER (Seganti et al., 2021), do exist, they have significant limitations. ACE05, for instance, covers only six relation types and requires a paid license for access. SMiLER, though more recent and extensive in both size and relation type coverage, lacks human-annotated samples, making it unsuitable for reliable evaluation and for training End-to-End RE systems. The development and availability of large, high-quality resources are critical to enabling Large Language Models (LLMs) to be trained and evaluated on robust multilingual RE benchmarks.

To address these issues, in this chapter we start by describing our new methodology for the creation of a new resource for enabling the training and evaluation of multilingual Relation Extraction systems (Section 5.1), then we introduce mREBEL, our new model for the task (Section 5.2), and, finally, we evaluate the performance of mREBEL on our resource (Section 5.3).

5.1 Resource Creation

In this section, we present RED^{FM}, our supervised and multilingual dataset for Relation Extraction, and SRED^{FM}, a larger, silver-annotated dataset covering more languages and relation types. Specifically, the creation process consists of several steps: data collection

[1] https://catalog.ldc.upenn.edu/LDC2006T06

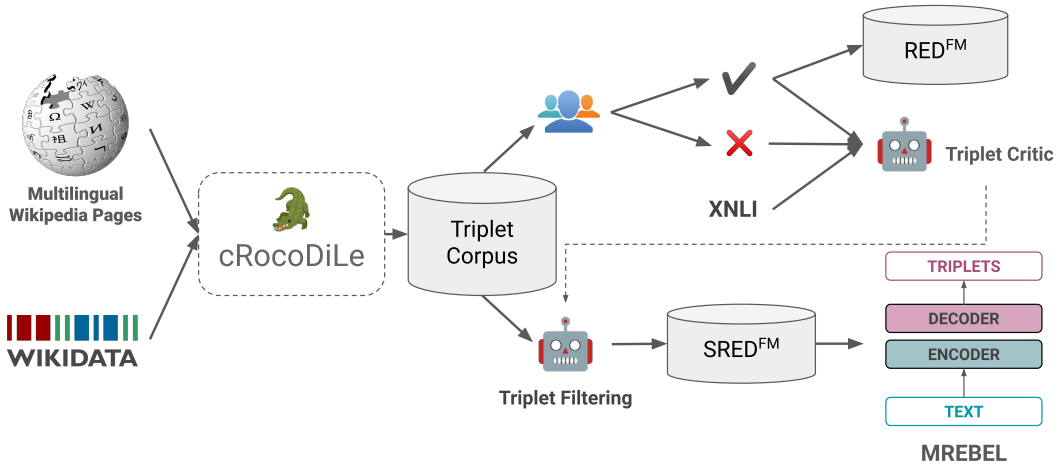

Figure 5.1. Our full pipeline for the creation of RED^{FM}, SRED^{FM} and mREBEL.

and processing (Section 5.1.1), manual annotation (Section 5.1.2), triplet filtering (Section 5.1.3) and entity typing (Section 5.1.4). Figure 5.1 shows an overview of this process.

5.1.1 Data Extraction

We base our dataset on Wikidata and Wikipedia, and expand cRocoDiLe, the data extraction pipeline from Huguet Cabot and Navigli (2021), to obtain a large collection of triplets in multiple languages. In the original work, they filtered triplets using NLI. For each triplet, they input the text containing both entities from the Wikipedia abstract, and the triplet in their surface forms, `subject` + `relation` + `object`, separated by the `<sep>` token. If the score was less than 0.75 for the entailment class, it was removed to ensure higher precision. In our work, we set a lower threshold, 0.1, since we further filter triplets using manual annotation or our Critic model (more details in Section 5.1.3).

We use the hyperlinks from Wikipedia abstracts, i.e. the content before the Table of Contents, as entity mentions and the relations in Wikidata between them. We run our pipeline in the following 18 languages: Arabic, Catalan, Chinese, Dutch, German, Greek, English, French, Hindi, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, Swedish, and Vietnamese. Then, we collapse inverse relations and keep the 400 most frequent ones. We highlight that some extracted relations are not necessarily entailed by the Wikipedia text; therefore, we apply a multilingual NLI system[2] to filter out those with a low

[2] xlm-roberta-large-xnli

entailment score (i.e. <0.1).

Despite using NLI techniques to filter out false positives, distant RE annotations still present noisy labels. This can result in unfair or misleading evaluations, as demonstrated for TACRED (Zhang et al., 2017), which had 23.9% wrongly annotated triplets that were later revised and corrected by Stoica et al. (2021). Moreover, our triplet corpus extraction pipeline relies on triplets in Wikidata, similar to T-REx (Elsahar et al., 2018). These latter showed how specific relation types, such as "capital", have a lower entailment score and may not be entailed by a given text even though the entities share the relation in Wikidata.

Given these challenges in distant RE annotation, manual filtering of a portion of the data is necessary to ensure high-quality, accurate annotations.

5.1.2 Manual Annotation

We manually filter a portion of the data to deal with false positives present in the dataset for a subset of languages (i.e. Arabic, Chinese, German, English, French, Italian, and Spanish) through crowdsourced annotation. Specifically:

1. We select a portion of our silver annotated data consisting of i) common Wikipedia pages across those languages and ii) a random sample with less frequent relations to balance the dataset.

2. We ask human annotators to validate each triplet. They are shown the context text with subject and object entities highlighted, and the possible relation between them from the silver extraction. They must answer whether the text conveys the necessary information to infer that the relationship between those two entities is true.

3. We annotate each triple three times using different annotators, obtaining an average inter-rater reliability (Krippendorff's alpha) across languages of $\alpha_\kappa = 0.73$.

4. We keep as true positives those relations with at least two annotators answering true. We consider the rest false positives.

Additionally, we reduce the coverage of the annotated data to the top 32 most frequent relation types. Annotators are presented with descriptions for each relation, which they can check at any time by hovering the label or opening the instructions. Figure 5.2 shows

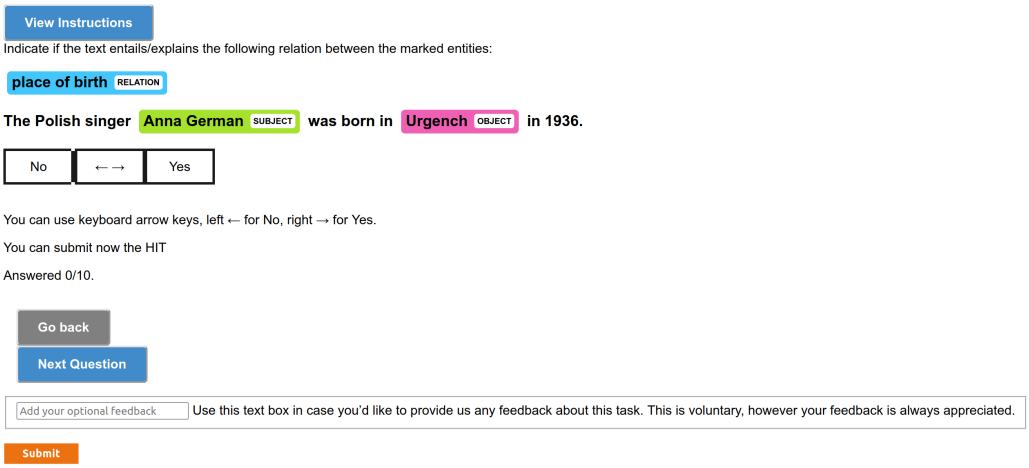

Figure 5.2. Annotation example from the MT platform. The interface was translated to each language by native speakers.

the annotation interface provided to the annotators, and the English descriptions of the 32 relation types are provided below:

- **located in the administrative territorial entity**: the item is located on the territory of the following administrative entity
- **country:** sovereign state of this item (not to be used for human beings)
- **instance of:** that class of which this subject is a particular example and member
- **shares border with:** countries or administrative subdivisions, of equal level, that this item borders, either by land or water
- **part of:** object of which the subject is a part
- **capital:** seat of government of a country, province, state or other type of administrative territorial entity
- **follows:** immediately prior item in a series of which the subject is a part
- **headquarters location:** city, where an organization's headquarters is or has been situated
- **located in or next to body of water:** sea, lake, river or stream
- **sport:** sport that the subject participates or participated in or is associated with

- **subsidiary:** subsidiary of a company or organization; generally a fully owned separate corporation
- **member of:** organization, club or musical group to which the subject belongs
- **owned by:** owner of the subject
- **manufacturer:** manufacturer or producer of this product
- **genre:** creative work's genre or an artist's field of work (P101)
- **located on terrain feature:** located on the specified landform
- **child:** subject has object as child
- **author:** main creator(s) of a written work (use on works, not humans); use P2093 when Wikidata item is unknown or does not exist
- **named after:** entity or event that inspired the subject's name, or namesake (in at least one language)
- **country of origin:** country of origin of this item (creative work, food, product, etc.)
- **replaces:** person, state or item replaced
- **inception:** date or point in time when the subject came into existence as defined
- **cast member:** actor in the subject production
- **subclass of:** next higher class or type; all instances of these items are instances of those items; this item is a class (subset) of that item
- **league:** league in which team or player plays or has played in
- **developer:** organization or person that developed the item
- **location:** location of the object, structure or event
- **occupation:** occupation of a person
- **spouse:** the subject has the object as their spouse (husband, wife, partner, etc.)

	ar	de	fr	en	es	it	zh
Annotators	3	9	12	9	10	13	7
α_κ	0.76	0.80	0.77	0.72	0.61	0.73	0.70
Filtered (%)	6.79	5.47	4.40	8.54	12.53	8.93	12.11

Table 5.1. Number of annotators, their agreement based on Krippendorff's alpha, and the percentage of filtered triplets for each language.

- **characters:** characters which appear in this item (like plays, operas, operettas, books, comics, films, TV series, video games)
- **notable work:** notable scientific, artistic or literary work, or other work of significance among subject's works
- **place of birth:** most specific known (e.g. city instead of country, or hospital instead of city) birth location of a person, animal or fictional character
- **mouth of the watercourse:** the body of water to which the watercourse drains
- **country of citizenship:** the object is a country that recognizes the subject as its citizen
- **founded by:** founder or co-founder of this organization, religion or place
- **director:** director(s) of film, TV-series, stageplay, video game or similar
- **sibling:** the subject and the object have the same parents (brother, sister, etc.)
- **participant:** person, group of people or organization (object) that actively takes/took part in an event or process (subject)

We employ Mechanical Turk for our annotation task and each annotator is paid 0.1$ for every ten instances annotated, constituting 1 HIT; an average of $10 hourly rate. We restrict annotators to countries where each of the languages is spoken, plus the USA. We manually screen annotators in each language separately by having them annotate a small sample of fewer than 10 HITs, and allowing only those who correctly performed the task to annotate the final corpus. From Table 5.1, we see that around 8% of annotated triplets, on average, were labeled as non-entailed by the context provided, albeit the percentage varies across languages. For instance, Spanish had a lower agreement across annotators and a higher number of filtered instances. Finally, in Table 5.2, we provide data statistics for the resulting dataset, namely RED$^{\text{FM}}$.

	Train						Validation								Test							
	de	en	es	fr	it	Total	ar	de	en	es	fr	it	zh	Total	ar	de	en	es	fr	it	zh	Total
location	896	1071	542	602	528	3639	57	53	99	57	54	55	77	452	73	55	76	61	79	62	100	506
instance of	639	1025	511	603	589	3367	109	74	116	57	78	105	48	587	80	66	116	64	86	115	41	568
country	865	535	646	703	548	3297	136	105	70	79	119	166	104	779	105	127	42	104	142	167	92	779
part of	301	668	353	393	248	1963	35	29	61	39	57	34	42	297	23	23	88	31	29	30	39	263
follows	166	301	210	234	294	1205	52	34	47	27	58	48	35	301	36	16	47	37	35	54	21	246
manufacturer	258	307	218	201	207	1191	92	73	59	42	72	49	61	448	51	51	77	35	74	49	30	367
sport	203	288	172	268	249	1180	45	51	101	41	75	103	11	427	43	39	131	39	61	100	6	419
owned by	199	398	170	198	159	1124	38	28	55	24	33	31	30	239	33	30	49	22	37	28	21	220
located in or next to body of water	169	326	252	159	100	1006	17	4	24	7	10	13	23	98	23	15	16	10	12	2	26	104
member of	125	347	163	159	129	923	19	11	20	16	13	7	19	105	15	20	49	18	23	12	11	148
genre	183	272	145	187	128	915	23	33	59	29	34	72	14	264	28	34	67	44	45	68	11	297
author	150	209	120	139	127	745	29	27	51	23	67	48	28	273	35	58	46	41	44	29	14	267
headquarters location	146	226	110	137	98	717	20	28	28	11	13	17	15	132	9	19	35	9	26	21	17	136
capital	112	191	105	109	104	621	7	9	7	4	9	2	12	50	12	8	4	2	6	3	14	49
child	107	147	104	134	102	594	23	12	16	9	14	12	25	111	15	23	15	28	36	15	33	165
notable work	86	204	61	137	97	585	21	14	47	10	37	34	19	182	19	20	38	17	31	18	4	147
mouth of the watercourse	98	156	205	47	72	578	2	1	15	4	2	6	3	33	4	4	6	3	6	2	3	28
inception	65	335	90	13	69	572	55	10	46	10	3	5	21	150	48	21	36	9	0	9	13	136
named after	139	192	81	80	66	558	10	11	6	4	5	5	7	48	8	6	3	9	10	6	9	51
occupation	172	40	105	139	70	526	34	12	12	13	49	89	12	221	17	18	11	14	43	87	4	194
shares border with	77	56	153	117	63	466	3	2	3	1	4	1	3	17	6	4	4	1	7	1	5	28
participant	48	192	86	66	51	443	6	0	16	5	8	17	6	58	7	2	20	4	13	4	14	64
country of citizenship	111	90	105	71	64	441	23	19	13	18	20	111	29	233	19	26	7	21	18	108	44	243
founded by	81	134	58	97	68	438	16	13	16	2	10	6	7	70	9	4	17	7	10	8	3	58
place of birth	128	94	90	101	23	436	14	18	7	12	12	1	4	68	8	20	3	10	11	2	6	60
replaces	78	118	55	70	70	391	5	4	12	1	6	7	16	51	3	9	11	7	6	7	11	54
cast member	53	145	95	12	43	348	35	38	73	20	20	18	4	208	60	24	102	26	12	16	29	269
league	81	125	42	38	47	333	24	24	33	18	8	24	20	151	35	22	49	18	8	11	12	155
characters	39	113	38	64	46	300	15	9	19	15	16	9	4	87	8	11	24	10	13	13	9	88
director	66	69	37	66	58	296	11	22	16	6	35	26	4	120	20	19	30	12	22	32	7	142
spouse	32	76	49	57	41	255	3	8	8	6	13	7	6	51	8	5	9	11	17	5	9	64
sibling	36	54	23	51	39	203	3	1	5	1	2	1	12	25	4	12	7	9	13	2	5	52
total	5909	8504	5194	5452	4597	29656	982	777	1160	611	956	1129	721	6336	864	811	1235	733	975	1086	663	6367

Table 5.2. Breakdown for RED^{FM}.

5.1.3 Triplet Critic

Our manual annotation procedure (Section 5.1.2) filtered a portion of the silver data in order to have a higher-quality subset on which to train and evaluate our models. However, by removing the negative triplets we disregard valuable information that can be used to improve the quality of the remaining annotations, i.e. all those not validated by humans. Inspired by West et al. (2021), who trained *critics* based on human annotations on commonsense triplets, we use our annotated triplets, both true and false positives with their contexts, to train a Triplet Critic. Specifically, given a textual context c and an annotated triplet t that may appear in c, we train a cross-encoder $T(c, t)$ to predict whether c, the premise, entails t, the hypothesis. This setup was inspired by NLI and results in training examples such as:

Premise	Hypothesis	Label
Telefe (acronym for Televisión Federal) is a television station located in Buenos Aires, Argentina.	Telefe instance of television station	True
	Buenos Aires country Argentina	True
	Argentina capital Buenos Aires	False

Once T is trained, we can use it on our silver data to filter out other false positives, i.e., triplets that, albeit present in Wikidata for two entities within the context, are not entailed by that context.

We test our approach by training T in English, French, Italian, Spanish and German, namely European languages with shared families (Romance and Germanic), and testing on Arabic and Chinese. This setup will test the zero-shot multilingual capabilities on unseen languages in order to determine whether the Critic can be applied to any language. We base our Triplet Critic on DeBERTaV3 (He et al., 2021) with a classification head on top of the [CLS] token that produces a binary prediction, trained using a Cross-Entropy loss criterion. Furthermore, since the task is similar to and inspired by NLI, we explore a multi-task approach using the XNLI dataset (Conneau et al., 2018), aiming at improving cross-lingual performance. To this end, we add an additional linear layer at the end of the model for NLI that projects the output layer to the three possible predictions (neutral, contradiction, entailment), again using a Cross-Entropy loss.

