FACULTY OF ENGINEERING AND ARCHITECTURE

Neural Approaches to Entity-Centric Information Extraction

Klim Zaporojets

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors
Prof. Chris Develder, PhD - Prof. Thomas Demeester, PhD
Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

December 2022

GHENT UNIVERSITY

ISBN 978-94-6355-663-7
NUR 984
Wettelijk depot: D/2022/10.500/104

Members of the Examination Board

Chair

Prof. Filip De Turck, PhD, Ghent University

Other members entitled to vote

Prof. Isabelle Augenstein, PhD, University of Copenhagen, Denmark
Prof. Tom Dhaene, PhD, Ghent University
Prof. Veronique Hoste, PhD, Ghent University
Pasquale Minervini, PhD, University College London, United Kingdom
Prof. Yvan Saeys, PhD, Ghent University

Supervisors

Prof. Chris Develder, PhD, Ghent University
Prof. Thomas Demeester, PhD, Ghent University

Acknowledgements

First and foremost, I am extremely grateful to my supervisors Prof. Chris Develder and Prof. Thomas Demeester for giving me the great opportunity to pursue my PhD studies and for their continued support and encouragement. For their patience, assertive advice, and for being on my side and not giving up on me even in the hardest moments throughout my PhD journey, specially during the initial years when I was struggling with rejections and getting the first work published. Finally, I also want to thank my advisors and the IDLab and UGent administration in general for allowing me to pursue my doctoral studies despite working remotely and spending much of the time with my family in Denmark.

I would also like to express special gratitude to ir. Johannes Deleu for all the hours and hours of brainstorming and discussion sessions together. I am sure that without his guidance, this thesis would not have been possible. In fact, it would not be an overstatement to acknowledge him as the initial generator of the main ideas and conceptualizations this thesis is based upon. I also would like to thank him for sharing the initial code of his models, so I could experiment and extend it for each of the different projects I was involved in.

My special gratitude goes to the members of my PhD thesis committee, Prof. Filip De Turck, Prof. Veronique Hoste, Prof. Isabelle Augenstein, Prof. Yvan Saeys, Prof. Tom Dhaene, dr. Pasquale Minervini for dedicating time and effort to read my thesis and to provide constructive feedback.

I would like to express my deepest appreciation to all of the current and former members of our group. I feel extremely lucky and blessed to have been part of such an amazing team: Dr. Lucas Sterckx, Dr. Giannis Bekoulis, Dr. Nasrin Sadeghianpourhamami, Dr. Matthias Strobbe, Dr. Fréderic Godin, Dr. Cedric De Boom, Semere Kiros Bitew, Amir Hadifar, Manu Lahariya, Maarten De Raedt, François Remy, Sofie Labat, Yiwei Jiang, Ruben Janssens, Jens-Joris Decorte, Karel D'Oosterlinck, Paloma Rabaey and Cédric Goemaere.

I am also extremely grateful to all the collaborators I worked with at Ghent University as well as at other institutions. I would like to particularly acknowledge all the members of the CopeNLU team at University of Copenhagen, where I interned during the Spring 2022. Specially I would like to express enormous gratitude to Prof. Isabelle Augenstein and dr.

Lucie-Aimée Kaffee, whose guidance was critical to get our temporal entity linking paper accepted to NeurIPS 2022 conference.

I would also like to recognize a huge effort of all the master students I worked with. Specially, I am deeply indebted to MSc Severine Verlinden, whose hard work and dedication has resulted in the publication at the top ACL conference of the work on injecting external knowledge to text, presented in one of the chapters of this thesis. I also want to express a huge gratitude to MSc Vincent Schelstraete and MSc Nick Peeters.

Finally, I can not end this section without getting too personal and acknowledging my family for all the love and support during my PhD! I could not have undertaken this journey without the support of my wife and kids. Despite living in another country and having to spend many days, weeks or even months without me, they unconditionally encouraged me to pursue my studies. I would also like to thank from the bottom of my heart to my parents-in-law Chris and Ria, who always made me feel welcome in their house, and are now also giving refuge to my aunt and cousin who escaped the war in Ukraine. Finally, I am extremely grateful to my parents and my family in Argentina who, despite living in another continent, have supported and motivated me throughout all of my PhD.

Ghent, Fall 2022
Klim

Table of Contents

List of Figures

List of Tables

Samenvatting
– Summary in Dutch –

Artificiële intelligentie (AI) heeft een enorme impact op ons dagelijks leven met toepassingen zoals stemassistenten, gezichtsherkenning, chatbots, autonoom rijdende auto's, enz. Natural Language Processing (NLP) is een disciplineoverschrijdende AI en linguïstiek die zich toelegt op het bestuderen van het begrip van de tekst. Dit is een zeer uitdagend gebied vanwege de ongestructureerde aard van de taal met veel dubbelzinnige en hoekgevallen. Zinnen met meerdere betekenissen, zoals 'De geit is klaar om te eten', zijn bijvoorbeeld extreem moeilijk te interpreteren voor computers (en zelfs mensen) zonder aanvullende contextuele kennis. Toch is er de afgelopen jaren een snelle vooruitgang geboekt op het gebied van NLP met zeer nuttige toepassingen zoals automatische tekstvertaling, gespreksagenten, nepnieuws en detectie van haatspraak in onder andere sociale media.

In dit proefschrift behandelen we een zeer specifiek gebied van NLP dat het begrip van *entities* in tekst aanpakt. Het concept van *entity* is erg dubbelzinnig en kan voor verschillende interpretaties vatbaar zijn, afhankelijk van een specifieke toepassing en studiegebied. Het meest klassieke gebruik in Natural Language Processing is om te verwijzen naar *named entity's* om echte of fictieve objecten aan te duiden die worden weergegeven met eigennamen. Typische voorbeelden zijn organisaties (e.g., "Google", "Gent University"), mensen (e.g., "Lionel Messi", "Joe Biden"), karakters (e.g., "Batman", "Superman"), locaties (e.g., "Gent", "Denemarken"), oa. Deze entiteitsaanduidingen in tekst worden gebruikt om te worden verbonden met een rijkere Knowledge Base (KB) zoals Wikipedia. Het gebruik van deze KB's is nuttig om aanvullende