Table 5.3 shows the Triplet Critic results when trained under different setups. We see how the use of our data dramatically improves upon the XNLI baseline, especially in

		All True	XNLI	Ours	+XNLI$^-$	+XNLI
Arabic	**R.**	100.0	76.6	96.9	**98.6**	98.5
	P.	93.2	**94.9**	**94.9**	94.5	94.5
	F1	96.5	84.8	95.9	**96.5**	**96.5**
	Acc.	93.2	74.3	92.4	**93.3**	**93.3**
Chinese	**R.**	100.0	82.3	98.7	98.8	**99.6**
	P.	87.9	89.9	**90.1**	**90.1**	89.7
	F1	93.5	85.9	94.2	94.2	**94.4**
	Acc.	87.9	76.3	89.3	89.4	**89.5**
Total	**R.**	100.0	79.1	97.7	98.7	**98.9**
	P.	90.8	92.6	**92.8**	92.6	92.4
	F1	95.2	85.3	95.2	**95.5**	**95.5**
	Acc.	90.8	75.2	91.0	**91.6**	**91.6**

Table 5.3. Performance for the Triplet Critic. **All True** shows baseline when all triplets are marked as True. **XNLI** uses the entailment prediction of a model trained solely on XNLI. **Ours** is trained solely on our annotated data without ar/zh. Last two columns show the multi-task approach with (**+XNLI**) and without ar/zh (**+XNLI$^-$**) data from XNLI.

terms of accuracy. While we are primarily interested in precision so as to guarantee that triplets are valid, a low accuracy would lead to missing annotations, which we also want to avoid. Additionally, when our Triplet Critic is trained simultaneously on our data and XNLI, even if no Arabic or Chinese data is used (i.e. +XNLI$^-$), performance further improves: the system achieves an average 92.6% precision, on par with using only our data, but sees a point increase in recall, which is remarkable, taking into account the high class inbalance. In contrast, when XNLI data from those languages is added (i.e. +XNLI), we observe a small trade-off between precision and recall.

Overall, these zero-shot results certainly legitimize the use of our Triplet Critic to refine our silver data for unseen languages, and suggest even more promising benefits for seen languages. Furthermore, the Triplet Critic serves as a feedback on the consistency of human annotations since the models have successfully learned from them.

5.1.4 Entity Typing

In RE datasets, the entity types are commonly included in the triplets (Riedel et al., 2010) and, therefore, are taken into account under the strict evaluation, where a triplet is only considered correctly extracted when entity boundaries, entity types, and relation type are all

predicted, versus the boundaries evaluation, where only entity boundaries and relation type are taken into account (Taillé et al., 2020). This Section describes the procedure through which we automatically label entities with their types.

Following the procedure described in Section 3.1.2, we start by mapping entities in Wikipedia to BabelNet (Navigli and Ponzetto, 2012b) synsets by exploiting the one-to-one linkage between them. Then, we annotate synsets by applying the knowledge-based semantic classifier introduced in Section 3.1.3, which exploits the relational information in BabelNet such as hypernymy and hyponymy relations. This procedure yields ∼7.5M entities labeled with an entity type. However, since the annotations are automatically derived and prone to errors, we devise a new strategy to improve their quality. Specifically, we design a Transformer-based classifier that takes a synset and returns its NER category. More formally, let us define the functions $L(s)$ and $D(s)$ that output the main lemma and the textual description of a synset s, respectively. Then, given a synset s and the above-defined functions, we provide the string $I(s, D, L)$ = [CLS] $L(s)$ [SEP] $D(s)$ [SEP] as input to the classifier that predicts a label e ∈ E[3]. The tagset E is obtained by refining the categorization of named entities introduced in Section 4.1.2 based on the ability of automated systems to distinguish NER categories and on the frequency of these categories in Wikipedia articles (cf. Section 3.3.4). To train the classifier, we construct a dataset by selecting a high-quality subset from the 7.5M automatically-produced annotations by taking only synsets with a maximum distance equal to 1 from one of the 40k synsets in WordNet (Miller, 1995), this latter being a manually-curated subset of BabelNet. By doing this, we obtain a set consisting of 1.2M high-quality annotations that we split into 80% for training and 20% for validation, and convert these to the above-specified $I(s, D, L)$ format.

Finally, we use the trained classifier to confirm or replace the previous 7.5M annotations, resulting in 6.2M (82.4%) confirmations and 1.3M (17.6%) changes, and employ it to label new Wikidata instances as well, thus obtaining a final mapping consisting of ∼13M Wikidata entries annotated with their entity types. By manually inspecting a sample of 100 changes, we observed that our NER classifier was right 68% of the time, wrong 17%, while the remaining 15% of the time both annotations were wrong, providing an improvement of 51% over 1.3M changes. We highlight that 68% is not the accuracy of our classifier as it

[3]E = {location, person, number, time, organization, date, event, celestial body, media, disease, concept, miscellaneous and unknown}

is computed on 100 items where there is a disagreement between the original annotations produced by WikiNEuRal (Section 3.1) – the current state of the art in entity typing – and the annotations produced by our model. Indeed, an accuracy of 68% on this subset means that our classifier corrected most of the instances that were previously mistaken by WikiNEuRal. For completeness, we report that when the two systems agree, i.e. 82% of the time, they are correct in 98% of these cases, as measured on another subset of 100 instances.

5.1.5 SRED^{FM}

The current datasets for Relation Extraction often lack complete coverage of relations. The SMiLER dataset (Seganti et al., 2021) only annotates one triplet per example, resulting in a limited understanding of the relationships therein. For instance, in the example "*Fredrik Hermansson (born 18 July 1976) is a Swedish musician. He was a keyboardist and backing vocalist in the Swedish progressive rock band Pain of Salvation.*", the triplet (Fredrik Hermansson, has-genre, progressive rock) is annotated, but other triplets such as (Fredrik Hermansson, has-occupation, musician) and (Fredrik Hermansson, has-nationality, Swedish) are also valid.

Another common issue with RE datasets is the high class imbalance, particularly for distantly annotated datasets. This is often due to the skewed distributions that are intrinsic in knowledge bases such as Wikidata, which are often used to construct RE resources. This leads to low coverage for lower frequency classes, as seen in Figure 5.3 for certain languages in the SMiLER dataset. In DocRED (Yao et al., 2019), another distantly annotated dataset, location-based relations constitute over 50% of instances.

Fully human-annotated datasets that overcome these limitations are scarce and often not widely accessible. Additionally, they often cover a narrow set of languages and relations (Table 5.4). To address these issues, we introduce Silver RED^{FM}. SRED^{FM} is a large, multilingual RE dataset that contains more than 45M triplets and covers 400 relation types and 18 languages. It is created using the data extraction procedure described in Section 5.1.1 and the Triplet Critic introduced in Section 5.1.3. SRED^{FM} overcomes some of the previous shortcomings of current datasets by providing a higher coverage of annotation and more evenly distributed classes. Using the same example sentence, in SRED^{FM}, the following triplets are annotated: (Fredrik Hermansson, country of citizenship, Swedish),

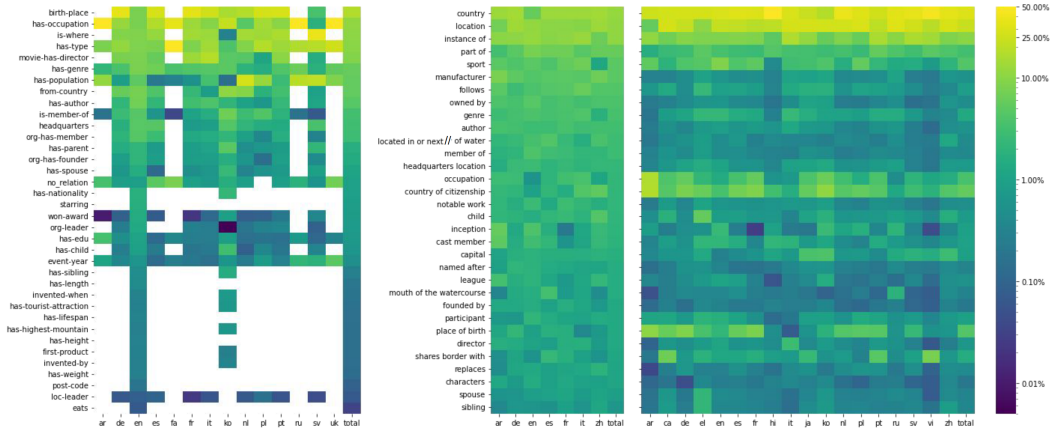

Figure 5.3. Comparison of relation type distribution by percentage between SMiLER (left), RED^{FM} and SRED^{FM} (right). Best seen in color.

(Fredrik Hermansson, occupation, musician), (Fredrik Hermansson, date of birth, 18 July 1976), and (Fredrik Hermansson, member of, Pain of Salvation). Regarding class balance, in the English portion of SRED^{FM} location-based relations make up less than 37%. Hence our datasets have more evenly distributed classes. Figure 5.3 (right) shows the distribution for the top 32 of the 400 relation types in SRED^{FM}.

Additionally, we provide a pipeline that enables the automatic creation of an RE dataset in any language. So, even though we release the SRED^{FM} dataset as described in this section (i.e. covering 18 different languages), we encourage the expansion to other languages by using our pipeline available here. In summary, SRED^{FM} is a large, multilingual dataset that addresses the shortcomings of current datasets by providing a higher coverage of annotation and more evenly distributed classes.

5.2 mREBEL

In this section, we present our system, mREBEL (Multilingual Relation Extraction By End-to-end Language generation), which is a multilingual relation extraction model pre-trained on SRED^{FM}. It is a multilingual extension of the REBEL model introduced in Huguet Cabot and Navigli (2021), which uses a seq2seq architecture to convert relations into text sequences that can be decoded by the model. We convert triplets into text sequences and pre-train our model using mBART-50 (Tang et al., 2021). To support multiple languages, we prepend the input text with a special language token (i.e. `en_XX`). Additionally, we include relation

	Human Annotated				Distant supervision			
	ACE05	CONLL04	DocRED	RED^{FM}	DocRED	NYT	SMILER	SRED^{FM}
Docs.	1.6K	1.4K	5K	15.4K	100K	66.2K	1.1M	12.3M
Sents.	30.9K	1.4K	-	43.7K	-	66.2K	1.1M	46.6M
Rels.	6	5	96	32	96	24	36	400
Ents.	5	4	6	13	6	3	-	13
AR	4.7K	-	-	1.8K	-	-	9K	3.3M
CA	-	-	-	-	-	-	-	1.7M
DE	-	-	-	7.5K	-	-	53K	4.8M
EL	-	-	-	-	-	-	-	325K
EN	8.7K	6.8K	58.6K	10.9K	1M	111K	748K	12.4M
ES	-	-	-	6.5K	-	-	12K	4.2M
FA	-	-	-	-	-	-	3K	-
FR	-	-	-	7.4K	-	-	62K	4.2M
HI	-	-	-	-	-	-	-	301K
IT	-	-	-	6.8K	-	-	76K	2M
JA	-	-	-	-	-	-	-	3.3M
KO	-	-	-	-	-	-	20K	1M
NL	-	-	-	-	-	-	40K	3M
PL	-	-	-	-	-	-	17K	3.7M
PT	-	-	-	-	-	-	45K	2.7M
RU	-	-	-	-	-	-	7K	1.6M
SV	-	-	-	-	-	-	5K	7.2M
UK	-	-	-	-	-	-	1K	-
VI	-	-	-	-	-	-	-	1.4M
ZH	9.3K	-	-	1.4K	-	-	-	3M

Table 5.4. Number of relation types (Rels.), entity types (Ents.) and annotated triplets in RE resources.

classification (RC) in the pre-training phase of mREBEL. Specifically, for 5% of the training data, we select a random triplet, mark the subject and object entities in the input text, and use a special token `<relation>` to indicate to the model that only one triplet needs to be decoded. Finally, to promote cross-lingual transfer, we use the English names for the relation types when decoding the triplets. Table 5.5 shows how instances from SRED^{FM} are used to train mREBEL. We train three versions of mREBEL:

1. mREBEL$^{T}_{400}$, trained on 400 relation types, including entity types;

2. mREBEL$^{T}_{32}$, fine-tuned on top of the previous one but including only the 32 relation types from RED^{FM};

3. mREBEL$^{T}_{B400}$, trained on top of M2M100 (Fan et al., 2020).

For (1) and (2) we also train their untyped versions, mREBEL$_{400}$ and mREBEL$_{32}$.

		Input text	Triplets	Linearized Triplets
Classification	nl	nl_XX # Mumbai Mirror # is een Engelstalige tabloid, die verschijnt in de Indiase stad Mumbai. Het is hier met een oplage van zo'n 700.000 exemplaren de belangrijkste krant. Het dagblad verscheen voor het eerst op @ 30 mei 2005 @,	(Mumbai Mirror, inception, 30 mei 2005)	tp_XX<relation>Mumbai Mirror <media>30 mei 2005 <date>inception
Extraction	ca	ca_XX Can Verboom és una masia amb elements gòtics i barrocs de Premià de Dalt (Maresme) protegida com a bé cultural d'interès local.	(Can Verboom, located in the administrative territorial entity, Premià de Dalt) (Can Verboom, heritage designation, bé cultural d'interès local)	tp_XX<triplet>Can Verboom <loc> Premià de Dalt <loc>located in the administrative territorial entity <loc> bé cultural d'interès local <loc> heritage designation

Table 5.5. Examples from SRED^{FM} with their triplet linearization used to train mREBEL.

	Learning Rate	Warm-up	Batch size	Max Steps
mREBEL$_{400}$	10^{-5}	5000 steps	32	1.6M
mREBEL$_{32}$	10^{-5}	5000 steps	32	+ 264K
mREBEL$_{B400}$	5×10^{-5}	5000 steps	32	1M
SMILER	5×10^{-5}	3000 steps	32	+ 10K
RED^{FM}	10^{-5}	1000 steps	32	+ 10K

Table 5.6. Hyperparameters for the different datasets. Top half shows used the values used in the pretraining phase, while the bottom part shows those used during fine-tuning.

5.3 Experimental Setup

We evaluate mREBEL and its variants on our own datasets, i.e. RED^{FM} and SRED^{FM}, and on SMiLER (Seganti et al., 2021). Unless stated otherwise, we train on the training sets of all languages simultaneously and apply early stopping based on the Micro-F1 obtained on the overall validation set. We use the Adafactor optimizer and the same Cross-Entropy loss with teacher forcing from Huguet Cabot and Navigli (2021). Experiments were performed using a single NVIDIA 3090 GPU with 64GB of RAM and Intel® Core™ i9-10900KF CPU. The hyperparameters were manually tuned on the validation sets for each dataset, but mostly left at default values for mBART. The ones used for the final results can be found in Table 5.6.

Multilingual Relation Extraction We report the Micro-F1 score per language for both SMiLER and RED^{FM} test sets. When evaluating on SMiLER, we use mREBEL$_{400}$ variants as starting checkpoints and fine-tune them on SMiLER training sets. For RED^{FM}, instead, we include the mREBEL$_{32}$ model in our experiments as it was trained on the same set of relations. The inclusion of this model lets us analyze the impact of further fine-tuning on RED^{FM} gold data against the quality of our silver annotation process. As an extrinsic

	ar	de	en	es	fa	fr	it	ko	nl	pl	pt	ru	sv	uk	Avg.
Support	190	1049	731	225	54	1241	1501	228	791	343	880	131	92	20	
HERBERTa	49.0	63.0	77.0	46.0	72.0	58.0	69.0	51.0	61.0	50.0	62.0	30.0	59.0	25.0	54.7
mREBEL$^{T}_{400}$	**75.3**	63.1	**77.7**	**57.8**	69.2	64.3	**83.5**	**62.5**	72.8	76.1	69.8	71.0	76.1	**70.0**	70.5
mREBEL$_{400}$	73.7	**64.2**	77.7	54.7	**74.8**	**66.1**	82.9	61.7	**73.6**	**76.4**	**70.1**	**75.6**	**78.3**	65.0	**70.8**
mT5$_{BASE}$*	95.1	95.4	96.1	81.1	73.1	97.2	98.3	83.2	96.9	95.6	96.9	87.6	63.0	71.8	88.0
mT5$_{BASE}$(en)	94.0	94.9	-	91.7	91.1	96.0	97.5	78.2	97.5	93.3	95.2	93.8	97.8	94.7	93.5
mREBEL$^{T}_{B400}$	**99.5**	96.8	96.6	95.1	**100.0**	97.7	98.7	**94.7**	98.3	97.4	97.8	**100.0**	96.7	**100.0**	97.8
mREBEL$^{T}_{400}$	**99.5**	97.4	97.5	94.9	97.0	**97.8**	98.8	93.1	**98.7**	98.4	98.3	**100.0**	97.8	**100.0**	97.8
mREBEL$^{T}_{400}$*	**99.5**	96.9	97.5	93.5	99.0	**97.8**	**98.9**	94.1	98.0	98.3	97.7	98.8	**98.4**	97.4	97.6
mREBEL$_{400}$	**99.5**	**97.5**	**97.7**	**95.3**	**100.0**	97.6	**98.9**	**94.7**	98.6	**98.7**	**98.4**	99.2	97.8	**100.0**	**98.1**

Table 5.7. Results on SMiLER. Micro-F1 scores per language. Top half shows RE, bottom half RC. Selected top performing multilingual HERBERTa per language from (Seganti et al., 2021) and mT5$_{BASE}$ from Chen et al. (2022). Chen et al. (2022)(en) was trained on English data. * indicates separate training per language.

Model	Fine-tuning	ar	de	en	es	fr	it	zh	P	R	F1
mREBEL$^{T\dagger}_{32}$	✘	43.4	53.9	50.0	46.9	48.3	56.4	38.5	43.1	56.1	48.7
mREBEL$^{T}_{32}$	✘	**45.5**	57.8	53.3	**51.4**	**52.5**	**57.8**	**41.8**	47.6	**57.3**	**52.0**
mBART	✔	16.1	39.6	32.6	27.3	28.5	30.0	0.0	26.8	29.0	27.9
mREBEL$^{T}_{B400}$	✔	39.9	50.3	49.0	41.1	41.7	50.6	38.0	45.0	45.1	45.1
mREBEL$_{400}$	✔	33.2	50.0	42.2	38.4	40.3	49.2	30.7	40.9	41.7	41.3
mREBEL$^{T}_{400}$	✔	39.3	52.8	49.5	45.9	46.8	54.7	35.2	47.5	47.0	47.2
mREBEL$^{T\dagger}_{32}$	✔	43.7	55.4	**54.0**	46.9	50.5	57.1	38.8	47.9	53.0	50.3
mREBEL$^{T}_{32}$	✔	43.8	**58.3**	53.7	50.1	51.8	**57.8**	41.3	**48.9**	54.7	51.6

Table 5.8. Results on RED$^{\text{FM}}$ test set. Micro-F1 scores per language. † indicates the Critic was not used to filter pre-training data. Fine-tuning indicates fine-tuning on RED$^{\text{FM}}$ training set.

evaluation of our Triplet Critic model from Section 5.1.3, we train a version of mREBEL$^{T}_{32}$ without filtering triplets.

Multilingual Relation Classification Even though Seganti et al. (2021) introduced SMiLER as a RE dataset, each sentence contains just one annotated triplet and includes the "no relation" class as part of its annotation scheme. Therefore, it is better approached as an RC task and it is more akin to a dataset like TACRED. For instance, Chen et al. (2022) use it as an RC dataset with an array of prompt-based approaches, and we compare our approach with theirs for RC.

5.4 Results

Multilingual Relation Extraction. First, in Table 5.7 we show how our system performs compared to HERBERTa, the system proposed by Seganti et al. (2021) for SMiLER, using

	ar	ca	de	el	en	es	fr	hi	it	ja	ko	nl	pl	pt	ru	sv	vi	zh	all
Sents. (K)	4.5	2.7	5.7	0.9	6.9	8.8	4.0	0.4	2.2	3.4	0.8	2.8	6.5	4.5	2.3	7.6	2.0	2.7	68.6
Trip. (K)	27.9	15.5	32.1	4.9	40.0	54.9	26.1	2.8	12.8	29.4	4.4	16.4	36.6	27.5	12.9	44.9	10.5	27.4	427.0
Precision	61.1	57.0	58.5	46.1	62.3	60.6	60.0	50.2	57.7	48.6	48.5	57.5	64.6	60.0	53.4	63.2	59.0	41.0	58.5
Recall	48.7	45.9	44.5	34.1	47.2	51.2	45.9	29.6	44.9	44.2	37.7	49.0	55.4	48.9	38.2	55.7	45.3	36.6	47.9
Micro-F1	54.2	50.8	50.6	39.2	53.7	55.5	52.0	37.2	50.5	46.3	42.5	52.9	59.6	53.9	44.6	59.2	51.2	38.7	52.7
Macro-F1	24.0	24.8	29.3	12.5	36.3	30.5	29.1	8.3	27.9	25.1	17.3	27.6	30.5	29.5	23.5	30.1	19.5	20.3	24.8

Table 5.9. Results for SRED$^{\text{FM}}$ test set with mREBEL$^{T}_{400}$ on 400 relation types.

their best-performing setup for each language. We consider this dataset better suited for RC. However, as it originally reports on RE, we demonstrate how our system can perform better when pretrained on SRED$^{\text{FM}}$. In particular, mREBEL$^{T}_{400}$ provides an improvement of about 15 Micro-F1 points compared to HERBERTa. Additionally, as SMiLER does not include entity types, we observe that mREBEL$_{400}$ performs marginally better than mREBEL$^{T}_{400}$.

Table 5.8, instead, shows the results on RED$^{\text{FM}}$, compared against an mBART baseline. Specifically, we analyze model performance when fine-tuning is, or is not, performed on the train set of RED$^{\text{FM}}$. While performances vary across languages, the best overall Micro-F1 (52.0) is obtained when training on SRED$^{\text{FM}}$, mREBEL$^{T}_{32}$, without further fine-tuning. This confirms that our silver annotation procedure produces high-quality data, as there is no need for further tuning with RED$^{\text{FM}}$, which achieved 51.6. We also see how filtering by the Triplet Critic was crucial: when removed, performance dropped by more than 3 points.

Training on 400 relation types does lead to lower results, since there is a mismatch between the two stages of training. However, mREBEL$^{T}_{400}$ showed decent performance on SRED$^{\text{FM}}$ as shown in Table 5.9. This provides the first RE system to competitively extract up to 400 relation types in multiple languages. Additionally, in Figure 5.4 we provide a heatmap that shows the scores attained by mREBEL$^{T}_{32}$ (without fine-tuning) on each of the 32 relations covered by RED$^{\text{FM}}$, and for each of its 7 languages. Similarly, in Figure 5.5, we report the scores obtained by the fine-tuned version of mREBEL$^{T}_{32}$. By looking at these two heatmaps, it is easy to identify our model's strengths and weaknesses across relations and languages. We can see how relations such as *named after* or *shares border with* had low scores, probably due to their lower frequency at evaluation time, where a few errors lead to a low score. On the other hand, domain-specific relations such as *cast member*, *league* or *author* show a strong performance on most languages.

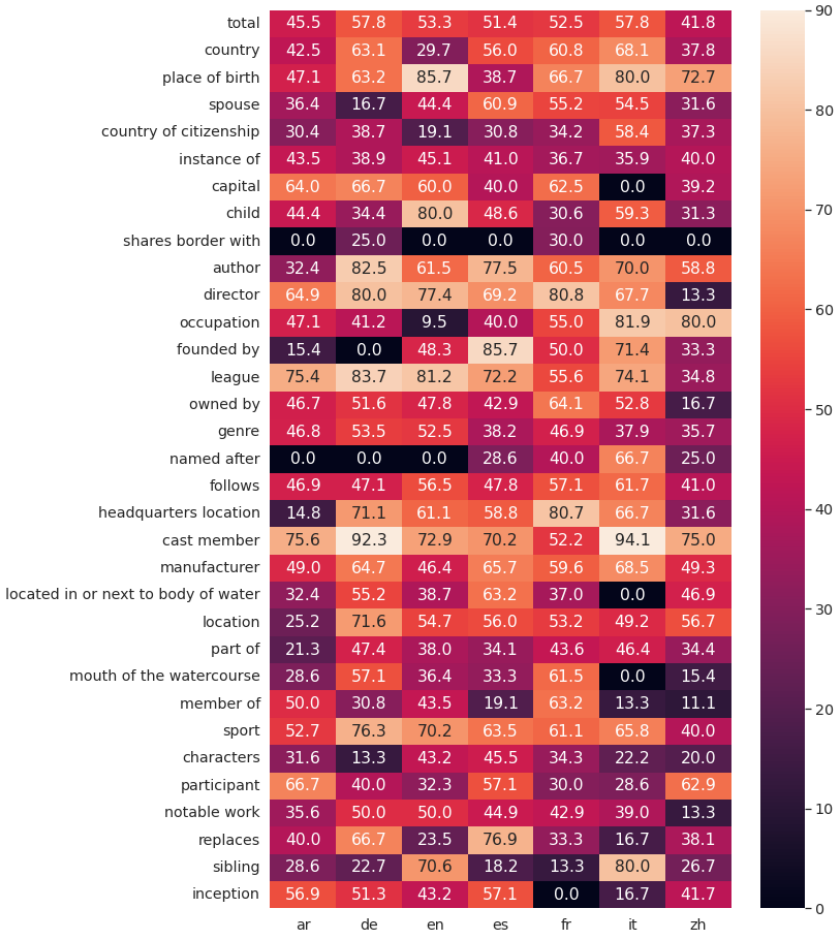

Figure 5.4. Results for mREBEL$^{T}_{32}$ on RED$^{\text{FM}}$ without fine-tuning.

Multilingual Relation Classification. From the bottom half of Table 5.7, we can observe how our mREBEL models consistently outperform competitive baselines, i.e. mT5$_{BASE}$* and mT5$_{BASE}$(en), by a large margin on all tested languages.

5.5 Error Analysis

We performed an error analysis with mREBEL$^{T}_{32}$ to understand the sources of error when training on RED$^{\text{FM}}$. Our study revealed that 27.8% of errors in the test set can be attributed to specific reasons.

First, there were discrepancies between predicted entity types and annotations (7.2%). These errors may have arisen from the automatic nature of the typing annotation, errors by the system or ambiguity in some cases, such as fictional characters, which can be

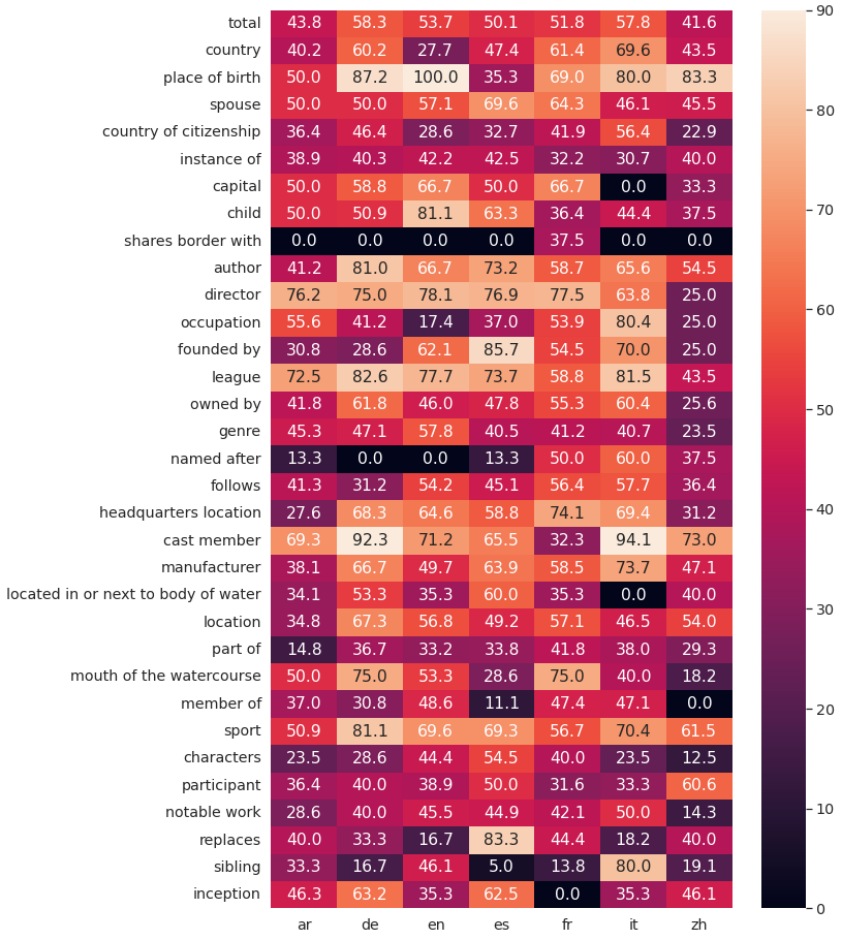

Figure 5.5. Results for mREBEL$^{T}_{32}$ on RED$^{\text{FM}}$ with fine-tuning.

considered either PERSON or MEDIA. Additionally, a portion of errors (8.1%) resulted from mismatches between the predicted and annotated spans for each entity, which may also be ambiguous (see the Span overlap example in Table 5.10). Another 10.6% of errors were caused by either the subject or object entity being completely misaligned with the annotation. We identify some of these as co-reference errors, such as the Subject example in Table 5.10. Evaluation for RE systems often ignores other mentions of an entity. We believe co-reference resolution has not been properly explored within RE evaluation and this may open interesting opportunities for future work.

Finally, it is worth noting that only 1.8% of errors were due to the wrong relation type being predicted between entities. We consider this to be a strong indicator of the quality of annotated relations between entities. However, we also observed that 72.2% of the errors were caused by incorrect predictions or missing annotations, highlighting the main

Error Type	Count	Example
Entity type	262 (7.2%)	The Peugeot 408 (CONCEPT, MISC) is a small family car produced by Peugeot (ORG) . [manufacturer]
Span underlap	157 (4.3%)	La Changhe Aircraft Industries Corporation (ORG) è un'azienda cinese, specializzata nella produzione di elicotteri (CONCEPT) . [manufacturer]
Span overlap	138 (3.8%)	La caja de Pandora (MEDIA), película muda (CONCEPT) alemana, dirigida por G. W. Pabst. [genre]
Subject	269 (7.4%)	L'œuvre contient notamment " Le Pré Béjine (MEDIA) ", dont le titre a été repris par Serguei Eisenstein (PERSON) pour un film inachevé (MEDIA) . [director] [director]
Object	117 (3.2%)	OpenSolaris (MISC) ist ein auf Unix (CONCEPT) basierendes Betriebssystem (CONCEPT) für die Plattformen PC, SPARC und andere. [instance of] [instance of]
Relation	66 (1.8%)	The Red Hot Chili Peppers (ORG) were formed in Los Angeles by Kiedis, Flea, guitarist Hillel Slovak (PER) and drummer Jack Irons. [member of] [part of]
Other	2606 (72.2%)	René L'Hermitte (PER) est un journaliste (CONCEPT) et un professeur des universités français, membre de la Résistance et du Parti communiste [occupation]

Table 5.10. Types of error encountered in RED^{FM}. Orange shows the mismatched prediction, and green is the annotated counterpart. Best seen in color.

shortcoming of our annotation procedure. Our approach is based on annotated hyperlinks in Wikipedia and relations in Wikidata, which can result in recall issues where entities in the text are not identified as hyperlinks or relational facts are not present in Wikidata.

5.6 Open Source Data and Models

Following the methodologies described above, we created extensive, high-quality datasets for multilingual RE, encompassing several languages (e.g., en, es, fr, it, zh) and a variety of domains, resulting in millions of annotated sentence pairs with relation labels. Therefore, to advance research in multilingual RE, we publicly release two large-scale datasets, SREDFM[4] and REDFM[5], on Hugging Face. These datasets are designed to support robust model training and evaluation across diverse languages and relation types, respectively.

In addition to datasets, we provide a new family of pre-trained multilingual language models to further stimulate the adoption and experimentation with multilingual RE. Specifically, we fine-tune a base version of a multilingual sequence-to-sequence model (mBART) on the SREDFM dataset and release it as mREBEL-base[6]. Moreover, for enhanced extrac-

[4] https://huggingface.co/datasets/Babelscape/SREDFM
[5] https://huggingface.co/datasets/Babelscape/REDFM
[6] https://huggingface.co/Babelscape/mrebel-base

tion capabilities, we offer two larger variants, mREBEL large[7] and mREBEL-large-32[8], with the latter being trained to predict only the 32 most popular relationship types. These models aim at balancing parameter efficiency with high performance, enabling accurate multilingual relation extraction across varied datasets and languages.

Further technical documentation and software details are available in the official REBEL GitHub repository[9], which hosts the code and instructions for using both the datasets and models effectively in multilingual NLP tasks.

5.7 Summary

In this chapter, we have addressed some of the key issues facing current multilingual relation extraction datasets by presenting two new resources: SRED^{FM} and RED^{FM} (Section 5.1). SRED^{FM} is an automatically annotated dataset covering 18 languages, 400 relation types, 13 entity types, and more than 40 million triplet instances, while RED^{FM} is a smaller, humanly-revised dataset for seven languages. We improved the quality of the entity type annotations in these datasets by using a Transformer-based NER classifier. We also introduced the Triplet Critic (Section 5.1.3), a cross-encoder that is trained on annotated data to predict whether a given context entails a triplet. We then presented mREBEL, the first multilingual end-to-end relation extraction system that extracts triplets, including entity types (Section 5.2), and demonstrated the utility of our new for training new, capable multilingual relation extraction models and evaluating them using our supervised data (Sections 5.3 and 5.4). Finally, we performed an extensive error analysis (Section 6.1.6) and publicly released our artifacts (Section 5.6), thus contributing to the development of better multilingual relation extraction systems and providing valuable resources for future research.

[7] https://huggingface.co/Babelscape/mrebel-large
[8] https://huggingface.co/Babelscape/mrebel-large-32
[9] https://github.com/babelscape/rebel

Chapter 6

Beyond Named Entities

6.1 Concepts

As we described in the previous chapters, the extraction of knowledge from unstructured texts is crucial for a wide range of applications, from Information Retrieval to Text Mining (Grishman, 2015). Within this context, *"nouns and noun phrases have a special status in describing the concepts that people are interested in searching for"* (Manning et al., 2008). Consequently, to work at their best, NLP systems should not only leverage the information about named entities, but also that concerning nominal concepts, as the nature of the two is strongly intertwined. However, although many tasks have been shown to benefit from the recognition of named entities (Mollá et al., 2006; Durrett and Klein, 2014; Khosla and Rose, 2020; Tedeschi et al., 2021b; Huguet Cabot et al., 2023), little attention has been devoted to the identification of concepts.

In this chapter, we fill this gap by introducing the Concept and Named Recognition (CNER) task, namely, the task of identifying and classifying both concepts and named entities into predefined semantic types. As shown in Figure 6.1, our new joint formulation enables a denser, richer, and more cohesive semantic annotation.

6.1.1 The CNER task

We start by formalizing the Concept and Named Entity Recognition (CNER) task and then introduce our process to obtain CNER categories. Formally, CNER is a sequence labeling task whose goal, given a sequence of tokens $t = (t_1, t_2, ..., t_n)$, is to identify the spans

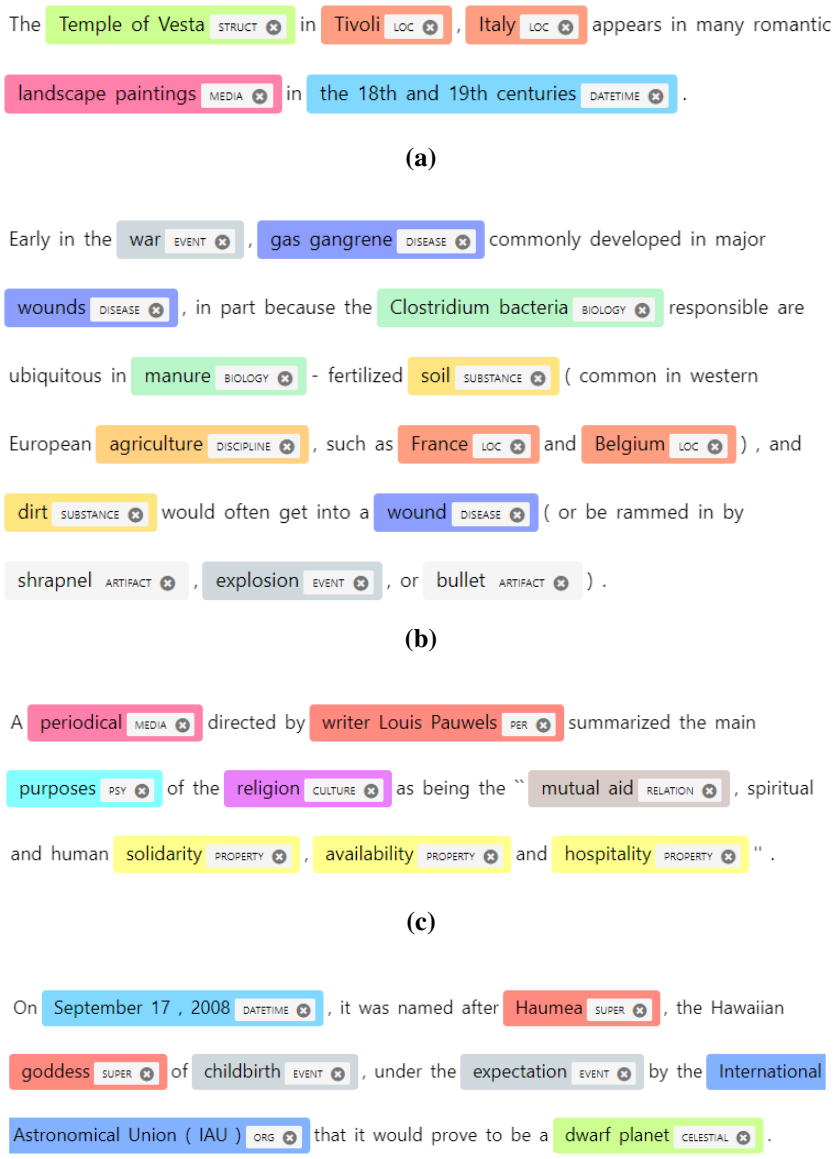

Figure 6.1. Four examples of CNER annotations where both nominal concepts and named entities are tagged with the proposed categorization.

of text S corresponding to nominal concepts and named entities, and categorize each of them into a predefined set of labels $L = \{l_1, l_2, ..., l_m\}$. Specifically, a text span $s_{ij} \in S$ is defined as a contiguous sequence of tokens $s_{ij} := (t_i, t_{i+1}, \ldots, t_j)$, with $1 \leq i \leq j \leq n$.

6.1.2 Categories

To design our comprehensive set of categories suitable for both concepts and named entities, we draw upon the robust cognitive foundations of WordNet (Miller, 1995) and OntoNotes (Pradhan et al., 2012a). Specifically, we leverage the completeness and broad semantic coverage of lexicographer files for nominal concepts, and integrate them with the widely-used semantic categories for named entities available in OntoNotes. Figure 6.2 shows the resulting 29 CNER categories (inner column) together with their alignment with the OntoNotes and WordNet ones (left and right columns, respectively). In particular, every category in OntoNotes and WordNet has a counterpart in our categorization, whereas the reverse does not hold. Specifically, i) every category highlighted in red has a direct link with a WordNet category (e.g. ANIMAL), ii) every category highlighted in yellow corresponds to an OntoNotes category (e.g. LAW), and iii) every category in a white cell is grounded on both resources (e.g. MEDIA).

To further assess the validity of such categorization and ensure that each instance is assigned to a reasonable category (i.e. semantically appropriate and neither too general nor too specific), we conducted a manual review of the most prominent WordNet synsets, i.e. the top 500 synsets ranked by their number of descendants in the WordNet taxonomy: a group of three human annotators independently reviewed this set of synsets and either assigned a category that is present in the already available categorization or proposed a new one, i.e. a category whose semantics better defines such an instance and that is well distinguished from the others. The result of this step is twofold. First, we obtained 500 synsets tagged with their CNER category which will be employed as a starting point for our automatic annotation procedure, explained in Section 6.1.4. Second, we identified six new categories, highlighted in green in Figure 6.2, that complement the WordNet and OntoNotes ones. The practical implication of the latter result is that, for example, biological concepts such as *molecule*, *protein* and *organism* will no longer be tagged with the OBJECT category, but as BIOLOGY. Analogously, terms such as *planet*, *quasar* and *star*, that would all have been included in

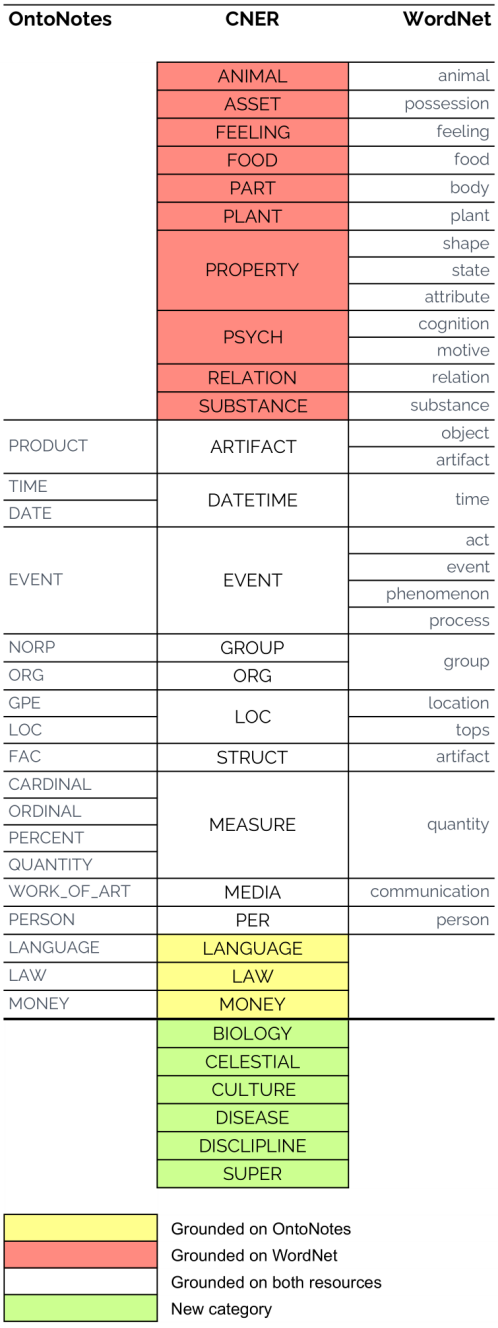

Figure 6.2. Comparison of our CNER categories (center) with the OntoNotes (left) and WordNet ones (right).

the OBJECT category, will now be assigned to a specific category named CELESTIAL. The same reasoning can be extended to the new categories CULTURE, DISEASE, DISCIPLINE and SUPER. For completeness, in Table 6.1 we provide a textual description along with instance examples for each CNER category.

6.1.3 Dataset

In the previous section, we described the procedure we used to obtain a new comprehensive categorization specifically designed to jointly capture concepts and named entities. We now provide datasets to enable the training and evaluation of CNER models. Crucially, we note that none of the existing datasets provides full coverage of both concepts and entities, usually expressed by common and proper nouns, respectively.

To fill this gap, we present a methodology for automatically annotating a large corpus of sentences with CNER categories (Section 6.1.4), which we refer to as CNER$_{silver}$. Then, we describe the annotation procedure we adopted to produce CNER$_{gold}$, a manually-curated dataset for model evaluation (Section 6.1.5). Finally, in Section 6.1.6 we present a detailed analysis of our two resources compared to the existing ones.

6.1.4 CNER$_{silver}$

At this stage, our goal is the creation of a large-scale training set for the CNER task. To achieve this, we further extend the WikiNEuRal methodology (cf. Chapters 3.1 and 3.3) to: first, disambiguate concepts and entities mentioned within text, and, second, to transform the resulting annotations into CNER categories. Again, we choose the English Wikipedia as our corpus because it offers a large number of heterogeneous texts spanning all domains of knowledge. To ensure high quality, we restrict ourselves to the subset of articles that the Wikipedia community has deemed to be "good" or "featured"[1] and apply our annotation strategy to these articles. However, as already discussed, Wikipedia articles contain a few terms that are linked manually to other articles, with many other terms left unlinked, leading to an issue of *sparsity*. Additionally, as most of the mentions linked in Wikipedia articles refer to named entities, this issue is particularly relevant for concepts[2]. Therefore, to ensure a high density of CNER annotations, we extend the WikiNEuRal pipeline by applying a state-of-the-art Word Sense Disambiguation model, ESCHER (Barba et al., 2021), to disambiguate each common noun in our dataset with a concept in BabelNet (Navigli and Ponzetto, 2012b). Finally, in order to annotate the remaining common and proper nouns with a CNER category, we use the Stanza NLP toolkit (Qi et al., 2020). This toolkit produces annotations using

[1] Wikipedia Good and Featured Articles.

[2] After applying a simple surface matching heuristic to propagate wikilink annotations, only 15% of common nouns are tagged, in contrast to 55.3% of proper nouns.

Category	Description	Examples
ANIMAL	*Living beings (excluding humans) with the ability to move and perceive their surroundings.*	*dog, cat, mammal, carnivore, brown bear, African Wild Dog, Great White Shark,*
ARTIFACT	*All the objects, artifacts, tools, products and items*	*vehicle, software, mouse, data stream, Windows XP, Fiat Panda*
ASSET	*Assets, resources, or possessions with economic or intrinsic value.*	*capital, stock, wealth, resource, phone bill, Federal Perkins Loan, Investment in Russia*
BIOLOGY	*Biological entities, including living organisms, cells, or biological components*	*protein, cell, living organism, lipid, Herpes Simplex Virus, Escherichia Coli*
CELESTIAL	*Celestial bodies as Planets, stars, asteroids, galaxies and other astronomical objects.*	*comet, nebulae, Sun, Neptune, Asteroid 187 Lamberta, Proxima Centauri*
CULTURE	*Cultural aspects, traditions, customs, and practices associated with specific groups or societies.*	*religion, feminism, socialism, capitalism, anarchism, doctrine, cult, Islam, Buddhism*
DATETIME	*Dates and times*	*18 March, Saturday, 1979, the evening of 19 November, 15:30 am*
DISEASE	*Medical conditions, illnesses, disorders, and health-related issues affecting living organisms.*	*infection, allergy, metastasis, complication, acne, Alzheimer's Disease, Cystic Fibrosis*
DISCIPLINE	*Specific fields of study, knowledge, or expertise. It includes academic disciplines, areas of research, and professional domains.*	*discipline, sport, football, computer science, anatomy, long jump.*
EVENT	*Events, phenomenon or activities that occur at specific times or places. It includes both significant and everyday occurrences*	*crime, professorship, temperature change, 2003 Wimbledon Championships, Cannes Film Festival.*
FEELING	*Emotions, sensations, and subjective experiences related to human or animal consciousness.*	*affection, attachment, agitation, craving, urge, temptation.*
FOOD	*Edible items, dishes, beverages, and culinary products that are consumed for nourishment or enjoyment*	*beverage, dish, pork, lasagna, Carbonara, Sangiovese, Cheddar Beer Fondue.*
GROUP	*Group of people or animals*	*staff, social group, panel, militia, community, trio, duo, family, genealogy, alliance, nationality, peoples*
LANGUAGE	*Individual language-related items, such as words, phrases, or idiomatic expressions*	*discourse, context, lexeme, morpheme, appellation, eponym, nickname, vowel, syllable, headword*
LAW	*Legal principles, regulations, and rules governing society and various aspects of life*	*law, civil law, administrative law, martial law, shariah, ordinance, civil right, Magna Carta.*
LOC	*Geographical locations, such as villages, towns, cities, regions, countries, continents, landmarks, or natural features*	*space, surface, street, road, town, Rome, Lake Paiku, Mississippi River.*
MEASURE	*Units of measurement and quantification used to determine the size, quantity, or quality of various objects or phenomena.*	*day, microsecond, millisecond, two, 35, 45%, first, temperature, length,*
MEDIA	*Various forms of communication and entertainment media, such as newspapers, tv shows, movies, social media or digital content.*	*soundtrack, report, publication, language, English, Forbes, American Psycho*
MONEY	*Monetary units, currencies, and financial values used in different contexts*	*monetary unit, dollar, 15 euros, 1116 CHF*
ORG	*Organizations, institutions, and companies involved in diverse sectors or activities*	*Industry, commercial enterprise, San Francisco Giants, Google, Democratic Party.*
PART	*Individual components or sections of larger entities or objects*	*finger, chin, head, tail, femur, airplane wing, airplane's wings, flower's stem*
PER	*Individuals or persons, including real people and historical figures*	*doctor, historian, professor, musician, Ray Charles, Jessica Alba*
PLANT	*Types of trees, flowers, and other plants, including their scientific names.*	*grass, peach tree, Forsythia, Artemisia Maritima.*
PROPERTY	*Properties or attributes of objects, entities, or concepts*	*thickness, height, dimension, shape, age*
PSYCH	*Psychological concepts, mental states, and phenomena related to the human mind and behavior*	*psycological feature, cognition, attention, necessity*
RELATION	*Relationships, connections, and associations between entities or concepts*	*apport, competition, comparison, bridge, relatedness, parentage, function, parity, transitivity*
STRUCT	*Physical structures, including buildings, architectural designs, and engineered constructions made by humankind*	*shelter, gravestone, refuge, tent, loft, San Peter's Church, Golden Bridge*
SUBSTANCE	*Chemical substances*	*acid, bactericide, carbonyl, explosive, fertilizer, Zyclon B*
SUPER	*Mythological and religious entities.*	*Apollo, Persephone, Aphrodite, Saint Peter, Pope Gregory I, Hercules.*

Table 6.1. Label, description, and instance examples of each of our CNER categories.

OntoNotes categories, which are then mapped to CNER categories by exploiting the one-to-one link between OntoNotes and CNER that was presented in Section 6.1.2. Remarkably, following the above-described process, we are able to annotate 98.4% of proper nouns, and 97.3% of common nouns, hence obtaining extremely dense annotations. This is in contrast

Sparsity heuristic	NOUN	PROPN
Wikilinks	7.1	33.2
+ surface matching heuristic	15.0	55.3
+ WSD	92.8	56.3
+ Stanza NER	97.3	98.4

Table 6.2. Impact of the various modules for solving sparsity in Wikipedia articles. The first row indicates the percentage of common and proper nouns hyperlinked in Wikipedia articles without applying any heuristic.

Dataset	Type	# Sentences	# Tokens	# Spans	# Tagged Tokens	AVG # Spans	Density %
OntoNotes 5.0	Gold	76 714	1 388 973	104 151	190 310	1.36	13.69
UFET$_{crowd}$	Gold	5878	154 802	5994	17 627	1.01	11.38
UFET$_{silver}$	Silver	2 272 421	59 500 651	3 121 857	7 016 134	1.37	11.80
CNER$_{gold}$	Gold	2000	56 843	14 730	22 461	7.37	39.56
CNER$_{silver}$	Silver	317 590	8 725 846	2 282 800	3 403 952	7.18	39.00

Table 6.3. Comparison between our newly-introduced CNER resource, OntoNotes and UFET. AVG # Spans is the average number of tagged spans – either concepts or named entities – per sentence. Density % is the percentage of tagged tokens over the total number of tokens.

to prior work as will be detailed later, in Section 6.1.6. Table 6.2 reports an ablation study highlighting the distinct contributions of the previously described modules. Importantly, for each annotated span, we include explicit information on whether it is a concept (C) or a named entity (NE), which is essential for the training of the specialized NER and CR models, as will be detailed in Section 6.1.7. Since BabelNet provides this information for its synsets, we include it for each instance that is annotated via taxonomy-based tagging and WSD. The instances annotated using Stanza NER, instead, are all considered to be named entities except for dates and numbers. We then convert the dataset that has been produced employing the BIO tagging scheme.

Once each article has been fully annotated, we transform each tag, be it a Wikipedia hyperlink, a BabelNet concept, i.e. synset, or a NER category, into its CNER category, and apply the semantic classifier introduced in Section 3.1.3. Specifically, to annotate each Wikipedia hyperlink w_j in an article W_i with a CNER category c, we first retrieve the corresponding BabelNet synset and map it to one of the 500 seed synsets introduced in Section 6.1.2, obtaining a CNER tag.

6.1.5 CNER$_{gold}$

To enable a proper and rigorous evaluation of CNER models, we introduce an annotation procedure for constructing CNER$_{gold}$, a manually curated dataset. The annotation process started by formulating guidelines through a meticulous examination of examples within our CNER$_{silver}$ dataset. However, the task presented multiple challenges related to the annotation of multiword nominal expressions. First, we established that non-compositional expressions such as *white shark* and *bald eagle*, which represent specific concepts, have to be identified as whole spans, while in compositional expressions, such as *black laptop*, only the noun has to be tagged. Second, in the presence of nested named entities, e.g. *Mr. Smith goes to Washington*, we decided to tag the largest available span denoting an entity, rather than tagging e.g. *Mr. Smith* and *Washington* separately. Third, we considered the cases in which the annotator is uncertain about whether a specific span of text is an entity or not (e.g. *One flew over the cuckoo's nest*), and what its potential meanings are (e.g. *mamihlapinatapai*). To help annotators disentangle these cases, we allowed them to use external world knowledge, i.e. giving access to web search, Wikipedia, etc., to properly identify and classify spans of text. We point out that this procedure proved to be effective for the annotation task: in a sample of 100 sentences, annotators, on average, utilized external resources for more than one span per sentence.

To start the annotation task, we randomly sampled 2,000 sentences from CNER$_{silver}$ and provided the annotators with a simple interface where each row displayed a specific token and its corresponding PoS tag. The annotators were asked to identify concepts and named entities by means of specific C vs NE tags and label them with the corresponding CNER categories by making their choice via a drop-down menu.

Because the dataset was annotated by a single expert linguist with a robust background in the annotation of lexical-semantic tasks, and an author of this work, we asked two other annotators to label 10% of the data. We then calculated the inter-annotator agreement on such instances and obtained a Fleiss' κ score of 89.8, indicating optimal agreement. The overall annotation process took 2 months, with a total of 320 hours spent on the task. This process led to the creation of a corpus of 2,000 sentences and more than 56,000 annotated tokens, $\sim$22,000 of which were tagged with a CNER category.

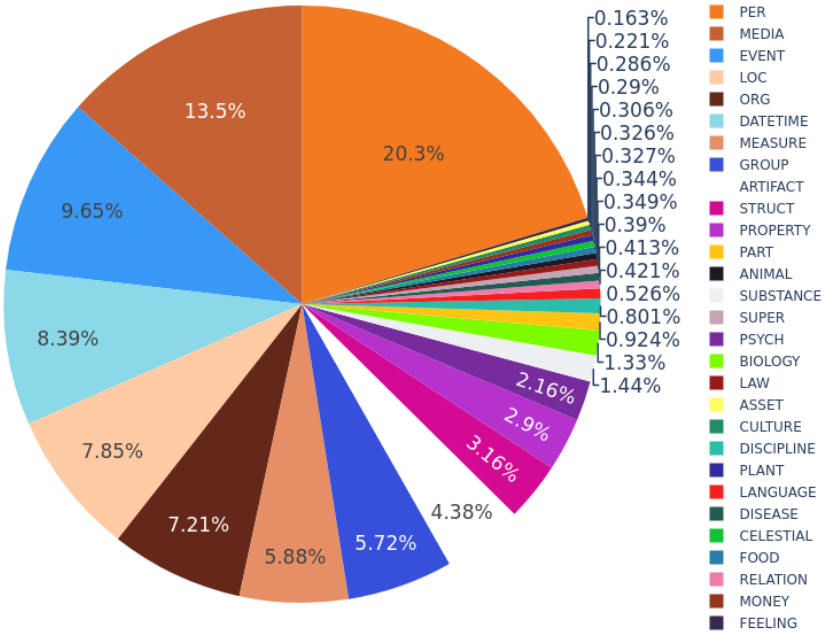

Figure 6.3. Pie chart showing the CNER$_{silver}$ category distribution.

6.1.6 Dataset Statistics

In Table 6.3, we provide the statistics of our CNER$_{gold}$ and CNER$_{silver}$ datasets compared to OntoNotes. We observe that our training set comprises a significantly larger number of annotated sentences, spans and tokens. Furthermore, from the last two columns, we observe a considerably higher annotation density, highlighting that tagging both named entities and nominal concepts leads to a denser and richer annotation compared to focusing on entities only. In Table 6.3 we also present the statistics of UFET (Choi et al., 2018), but while their formulation encompasses the classification of both named entities and concepts, the annotation density is even lower than that of OntoNotes, as they disregard the identification part.

In addition, the pie charts in Figures 6.3 and 6.4 show the category balance in CNER$_{silver}$ and CNER$_{gold}$, respectively. Finally, in order to evaluate the quality of the automatic annotation procedure (Section 6.1.4), we compute the agreement over the set of 2,000 sentences reserved for CNER$_{gold}$. The Cohen's kappa coefficient over this set of samples is $\kappa = 71.4$, indicating a substantial agreement between the automatic and manual procedures. Finally, in Figure 6.5, we provide the distribution of concept and named entities for each individual category in our datasets.

6.1.7 Experimental Setup

We now describe the setup, including datasets (Section 6.1.7) and model architecture (Section 6.1.7), used to answer the following research questions:

- **(RQ1)** How does a competitive neural model fare on the CNER task?
- **(RQ2)** Can a CNER system perform on par with, or even better than, specialized NER and CR systems on the respective subtasks?

Datasets. In our experiments we use several versions of our data:

- CNER$_{silver}$ and CNER$_{gold}$ are, respectively, the silver- and gold-standard datasets introduced in Section 6.1.3 covering both nominal concepts and named entities;
- NER$_{silver}$ and NER$_{gold}$ are the same datasets as above, in which only entity-related annotations are retained (i.e. concepts are mapped to the O class);
- CR$_{silver}$ and CR$_{gold}$ are the same datasets as above, in which only concept-related annotations are retained (i.e. named entities are mapped to the O class);

Based on these datasets, we train and evaluate the same model architecture under the following four different settings: i) we train on CNER$_{silver}$ and test on CNER$_{gold}$, ii) we train on CNER$_{silver}$ and test on NER$_{gold}$ and CR$_{gold}$, iii) we train on NER$_{silver}$ and test on NER$_{gold}$, and iv) we train on CR$_{silver}$ and test on CR$_{gold}$.[3] Differently from setting (i) in which a distinction between concepts and named entities is not required, in setting (ii) we train a CNER model on CNER$_{silver}$ which, for each predicted span, provides the type (C or NE) in addition to the CNER category (e.g. for the token *doctor* the model outputs B-C-PERSON). This allows us to retain only entity- or concept-related annotations when evaluating on NER$_{gold}$ and CR$_{gold}$, respectively, enabling a fair evaluation and ensuring comparability between the CNER model and the specialized NER and CR systems.[4] In fact, an unfiltered evaluation of CNER predictions would lead to uninformative results, since a CNER model would naturally provide dense annotations regarding all the nominal expressions in a sentence, independently of their type (C or NE).

[3] We split the gold data into half for validation, half for testing (we refer to the latter as CNER$^{test}_{gold}$, NER$^{test}_{gold}$ and CR$^{test}_{gold}$).

[4] We highlight that NER systems implicitly learn to identify NEs, in the same manner as CR does with concepts.

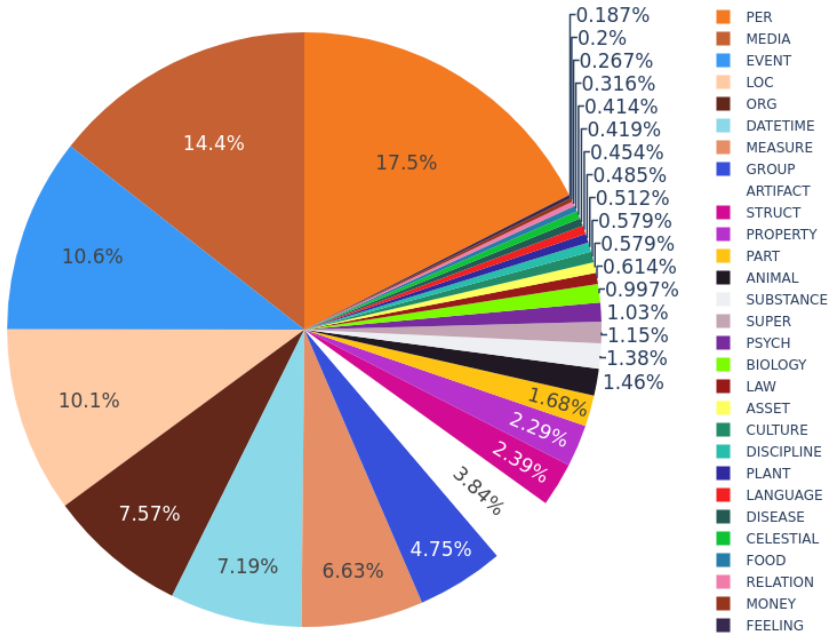

Figure 6.4. Pie chart showing the CNER$_{gold}$ category distribution.

As evaluation metrics, we adopt the macro F1 score at the token level and the span-based F1 score. Additionally, we also report token-level micro F1 scores, because – differently from NER – the O category is much less frequent, given the high density of CNER annotations (cf. last column of Table 6.3).

Architecture. For our experiments, we opted for an architecture that strikes a balance between accuracy and simplicity, offering the robustness required for our task while being efficient and easy to be fine tuned. Specifically, we employ a pretrained transformer encoder, namely DeBERTa-v3 base (He et al., 2023), and train a classification head on top of it to predict the category for each token in a sentence. The classification head consists of two linear layers and a normalization layer, with a dropout of 0.1 and the GeLU activation function. Since this architecture has been proven to achieve competitive performance compared to the current state of the art in standard NER benchmarks (He et al., 2023), we believe that it constitutes a strong baseline for the CNER task.

Our systems are developed using the PyTorch Lightning framework[5] and the HuggingFace models library.[6] We train them on a single RTX 4090 Ti, with a patience parameter

[5] https://www.pytorchlightning.ai/
[6] https://huggingface.co/docs/transformers/

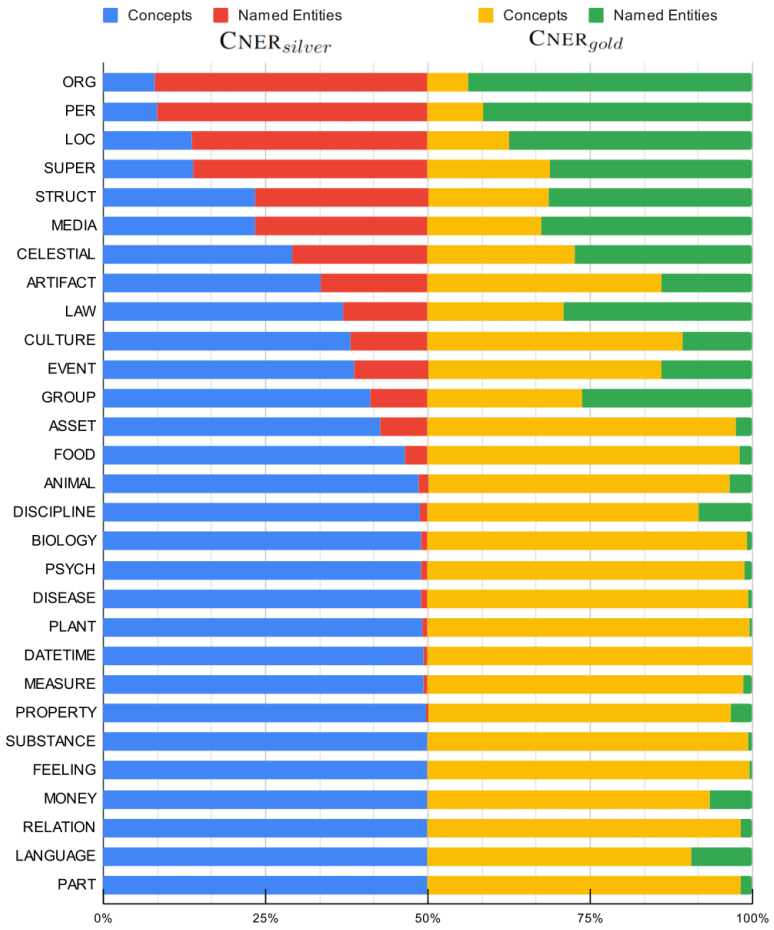

Figure 6.5. Bar chart with the distribution of named entities and concepts for every CNER category in the CNER$_{gold}$ and CNER$_{silver}$ resources.

of 20 and a validation step every 30% of the total number of steps per epoch, resulting in 4 epochs and approximately 2 hours of training time for each model. We use the RAdam optimizer with a learning rate of 5×10^{-6}, and a linear scheduler with a warm-up of 10% of the total steps. We adopt the same architectural setup and hyperparameters to train the CNER, NER and CR models. All models are trained to minimize the cross-entropy loss function. We select the best model based on its macro F1 score on the validation set.

6.1.8 Results

RQ1. The proposed architecture achieves 87.20 micro F1, 59.38 macro F1 and 66.72 span F1 score points, when asked to identify and classify both named entities and concepts with our 29-category tagset (cf. setting (i) in Section 6.1.7). In Figure 6.6, we provide the indi-

Train \ Test	NER^{test}_{gold}			CR^{test}_{gold}		
	Micro	Macro	Span	Micro	Macro	Span
(ii) CNER_{silver}	**94.07**	**39.65**	**69.15**	**91.41**	**52.47**	**63.77**
(iii) NER_{silver}	93.59	34.28	66.73	—	—	—
(iv) CR_{silver}	—	—	—	89.71	44.48	59.66

Table 6.4. Micro, Macro and Span F1 scores (%). (ii), (iii) and (iv) refer to the settings described in Section 6.1.7.

vidual span F1 scores for each category in our benchmark. In general, by cross-referencing the category scores with the distribution in Figure 6.3, we observe that performance has a moderate correlation with the number of training instances (Pearson correlation coefficient $\rho = 0.4$).

From a closer perspective, our system is affected by two main factors. First, when evaluated only on the identification of spans, the system achieves 86.83 span F1 score points, setting an upper bound to the overall system performance. Indeed, the model struggles to identify spans like *Triple J hottest 100 of all time in Australia* and *A full day's Work* in their entirety. Second, the confusion matrix Figure 6.7 shows that the model sometimes mistakes categories that have an intrinsic level of ambiguity (e.g. ANIMAL or PLANT with FOOD). In particular, terms that can be associated with multiple categories (e.g. *chicken* and *eggplant*) are often difficult to be classified based on the limited sentence context. Additionally, specific terms, often contained in Wikipedia articles, like *Synapsids Ophiacodon* or *metal umlaut*, require technical expertise to be properly classified. Similarly, the categories PROPERTY, PSYCH, and FEELING are frequently confused due to their intertwined semantics, while instances belonging to the LAW category, such as *peace treaty*, can occasionally overlap with instances from the MEDIA category, which encompasses general written documents conveying information.

RQ2. We summarize the results of our experiments in Table 6.4. Our findings clearly illustrate the significant synergistic benefits of training a single model to simultaneously identify and classify concepts and named entities, compared to restricting the same model to either concepts or named entities. The CNER model exhibits a notable increase of $\sim$0.5 micro, $\sim$5.4 macro and $\sim$2.4 span F1 points over the NER model. A similar improvement

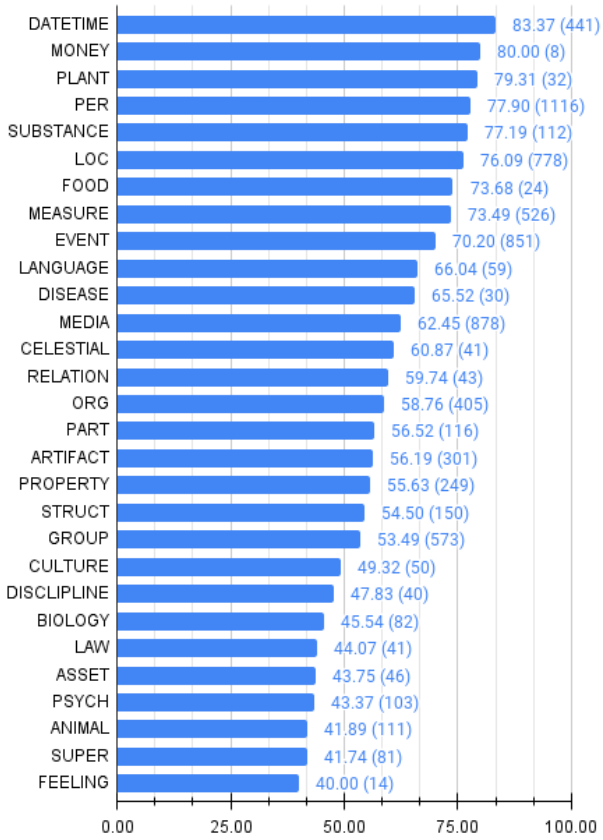

Figure 6.6. Bar chart with span F1 score (%) for each CNER category sorted in descending order. The support for each category is in parentheses.

is achieved over the CR model with a ∼1.7 micro, ∼8.0 macro, and ∼4.1 span F1 point increase. These findings underline the advantages of our proposed CNER approach in improving the accuracy and effectiveness of both concept and named entity recognition tasks. In Section 6.1.9, we provide a qualitative analysis of our results.

6.1.9 Qualitative Analysis

To better understand the behaviour of CNER, NER and CR models, we conduct a qualitative error analysis on $\text{CNER}^{test}_{gold}$ and report some representative examples in Table 6.13.

In the first example, we can immediately appreciate the higher annotation density of both concepts and named entities produced by the CNER model compared to the specialized NER and CR alternatives. We also report, in the second example, an instance in which the CNER model correctly predicts the label PROPERTY for the concept *role* while CR

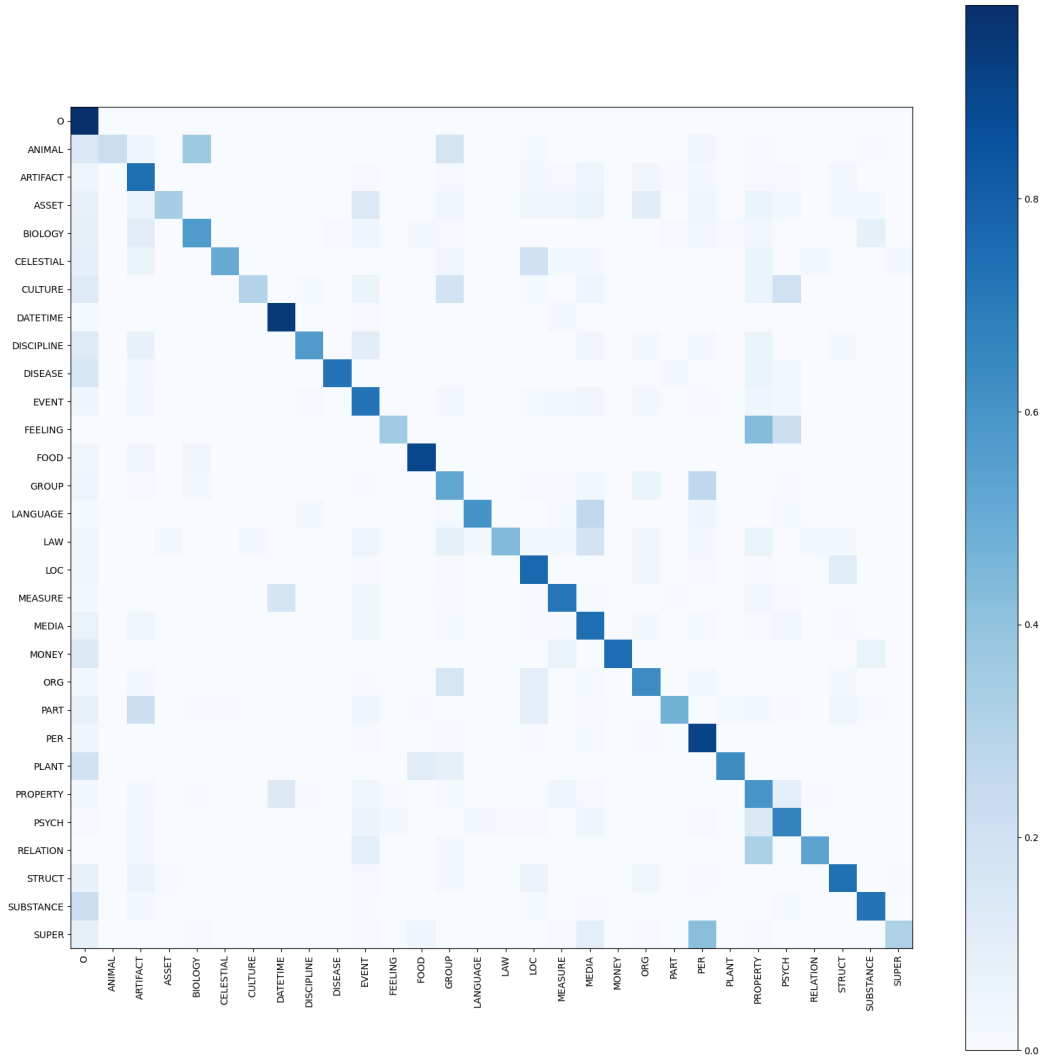

Figure 6.7. Confusion matrix on the CNER task.

misclassifies it as an EVENT. Furthermore, in the same sentence, we have an example of an instance for which there is one correct label, but, based on the given context, other tags could be associated with it. Specifically, both the CNER and CR models misclassify *tobacco* by tagging it with FOOD and SUBSTANCE, respectively, which are less appropriate, but both plausible annotations within the given context. In the third example, instead, we present a sentence in which our system shows better generalization capabilities: while the named entity *John* is misclassified as PER by both the NER and CNER models, the CNER system correctly assigns the label SUPER to the concept *zombie*, which is misclassified by the CR model with the tag BIOLOGY.

Additionally, in Figure 6.8, we illustrate the category-wise impact of the CNER

	Type	Prediction
Example 1	Gold	The **[first]**$_{MEASURE}$ **[town]**$_{LOC}$ to fall into the **[Lombards]**$_{GROUP}$' **[hands]**$_{PART}$ was **[Forum Iulii]**$_{LOC}$ (**[Cividale del Friuli]**$_{LOC}$), the **[seat]**$_{LOC}$ of the local **[magister militum]**$_{PER}$.
	NER	The first town to fall into the **[Lombards]**$_{GROUP}$' hands was **[Forum Iulii]**$_{STRUCT}$ (**[Cividale del Friuli]**$_{LOC}$), the seat of the local magister militum.
	CR	The **[first]**$_{MEASURE}$ **[town]**$_{LOC}$ to fall into the Lombards' **[hands]**$_{PART}$ was Forum Iulii (Cividale del Friuli), the **[seat]**$_{LOC}$ of the local **[magister]**$_{PER}$ militum
	CNER	The **[first]**$_{MEASURE}$ **[town]**$_{LOC}$ to fall into the **[Lombards]**$_{GROUP}$' **[hands]**$_{PART}$ was **[Forum Iulii]**$_{STRUCT}$ (**[Cividale del Friuli]**$_{LOC}$), the **[seat]**$_{LOC}$ of the local **[magister]**$_{PER}$ militum
Example 2	Gold	**[Tom McCarthy]**$_{PER}$ highlighted the prominent **[role]**$_{PROPERTY}$ of **[tobacco]**$_{PLANT}$ in the **[story]**$_{MEDIA}$, drawing on the **[ideas]**$_{PSYCH}$ of **[philosopher Jacques Derrida]**$_{PER}$ to suggest the potential **[symbolism]**$_{MEDIA}$ of this.
	NER	**[Tom McCarthy]**$_{PER}$ highlighted the prominent role of tobacco in the story, drawing on the ideas of **[philosopher Jacques Derrida]**$_{PER}$ to suggest the potential symbolism of this.
	CR	Tom McCarthy highlighted the prominent **[role]**$_{EVENT}$ of **[tobacco]**$_{SUBSTANCE}$ in the **[story]**$_{MEDIA}$, drawing on the **[ideas]**$_{PSYCH}$ of Jacques Derrida to suggest the potential **[symbolism]**$_{PSYCH}$ of this.
	CNER	**[Tom McCarthy]**$_{PER}$ highlighted the prominent **[role]**$_{PROPERTY}$ of **[tobacco]**$_{FOOD}$ in the **[story]**$_{MEDIA}$, drawing on the **[ideas]**$_{PSYCH}$ of **[philosopher Jacques Derrida]**$_{PER}$ to suggest the potential **[symbolism]**$_{PSYCH}$ of this.
Example 3	Gold	**[John]**$_{SUPER}$ has new **[weapons]**$_{ARTIFACT}$, including **[holy water]**$_{SUBSTANCE}$, **[bait]**$_{BIOLOGY}$ for the **[undead]**$_{SUPER}$, and a **[blunderbuss]**$_{ARTIFACT}$ that uses **[zombie]**$_{SUPER}$ **[parts]**$_{ARTIFACT}$ as **[ammunition]**$_{ARTIFACT}$.
	NER	**[John]**$_{PER}$ has new weapons, including holy water, bait for the undead, and a blunderbuss that uses zombie parts as ammunition.
	CR	John has new **[weapons]**$_{ARTIFACT}$, including **[holy water]**$_{SUBSTANCE}$, **[bait]**$_{SUBSTANCE}$ for the **[undead]**$_{BIOLOGY}$, and a blunderbuss that uses **[zombie]**$_{BIOLOGY}$ **[parts]**$_{ARTIFACT}$ as **[ammunition]**$_{ARTIFACT}$.
	CNER	**[John]**$_{PER}$ has new **[weapons]**$_{ARTIFACT}$, including **[holy water]**$_{SUBSTANCE}$, **[bait]**$_{SUBSTANCE}$ for the undead, and a blunderbuss that uses **[zombie]**$_{SUPER}$ **[parts]**$_{ARTIFACT}$ as **[ammunition]**$_{ARTIFACT}$.

Table 6.5. Examples of annotations from the inference of our CNER, NER and CR models over the test split of our CNER$_{gold}$ dataset. Correct predictions are highlighted in green, while wrong predictions are highlighted in red.

system when compared to NER and CR systems, highlighting its positive and negative contributions. Notably, the CNER system consistently exhibits a positive influence over both named entities and concepts. Specifically, categories such as DISCIPLINE, FOOD, and CULTURE witness particularly positive contributions, with an increase of up to 40 span-based F1 score points. In some categories, the CNER system provides positive contributions only for named entities, as observed in GROUP and MEDIA, or exclusively for concepts, as evidenced in SUPER and LAW.

6.2 Idiomatic Expressions

Among multi-word expressions (MWEs), idioms are of particular interest in that their meaning cannot be obtained by compositionally interpreting their word constituents. These include non-compositional phrases, e.g. *kick the bucket*, and partially-compositional phrases, e.g. *rain cats and dogs* (Nunberg et al., 1994). Given their complex nature, idioms are hard to be automatically identified and pose a crucial challenge to the entire field of Natural

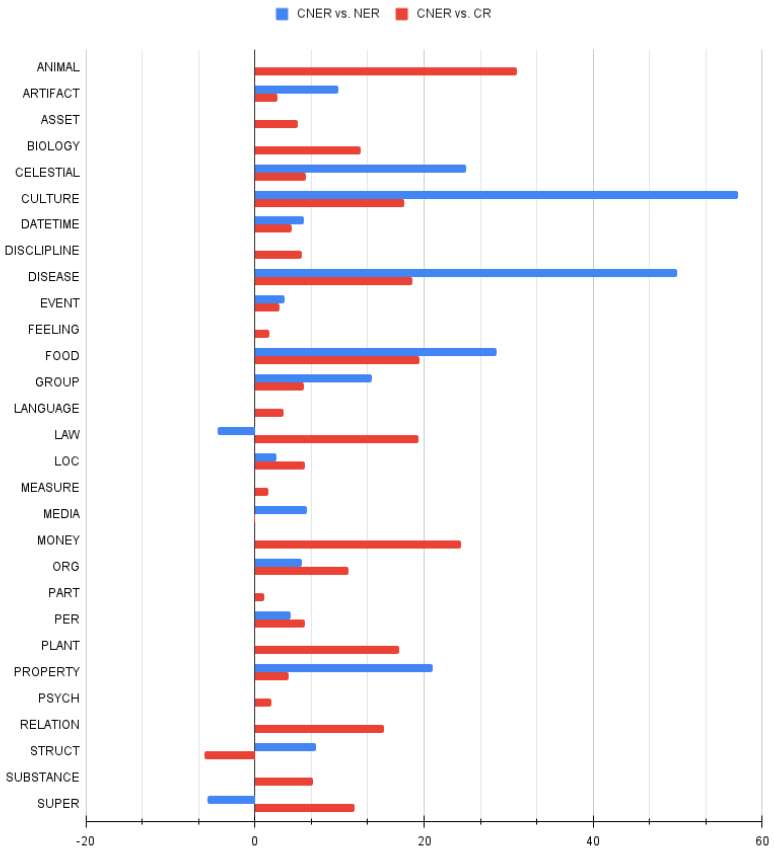

Figure 6.8. Bar chart with Span F1 score of the different categories.

Language Understanding (NLU). Although research in this field has recently achieved great advancements, the current formulation of many tasks tends to overlook the idiomatic usage of language. Instead, idioms ought to be playing an important role in NLU as they are a frequent phenomenon which can be observed in all languages. The correct identification of idioms in context is crucial for tasks such as Word Sense Disambiguation (Bevilacqua et al., 2021) and Entity Disambiguation (Sevgili et al., 2020), but also for many downstream applications. For instance, in Question Answering or dialog, a system must be able to understand *"It was a piece of cake"* in relation to the question *"How was the test?"* (Jhamtani et al., 2021; Mishra and Jain, 2016). Similarly, if the idiom *kick the bucket* is identified, then a Text Summarization system would be able to summarize all its occurrences within a text with "die" (Chu and Wang, 2018; Gambhir and Gupta, 2017). Finally, once an idiom is identified, a Machine Translation system would then be able to avoid its compositional

translation, and treat it as a whole (Anastasiou, 2010). Furthermore, idioms are widely studied in linguistics and psycholinguistics (Cacciari and Tabossi, 1988; Gibbs Jr, 1992; Nunberg et al., 1994; Cacciari and Tabossi, 2014; Liu, 2017), hence a system capable of effectively identifying idioms in texts would significantly improve many research areas, far beyond NLU. Additionally, most of the past idiom extraction strategies focused on specific domains and on a limited number of languages.

To tackle these shortcomings, in this chapter, we first present our new methodology for the creation of multilingual training data for the idiom identification task (Section 6.2.1), and then introduce our manually-curated evaluation benchmark (Section 6.2.2). Finally, we reframe the idiom identification task as a sequence labeling task and propose a new idiom identification system based on our data and task formulation (Section 6.2.4).

6.2.1 Silver-Standard Data Creation

Automatic Annotation. In order to create our training data, we exploit Wiktionary[7] as the main source, as it provides access to a large number of MWEs along with usage examples in multiple languages. However, since such examples are provided for a limited number of MWEs, we search for further textual contexts in a large external source, namely WikiMatrix[8] (Schwenk et al., 2021), a multilingual corpus that covers 83 languages and contains parallel sentences extracted from Wikipedia[9].

We perform data extraction as follows. Let E_l be the set of MWEs available in Wiktionary in the language l, with $|E_l| = n$, and let us define the function $L(p)$ that, given a phrase p, outputs its lemma. Then, we apply a heuristic which allows us, for each expression $e_i \in E_l$, to search for a sentence in WikiMatrix such that there exists at least a span of tokens S_{k-j} starting at index k and ending at index j, where $e_i = S_{k-j} \vee e_i = L(S_{k-j}) \vee L(e_i) = S_{k-j} \vee L(e_i) = L(S_{k-j})$. By applying this heuristic, not only do we obtain a large set of sentences containing potentially idiomatic expressions (PIEs), but – thanks to the lemmatization step – we also collect several morphological

[7] We employ the Wiktextract library to collect the necessary data from Wiktionary. WikiExtract (`https://pypi.org/project/wiktextract/`) provides a preprocessed version of the Wiktionary dump together with useful APIs.

[8] `https://github.com/facebookresearch/LASER/tree/main/tasks/WikiMatrix`

[9] The encyclopedia-style prose of Wikipedia could have a lower idiom density compared to other textual sources, but the large size of WikiMatrix should balance this lower density.

variations of the original expressions in E_l, e.g. starting from '*kick the bucket*', we also obtain '*kicked the bucket*' and '*kicks the bucket*'. In particular, if an MWE is marked as idiomatic in Wiktionary, we mark all its occurrences as idiomatic too. Similarly, if an MWE is not marked as idiomatic in Wiktionary, we mark all its occurrences as literal. However, this has a limitation: if an MWE is labeled as idiomatic (or literal) in Wiktionary, it will not necessarily always also be idiomatic (or literal) in the WikiMatrix sentences in which it appears.

We adopt the above-described procedure to create datasets in the following 10 languages: Chinese, Dutch, English, French, German, Italian, Japanese, Polish, Portuguese and Spanish.

Automatic Validation. Since the data derived from Wiktionary and WikiMatrix may contain errors, we aim at automatically improving their quality. To achieve this goal, we exploit the semantic idiosyncrasy property of idiomatic expressions, and the consequent fact that the meaning of the individual constituents of idiomatic expressions are unrelated to the surrounding context. Specifically, following this intuition, and by taking inspiration from recent advances in the main disambiguation tasks (Blevins and Zettlemoyer, 2020; Botha et al., 2020; Tedeschi et al., 2021b), we design a dual-encoder architecture (Figure 6.9) to produce a vector representation for both the expression and its context, and then, based on their cosine similarity, label the expression as idiomatic or literal.

More formally, let us define an expression encoder Ψ and a context encoder Ω. Then, given an expression-context pair $\langle e, c \rangle$, the output of the dual-encoder architecture Φ is defined as follows:

$$\Phi(e,c) = \begin{cases} 1, & \text{if } \dfrac{\Psi(e)^T \Omega(c)}{\|\Psi(e)\| \, \|\Omega(c)\|} \leq \delta \\ 0, & \text{otherwise} \end{cases}$$

where $\Phi(e, c) = 1$ means that e is idiomatic in c, while $\Phi(e, c) = 0$ if e has a literal meaning in c. δ is a manually-tuned threshold. Both encoders are `bert-base-multilingual-cased` architectures that take as input the tokenized versions of expressions and their contexts, respectively, surrounded by the special tokens [CLS] and [SEP]. To encode an expression, we take the sum of the individual representations of all its subwords. Instead, for the representation of the context we take the representation of the [CLS] token. We evaluate the

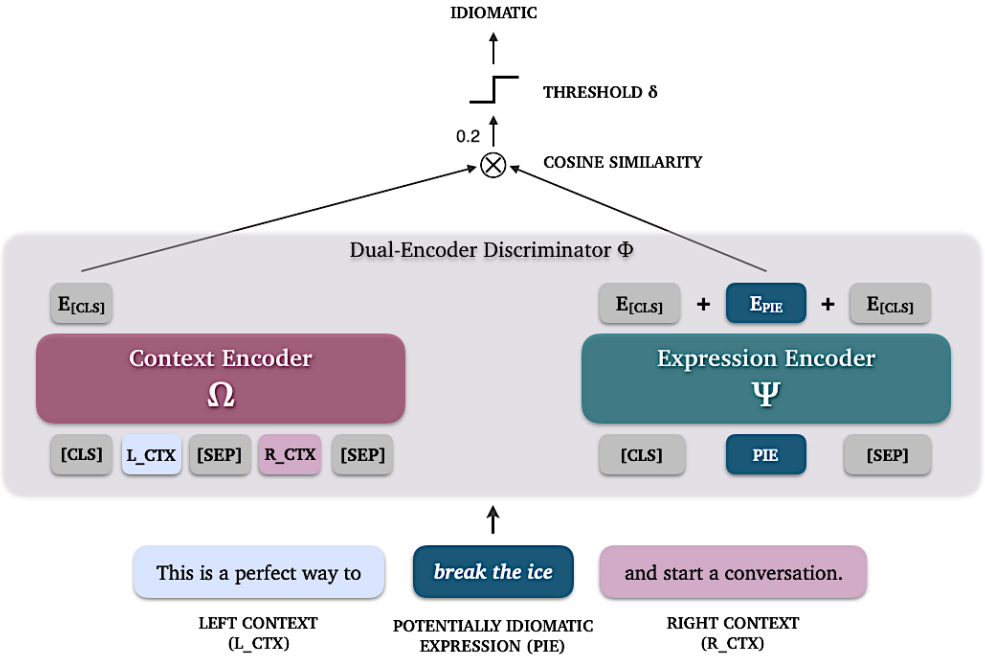

Figure 6.9. Graphical representation of the dual-encoder architecture given as input an example sentence. "E" stands for Embedding. A potentially idiomatic expression e is labeled as idiomatic when the cosine similarity score between the representations $\Omega(c)$ and $\Psi(e)$, where c is the surrounding context, is lower than the threshold δ.

quality of our dual encoder in Section 6.2.6.

Additionally, to further improve the quality of the annotations produced, we follow the recent findings of Tedeschi and Navigli (2022) which demonstrated how NER can be exploited to better discriminate between idiomatic and literal usages of potentially idiomatic expressions.

6.2.2 Gold-Standard Data Creation

To evaluate the performance of our idiom identification system, we manually create a novel evaluation benchmark in 4 languages, i.e. English, German, Italian and Spanish. As explained in Section 6.2.1, we start by producing a set of sentences containing PIEs. Then, to properly label the expressions, depending on the context in which they occur, we ask professional annotators[10] to perform the following binary classification task: given a context-expression pair $\langle e, c \rangle$, the goal is to tag this pair with a label $y \in \{Idiomatic, Literal\}$. In order to make our gold standard more challenging, and better evaluate the system

[10]We hired a mother-tongue professional annotator for each language.

	Language	# Sentences	# Tokens	# Idioms	# B	# I	# O	# Seen	# Unseen	# Literal
Silver Data	Chinese (ZH)	9543	244422	1301	5272	3823	235327	-	-	3918
	Dutch (NL)	20935	548872	189	4530	10543	533799	-	-	16366
	English (EN)	37919	1199492	4568	10102	19884	1169506	-	-	27408
	French (FR)	35588	939161	188	12112	25248	901801	-	-	23238
	German (DE)	26963	722109	819	8311	11500	702298	-	-	18488
	Italian (IT)	29523	813445	452	8768	12353	792324	-	-	20506
	Japanese (JA)	6388	211437	165	2534	1662	207241	-	-	3852
	Polish (PL)	36333	862265	648	12971	14364	834930	-	-	22467
	Portuguese (PT)	30942	764017	559	5824	8871	749322	-	-	24816
	Spanish (ES)	28647	648776	1229	9994	13927	624855	-	-	17851
Gold Data	English (EN)	200	3287	142	159	373	2755	62	80	41
	German (DE)	200	4529	111	181	377	3971	71	40	19
	Italian (IT)	200	5043	139	155	271	4617	87	52	48
	Spanish (ES)	200	2240	78	133	348	1759	19	59	66

Table 6.6. Statistics concerning the automatically-created (Silver Data) training sets and our manually-curated test sets (Gold Data). "# Seen" represents the number of expressions in the test set already encountered in the training set, whereas "# Unseen" is the number of expressions never encountered. In the count of individual idioms (# Idioms), morphological variations of a certain idiom are mapped to the same idiom.

performance, we also ask annotators to include unseen idioms, i.e. idioms that do not appear in the training set.

6.2.3 Idiom Identification: A New Task Formulation

Current and past approaches to idiom identification typically take expressions-context pairs $\langle e, c \rangle$ as input and limit themselves to determining whether e is used with a figurative meaning or not in c (Madabushi et al., 2021; Muzny and Zettlemoyer, 2013). However, this formulation has a major drawback: potentially idiomatic expressions need to be pre-identified. Importantly, we drop this requirement and reformulate the task as a sequence-labeling task, by employing the well-known BIO tagging scheme[11]. More formally, given as input a raw text sequence X of n tokens $x_1, \ldots, x_n$, each x_i must be labeled by the system with a tag $y_i \in \{B, I, O\}$ for each $i \in [1, n]$. This formulation also allows us to easily handle multiple idiomatic expressions within the same text.

In order to use our new formulation, we convert all the datasets constructed in Section 6.2.1 and Section 6.2.2 in BIO format. Table 6.7 shows an example of instance labeled using the BIO tagging scheme.

[11]The BIO tagging scheme (short for Beginning, Intermediate, Out) is a popular tagging scheme where the B label indicates that the corresponding token is the first token of a span, in this case an idiomatic expression, the I label denotes an intermediate token of a span, and O means out of a span.

Token	Label
After	O
some	O
reflection	O
,	O
he	O
decided	O
to	O
bite	B-IDIOM
the	I-IDIOM
bullet	I-IDIOM
.	O

Table 6.7. Example of instance labeled according to the BIO tagging scheme.

6.2.4 Our System

Our model for idiom identification is inspired by the BERT-based neural architecture of Mueller et al. (2020) used for Named Entity Recognition, however, rather than encoding a word with the first contextualized subword representation as indicated by Devlin et al. (2019), we take the mean of its subwords, as suggested by recent literature (Ács et al., 2021). Then, the resulting vectors are passed through a multi-layer sentence-level BiLSTM network, whose logits are finally fed into a CRF model, trained to maximize the log-likelihood of the span-based gold label sequences (Huang et al., 2015).

6.2.5 Experimental Setup

Architecture. We implement our idiom identification system (Section 6.2.4) and our dual-encoder discriminator (Section 6.2.1) with PyTorch (Paszke et al., 2019), using the Transformers library (Wolf et al., 2019) to load the weights of `BERT-base-multilingual-cased` (mBERT). We fine-tune our idiom identification system for 30 epochs with a Cross-Entropy loss criterion, adopting an early stopping strategy with a patience value of 5, Adam (Kingma and Ba, 2015c) optimizer and a learning rate of 10^{-5}, as standard when fine-tuning the weights of a pretrained language model. For our dual-encoder discriminator, instead, we use mBERT as feature extractor since no training data for the task were available.

The entire model training is carried out on a NVIDIA GeForce RTX 3090. Each

Hyperparameter name	Value
number of Bi-LSTM layers	2
LSTM hidden size	256
gradient accumulation steps	4
batch size	32
learning rate	0.00001
dropout	0.5
gradient clipping	1.0
adam β_1	0.9
adam β_2	0.999
adam ϵ	1e-8

Table 6.8. Hyperparameter values of the reference idiom identification system used for our experiments.

Language	Accuracy
English	84.12
German	81.98
Italian	82.74
Spanish	82.55
Avg.	**82.85**

Table 6.9. Accuracy of the annotations produced by our automatic system compared to those provided by the human annotators on the 4 languages covered by our gold-standard test sets.

training (i.e. for each language) requires ∼8min/epoch on average, for a mean of ∼20 epochs. Table 6.8 shows the full list of hyperparameters.

Data. The training and validation sets that we use in our experiments are those obtained by applying the methodology described in Section 6.2.1, with $\delta = 0.4$[12]. Although we automatically produce training data in 10 languages, we report results only on the 4 languages for which manually-curated test sets are available (see Section 6.2.2). However, since the training data has been created with the same procedure for each of the 10 languages, similar results are expected on non-tested languages. Statistics are provided in Table 6.6.

[12]We use the English validation set to manually search for the best value of δ by choosing from the following set of possible values: $\delta = \{0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7\}$

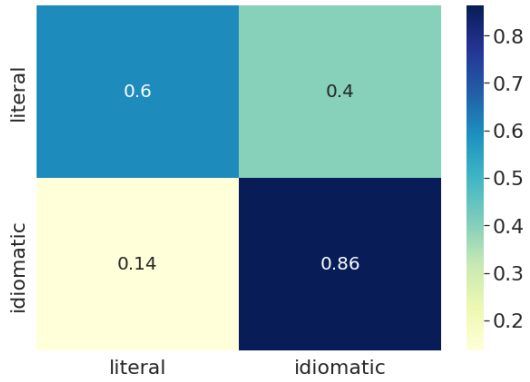

Figure 6.10. Confusion matrix of the predictions of our automatic system (X-axis) compared to the corresponding ground truth values (Y-axis). Results are averaged over the 4 languages covered by the test sets.

6.2.6 Results

In what follows, we measure both the quality of our automatic annotation methodology (Section 6.2.1) and of our idiom identification system (Section 6.2.4) by means of accuracy and token-level macro F1-score metrics, respectively. In the latter case, we rely on the macro-F1 metric due to the high class imbalance in the datasets, i.e. the number of O tags is much higher than the sum of the number of B and I tags, see Table 6.6.

Silver-Data Quality Evaluation. We first aim at providing an empirical evaluation of the effectiveness of the proposed automatic strategy for producing idiom-related[13] sentences. To do so, for each language, we apply our dual-encoder discriminator $\Phi(e, c)$ to the expression-context pairs $\langle e, c \rangle$ available in our manually-curated test set, and we measure the accuracy score by comparing the predictions produced by the system with the human annotations in the gold-standard test sets. The accuracy results obtained are reported in Table 6.9.

With this being a binary-classification task, we can observe that the performance achieved by our dual encoder is much higher than the 50% baseline of a random classifier, hence implying that the system is able to distinguish between idiomatic and literal usages of PIEs based on the surrounding contexts.

However, the accuracy is not sufficient for us to determine the strengths and the weaknesses of our system. Therefore, we group both the predictions and the labels coming from the 4 languages, and construct a confusion matrix in order to better analyze the system

[13]With the term “idiom-related sentences" we refer to sentences containing potentially idiomatic expressions.

	Tag	P	R	F1	% Seen
EN	B	84.2	53.5	65.4	-
	I	91.1	57.4	70.4	-
	O	92.5	99.1	95.7	-
	ALL	**89.2**	**70.0**	**77.1**	43.7%
DE	B	87.6	70.2	77.9	-
	I	90.7	72.7	80.7	-
	O	96.2	98.9	97.5	-
	ALL	**91.5**	**80.6**	**85.4**	64.0%
IT	B	72.2	61.9	66.7	-
	I	76.7	62.0	68.6	-
	O	96.7	98.2	97.4	-
	ALL	**81.8**	**74.0**	**77.6**	62.6%
ES	B	47.2	51.1	49.1	-
	I	67.4	45.1	54.0	-
	O	87.7	92.8	90.2	-
	ALL	**67.4**	**63.0**	**64.4**	24.4%

Table 6.10. Results of the idiom identification system in terms of Precision (P), Recall (R) and Macro-F1 (F1) scores on the four test languages. "% Seen" represents the percentage of idioms already encountered in the training set. Morphological variations of the same idiom are considered as a unique idiom.

behavior. From the confusion matrix in Figure 6.10, we can observe that the system is able to (almost always) identify idiomatic expressions as such, mainly thanks to their semantic distance from the meaning of the surrounding words. On the other hand, when dealing with literal expressions, the system again correctly predicts the majority of these, but it makes more errors. We attribute this to the fact that the context is often not sufficiently rich to find a strong similarity (i.e. higher than the threshold δ) with the meaning of the individual constituents of the idiomatic expression, and hence to label the expression as literal. Indeed, the lower the value of δ, the higher the number of literal expressions discovered, but the system inevitably classifies more idiomatic expressions as literal.

Multilingual Idiomatic Expression Identification. In the previous paragraph we evaluated the performance of our dual-encoder architecture on the binary *literal or idiomatic* classification task, where the PIE was pre-identified. In this paragraph, instead, we use the refined silver-data produced by the aforementioned dual encoder, and measure the identification capabilities of our idiom identification system on the sequence-labeling task we

	Seen			Unseen		
Language	P	R	F1	P	R	F1
EN	91.9	71.5	79.1	87.2	68.8	75.6
DE	98.7	95.5	97.1	64.5	49.1	53.8
IT	96.5	91.3	93.8	55.9	49.7	52.2
ES	94.9	96.9	95.9	60.2	55.6	56.9

Table 6.11. Results on the "Seen" and "Unseen" test set subsets in terms of token-level Precision (P), Recall (R) and Macro-F1 (F1) scores.

	Dual Encoder?	**F1**	**Δ**
EN	Yes	**77.1**	-
	No	73.6	+ 3.5
DE	Yes	**85.4**	-
	No	81.9	+ 3.5
IT	Yes	**77.6**	-
	No	73.4	+ 4.2
ES	Yes	**64.4**	-
	No	58.3	+ 6.1

Table 6.12. Comparison of the results obtained by training the system on the silver-standard data validated by our dual encoder (Yes) and non validated ones (No).

introduced (Section 6.2.3) by comparing the BIO tags produced with the corresponding gold labels. The results obtained are reported in Table 6.10.

The first thing that catches the eye is that the performances on the O tags are much higher than those on the B and I tags, on all tested languages. However, this is not surprising, owing to the fact that there is a high class imbalance. An interesting result, instead, is that the system achieves an average score of about 76 F1 points, while the percentage of seen entities[14] is only 48.7% on average. This implies that the system is able to generalize, and consequently also to correctly predict unseen idioms. This phenomenon is particularly evident on English and Spanish, where the percentage of seen idioms is very low.

To better highlight the capability of the system to go beyond idioms already seen during training, we also analyze the system performance on the "seen" and "unseen" subsets independently, and report the results in Table 6.11. As we can observe, the system is able to correctly predict the majority of unseen idioms on all tested languages, achieving an F1 score

[14] Seen entities are entities in the test set which have already been encountered in the training set.

	Type	Prediction
DE	Correct ✔	Ich bin nur der Typ, der ***ihr die Stange hält***.
	Correct ✔	Wir haben dieses Geschäft ***von Grund auf*** aufgebaut.
	Wrong ✘	***Durch den Wind*** wurden 27 Häuser in der Region zerstört.
	Wrong ✘	Sei nicht so'ne beleidigte Leberwurst!
EN	Correct ✔	The old horse finally ***kicked the bucket***.
	Correct ✔	Written tests are his ***Achilles' heel***...
	Wrong ✘	Her aunt is a great cook, do you want a ***piece of cake***?
	Wrong ✘	It is difficult, but possible to quit smoking cold turkey.
IT	Correct ✔	Mi sono ***cavato gli occhi*** dopo aver decifrato la grafia farraginosa.
	Correct ✔	Invece di ***decidere su due piedi***, diedi disposizioni a Tom Donilon perché convocasse i delegati...
	Wrong ✘	A quel punto Smith lanciò ***a terra*** un bicchiere.
	Wrong ✘	Non era affatto scontato che Romney rientrasse nei ranghi, visti i suoi rapporti burrascosi con Trump.
ES	Correct ✔	A la inaguración fueron ***cuatro gatos***.
	Correct ✔	El gobierno sigue ***metiendo el dedo en la llaga***.
	Wrong ✘	¿Has visto alguna vez a tu gato ***meter la pata*** en su bebedero?
	Wrong ✘	El agente tiene vista de lince.

Table 6.13. Examples of idioms correctly and wrongly identified by our idiom identification system. Underline represents the target idiomatic expression (if any), while bold + italic represents the predicted idiomatic expression.

of 59.6 points, on average. Moreover, on seen idioms, the system behaves almost perfectly reaching an average score of 91.5 points. We underline that morphological variations of idioms encountered in the training sets are considered as seen idioms. Table 6.6 provides dimensions of the "seen" and "unseen" subsets, for each language.

Then, to further demonstrate the effectiveness of our dual-encoder architecture (Section 6.2.1), we compare the results obtained by training the system on the data produced with and without the validation performed by our dual encoder. The results reported in Table 6.12 highlight an average gap of 4.3 F1-score points between the refined version of the data and the original one, showing how the validation step is fundamental for improving the quality of the annotations, consequently leading the system to a better understanding of idioms.

Finally, the high results in Table 6.10 also prove that, thanks to our renewed task formulation (Section 6.2.3), common sequence-labeling architectures (e.g. those used for NER) can be successfully imported into the idiom identification task, thus enabling knowledge transfer from other research areas.

6.2.7 Qualitative Analysis

Together with the quantitative evaluation provided in Section 6.2.6, we now perform a qualitative analysis of our system. More specifically, in Table 6.13, we provide 4 examples

of system predictions (2 correct and 2 wrong) for each tested language. Although our system proves to be robust over literal PIEs (see Figure 6.10), its most common mistake consists in classifying a PIE used with its literal meaning as idiomatic. This is mainly due to the system bias towards the labels associated to such PIEs during training, e.g. if more than 90% of occurrences of a certain PIE are labeled as idiomatic in the training set, then the system will tend to classify as idiomatic any other of its occurrences in the test set. This result suggests that improvements over the distribution of labels of PIEs are possible. In Table 6.13 we provide an example of one such wrongly labeled PIE for each language. Another commonly observed error, again highlighted in Table 6.13, is that in which unseen idiomatic expressions are not identified by the system. However, as previously demonstrated in Table 6.11, the system is nevertheless able to correctly handle the majority of such cases.

On the other hand, we observe that the system is able to correctly identify both lemmatized and inflected forms of idiomatic expressions, for both seen and unseen ones.

6.2.8 Open Source Data and Models

In order to support advancements in the newly introduced task of Concept and Named Entity Recognition (CNER), we release both datasets and model checkpoints to encourage research in this joint task. The CNER dataset[15] provides extensive annotations covering thousands and concepts and named entities annotated with 29 categories. Additionally, we make available a pretrained CNER model, CNER-base[16], fine-tuned on this dataset to aid NLP researchers and practitioners in performing efficient and accurate concept and named entity recognition.

Concerning idiom identification, we produced an extensive multilingual resource that includes idiomatic expressions annotated across multiple languages and contexts. We release it on GitHub.[17], Additionally, we provide fine-tuned model checkpoints to accurately identify idioms in diverse multilingual texts.

[15] https://github.com/Babelscape/cner
[16] https://huggingface.co/Babelscape/cner-base
[17] https://github.com/Babelscape/ID10M

Chapter 7

Conclusions and Future Works

In this final chapter, we present the closing remarks of this thesis. We first review and summarize the contributions to the field of Information Extraction (IE) that have been introduced and developed throughout this document (Section 7.1). Following this, we discuss potential directions for future work, outlining several avenues we believe to be particularly promising for advancing the field (Section 7.2).

7.1 Summary of Contributions

This thesis has investigated challenges in IE, with a focus on overcoming obstacles to scalability, efficiency, and inclusivity across diverse linguistic contexts. We recall our research statement, introduced in Section 1.5:

> *We aim to address the intra- and inter-linguistic challenges in Information Extraction (IE) by developing novel multilingual resources and methods; our focus includes reducing training times and expanding the reach of IE to handle complex concepts, figurative language, and tail entities, thus paving the way for scalable, widespread adoption of IE systems across diverse linguistic contexts.*

and our main objectives:

1. **Objective 1:** Creating resources for diverse IE subtasks across languages to address data paucity;

2. **Objective 2:** Building robust multilingual IE models capable of achieving consistent, high-quality performance across languages and domains;

3. **Objective 3:** Reducing computational requirements of IE models making IE systems more efficient, accessible, and suitable for large-scale deployment;

4. **Objective 4:** Enhancing IE capabilities for tail and rare entities;

5. **Objective 5:** Expanding the scope of IE beyond named entities allowing IE systems for a more nuanced and comprehensive extraction of meaning from diverse texts.

We now summarize the contributions presented in this thesis, outlining how each contribution has brought us closer to realizing these objectives.

Our journey started in Chapter 3, where we addressed the need for high-quality, multilingual training data for NER. We introduced a novel, language-agnostic approach called WikiNEuRal to generate robust training datasets, improving cross-linguistic performance in NER. Additionally, to tailor NER capabilities to specific domains such as *sports*, *politics*, and *finance*, we devised a domain adaptation strategy that successfully produced domain-specific data, significantly enhancing accuracy in these areas. Finally, we further expanded our method to create fine-grained NER annotations and released accurate, efficient models for multilingual NER containing less than 0.2B parameters. Together, the findings of this chapter significantly advance our progress toward fulfilling **Objectives 1, 2, and 3**.

Chapter 4 shifted focus to entity disambiguation in low-resource settings, where the overreliance on large-scale labeled datasets reduces the accessibility and scalability of such models. Additionally, popular entities are typically overrepresented in these datasets, leading to biased predictions and reduced performance on less common, tail, or rare entities. Therefore, we developed targeted strategies to boost ED performance under constrained resources, achieving significant reductions in the training data requirements (500x) without compromising accuracy. Furthermore, we introduced an innovative technique for enhancing disambiguation capabilities for infrequent entities, which are commonly overlooked in standard models. This approach increased the precision and recall for rare entities while still attaining satisfactory results on popular entities. Together, the two above-mentioned findings met **Objectives 3 and 4**, respectively.

In Chapter 5, we tackled relation extraction by introducing methods for automatic creation of multilingual training datasets and proposing a new evaluation benchmark. Based on the produced data, we released a family of specialized RE models with parameter counts between 0.4 and 0.7 billion, far smaller than modern large language models but capable of accurately extracting hundreds of relation types in 18 languages. These efficient models support high performance and wide-scale applicability of RE, further advancing our progress towards **Objectives 1 and 3** of ensuring effective RE across languages while reducing computational requirements of IE systems.

Lastly, expanding the scope of information extraction, Chapter 6 focused on no or under-investigated tasks, namely Concept Recognition (CR) and idiom identification, respectively. Specifically, we developed streamlined methods for automatic data generation, and, using the produced data, we trained accurate and efficient models capable of identifying more abstract forms of information such as concepts and figurative expressions across languages. Additionally, by framing these tasks as sequence labeling tasks we allow the resulting systems to be seamlessly integrated in standard IE pipelines. With this chapter, we expanded the capabilities beyond entity-centric tasks, matching **Objectives 1, 3, and 5**.

Through these contributions, this thesis has addressed major challenges in multilingual information extraction, achieving a more inclusive, adaptable, and scalable approach across diverse languages and domains.

7.2 Future Works

Although the methodologies proposed in this thesis for silver data creation have proven to be accurate and effective, there remains significant room for improvement. Future research could concentrate on refining these methods to further enhance the quality, consistency, and reliability of the generated annotations. Techniques such as semi-supervised learning and the integration of domain-specific knowledge could lead to more robust methodologies for silver data generation.

Additionally, despite the methodologies developed in this thesis being language-agnostic, applying them to a wider array of languages would further enhance the accessibility and utility of IE systems on a global scale, with approximately 7,000 languages spoken

worldwide and hundreds of languages actively used on the internet. This expansion could facilitate a deeper understanding of linguistic variations and the impact of IE tasks. Future work could also lead to collaborative projects involving native speakers and linguistic experts to create culturally relevant datasets, ensuring that generated annotations are contextually appropriate and effective across different languages. Such efforts would significantly contribute to closing the multilingual gap in IE and fostering equitable access to advanced language technologies.

Finally, broadening the scope of information extracted to cover other complex linguistic elements across languages – such as actions, events, and various forms of figurative language, including metaphors and similes – will be essential. Capturing these rich semantic aspects will enable IE systems to better understand and interpret nuanced meanings in diverse contexts, particularly in domains like literature and social media analysis, where figurative language plays a critical role.

In summary, while this thesis lays a strong foundation for multilingual and cross-domain IE, pursuing these additional dimensions presents exciting opportunities to expand the reach, adaptability, and depth of future IE systems. By addressing the identified areas for improvement, researchers can work towards creating more sophisticated, accessible, and context-aware information extraction systems that better meet the needs of users across diverse linguistic and cultural landscapes.

Bibliography

Tom Aarsen. Spanmarker for named entity recognition.

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. POLYGLOT-NER: massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 586–594. SIAM.

Dimitra Anastasiou. 2010. *Idiom treatment experiments in machine translation*. Cambridge Scholars Publishing.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Elisa Bassignana and Barbara Plank. 2022. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the*

60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 67–83, Dublin, Ireland. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836, Brussels, Belgium. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Casimir Borkowski and Thomas J. Watson. 1967. An experimental system for automatic recognition of personal titles and personal names in newspaper texts. In *COLING 1967 Volume 1: Conference Internationale Sur Le Traitement Automatique Des Langues*.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational*

Natural Language Learning (CoNLL), pages 677–685, Hong Kong, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Cristina Cacciari and Patrizia Tabossi. 1988. The comprehension of idioms. *Journal of memory and language*, 27(6):668–683.

Cristina Cacciari and Patrizia Tabossi. 2014. *Idioms: Processing, structure, and interpretation*. Psychology Press.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval.

Yuxuan Chen, David Harbecke, and Leonhard Hennig. 2022. Multilingual relation classification via efficient and effective prompting. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Online and Abu Dhabi, the United Arab Emirates. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

BNC Consortium et al. 2007. British national corpus. *Oxford Text Archive Core Collection*.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 41–48, Prague, Czech Republic. Association for Computational Linguistics.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The vnc-tokens dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 19–22.

Nicola De Cao. 2024. *Entity Centric Neural Models for Natural Language Processing*. Ph.D. thesis, Universiteit van Amsterdam.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mona Diab and Pravin Bhutada. 2009. Verb noun construction mwe token classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009)*, pages 17–22.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Rafael Ehren. 2017. Literal or idiomatic? identifying the reading of single occurrences of German multiword expressions using word embeddings. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–112, Valencia, Spain. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165.

Fahim Faisal, Yinkai Wang, and Antonios Anastasopoulos. 2022. Dataset geography: Mapping language data to language users.

Samin Fakharian. 2021. *Contextualized embeddings encode knowledge of English verb-noun combination idiomaticity*. Ph.D. thesis, University of New Brunswick.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Zheng Fang, Yanan Cao, Dongjie Zhang, Qian Li, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning.

Afsaneh Fazly and Suzanne Stevenson. 2006. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 337–344, Trento, Italy. Association for Computational Linguistics.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2022. Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2777–2790, Seattle, United States. Association for Computational Linguistics.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023. SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0).

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Waseem Gharbieh, Virendra Bhavsar, and Paul Cook. 2016. A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118, Berlin, Germany. Association for Computational Linguistics.

Raymond W Gibbs Jr. 1992. What do idioms really mean? *Journal of Memory and language*, 31(4):485–506.

Hongyu Gong, Suma Bhat, and Pramod Viswanath. 2017. Geometry of compositionality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

David Graff and Cieri Christopher. 2003. English gigaword ldc2003t05. web download. In *Philadelphia: Linguistic Data Consortium, 2003.*

Ralph Grishman. 2015. Information extraction. *IEEE Intelligent Systems*, 30(5):8–15.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Zhaochen Guo and Denilson Barbosa. 2017. Robust named entity disambiguation with random walks. *Semantic Web*, 9:1–21.

Ben Hachey. 2009. Multi-document summarisation using generic relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 420–429, Singapore. Association for Computational Linguistics.

Reyhaneh Hashempour and Aline Villavicencio. 2020. Leveraging contextual embeddings and idiom principle for detecting idiomaticity in potentially idiomatic expressions. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 72–80.

Amir Hazem, Merieme Bouhandi, Florian Boudin, and Beatrice Daille. 2022. Cross-lingual and cross-domain transfer learning for automatic term extraction from low resource data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 648–662, Marseille, France. European Language Resources Association.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011a. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011b. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wenhao Huang, Qianyu He, Zhixu Li, Jiaqing Liang, and Yanghua Xiao. 2024. Is there a one-model-fits-all approach to information extraction? revisiting task definition biases.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli. 2023. Red$^{\text{fm}}$: a filtered and multilingual relation extraction dataset. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Nancy Ide and Catherine Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of corpus linguistics*, volume 3, pages 1–7. Citeseer.

Filip Ilievski, Piek Vossen, and Stefan Schlobach. 2018. Systematic study of long tail phenomena in entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 664–674.

Andrea Iovine, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Cyclener: an unsupervised training approach for named entity recognition. In *Proceedings of the ACM Web Conference 2022*, pages 2916–2924.

Rubén Izquierdo, Armando Suárez, and German Rigau. 2015. Word vs. class-based word sense disambiguation. *J. Artif. Int. Res.*, 54(1):83–122.

Rubén Izquierdo Beviá, Armando Suárez Cueto, German Rigau Claramunt, et al. 2007. Exploring the automatic selection of basic level concepts.

Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, Andreas Both, and Sören Auer. 2023. Information extraction pipelines for knowledge graphs. *Knowledge and Information Systems*, 65(5):1989–2016.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Taylor Berg-Kirkpatrick. 2021. Investigating robustness of dialog models to popular figurative language constructs.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Valentin Jijkoun, Jori Mur, and Maarten de Rijke. 2004. Information extraction for question answering: Improving recall through syntactic patterns. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1284–1290, Geneva, Switzerland. COLING.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Albert N Katz, Cristina Cacciari, Raymond W Gibbs Jr, and Mark Turner. 1998. *Figurative language and thought*. Oxford University Press.

Sopan Khosla and Carolyn Rose. 2020. Using type information to improve entity coreference resolution. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 20–31, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015a. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma and Jimmy Ba. 2015b. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma and Jimmy Ba. 2015c. Adam: A method for stochastic optimization. In

3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. SemEval-2013 task 5: Evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 39–47, Atlanta, Georgia, USA. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

Phong Le and Ivan Titov. 2019. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2018. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

Dilin Liu. 2017. *Idioms: Description, comprehension, acquisition, and pedagogy*. Routledge.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. Astitchinlanguagemodels: Dataset and methods for the exploration of idiomaticity in pre-trained language models.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, USA.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737, Dublin, Ireland. Association for Computational Linguistics.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, pages 1–8.

Rada Mihalcea and Andras Csomai. 2007. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

David Milne and Ian H Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518.

Amit Mishra and Sanjay Kumar Jain. 2016. A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3):345–361.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014a. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014b. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

David Mueller, Nicholas Andrews, and Mark Dredze. 2020. Sources of transfer in multilingual named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8093–8104, Online. Association for Computational Linguistics.

Grace Muzny and Luke Zettlemoyer. 2013. Automatic idiom identification in Wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1421, Seattle, Washington, USA. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007a. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

David Nadeau and Satoshi Sekine. 2007b. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2023. Transfer learning for low-resource multilingual relation classification. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2).

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Roberto Navigli and Simone Paolo Ponzetto. 2012a. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217 – 250.

Roberto Navigli and Simone Paolo Ponzetto. 2012b. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217 – 250.

Vasudevan Nedumpozhimana and John Kelleher. 2021. Finding bert's idiomatic key. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 57–62.

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

Geoffrey Nunberg, Ivan A Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe,

Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Laurel Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Christopher Re. 2020. Bootleg: Chasing the tail with self-supervised named entity disambiguation.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured

prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Sachin Pawar, Pushpak Bhattacharyya, and Girish Palshikar. 2017a. End-to-end relation extraction using neural networks and Markov Logic Networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 818–827, Valencia, Spain. Association for Computational Linguistics.

Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. 2017b. Relation extraction : A survey.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Francesco Piccinno and Paolo Ferragina. 2014. From TagME to WAT: a new entity annotator. In *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, pages 55–62.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Jakub Piskorski, Lidia Pivovarova, Jan Šnajder, Josef Steinberger, and Roman Yangarber. 2017. The first cross-lingual challenge on recognition, normalization, and matching of named entities in Slavic languages. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 76–85, Valencia, Spain. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012a. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012b. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Lorenzo Proietti, Stefano Perrella, Simone Tedeschi, Giulia Vulpis, Leonardo Lavalle, Andrea Sanchietti, Andrea Ferrari, and Roberto Navigli. 2024. Analyzing homonymy disambiguation capabilities of pretrained language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 924–938, Torino, Italia. ELRA and ICCL.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual entity linking by neural type system evolution.

Dipti Misra Sharma Rajeev Sangal and Anil Kumar Singh, editors. 2008. *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. Asian Federation of Natural Language Processing.

Carlos Ramisch, Aline Villavicencio, Leonardo Moura, and Marco Idiart. 2008. Picking them up and figuring them out: Verb-particle constructions, noise and idiomaticity. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 49–56, Manchester, England. Coling 2008 Organizing Committee.

Delip Rao, Paul McNamee, and Mark Dredze. 2013. *Entity Linking: Finding Extracted Entities in a Knowledge Base*, pages 93–115. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011a. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011b. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical*

Methods in Natural Language Processing, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. GERBIL–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

Susanna Rücker and Alan Akbik. 2023. Cleanconll: A nearly noise-free named entity recognition dataset.

Manfred Sailer and Stella Markantonatou. 2018. *Multiword Expressions: Insights from a multi-lingual perspective*. Language Science Press.

Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat. 2016. SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 546–559, San Diego, California. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Alessandro Seganti, Klaudia Firląg, Helena Skowronska, Michał Satława, and Piotr Andruszkiewicz. 2021. Multilingual entity and relation extraction dataset and model. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1946–1955, Online. Association for Computational Linguistics.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Marco Silvio Giuseppe Senaldi, Yuri Bizzoni, and Alessandro Lenci. 2019. What do neural networks actually learn, when they learn to identify idioms? *Proceedings of the Society for Computation in Linguistics*, 2(1):310–313.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2020. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*.

Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570.

Hamed Shahbazi, Xiaoli Z. Fern, Reza Ghaeini, Rasha Obeidat, and Prasad Tadepalli. 2019. Entity-aware elmo: Learning contextual entity representation for entity disambiguation.

Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1002–1010, Beijing, China. Coling 2010 Organizing Committee.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece. Association for Computational Linguistics.

Caroline Sporleder, Linlin Li, Philip Gorinski, and Xaver Koch. 2010. Idioms in context: The IDIX corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13843–13850.

Laura Street, Nathan Michalov, Rachel Silverstein, Michael Reynolds, Lurdes Ruela, Felicia Flowers, Angela Talucci, Priscilla Pereira, Gabriella Morgon, Samantha Siegel, Marci

Barousse, Antequa Anderson, Tashom Carroll, and Anna Feldman. 2010. Like finding a needle in a haystack: Annotating the American national corpus for idiomatic expressions. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp.

Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Mario Sänger, Samuele Garda, Xing David Wang, Leon Weber-Genzel, Pia Droop, Benedikt Fuchs, Alan Akbik, and Ulf Leser. 2024. Hunflair2 in a cross-corpus evaluation of biomedical named entity recognition and normalization tools. *Bioinformatics*, 40(10).

Nasrin Taghizadeh and Heshaam Faili. 2022. Cross-lingual transfer learning for relation extraction using universal dependencies. *Computer Speech & Language*, 71:101265.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021a. Named Entity Recognition for Entity Linking: What works and what's next. In *Findings of the*

Association for Computational Linguistics: EMNLP 2021, pages 2584–2596, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021b. Named Entity Recognition for Entity Linking: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2584–2596, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021c. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. NER4ID at SemEval-2022 Task 2: Named Entity Recognition for Idiomaticity Detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003a. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003b. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien

Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 219–228, Berlin, Germany. Association for Computational Linguistics.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. 2014. AGDISTIS - Agnostic disambiguation of named entities using linked open data. In *ECAI*, volume 2014, pages 1113–1114. Citeseer.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: an entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2197–2200.

Rakesh Verma and Vasanthi Vuppuluri. 2015. A new approach for idiom identification using meanings and the web. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 681–687, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Zhigang Wang, Zhixing Li, Juanzi Li, Jie Tang, and Jeff Z. Pan. 2013. Transfer learning based cross-lingual knowledge extraction for Wikipedia. In *Proceedings of the 51st*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 641–650, Sofia, Bulgaria. Association for Computational Linguistics.

Leon Weber, Mario Sänger, Jannes Münchmeyer, Maryam Habibi, Ulf Leser, and Alan Akbik. 2021. HunFlair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. *Bioinformatics*, 37(17):2792–2794.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models.

Jake Williams, Abel Tadesse, Tyler Sam, Huey Sun, and George D. Montanez. 2020. Limits of transfer learning.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281, Hong Kong, China. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Shaodian Zhang and Noémie Elhadad. 2013. Unsupervised biomedical named entity recognition. *J. of Biomedical Informatics*, 46(6):1088–1098.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.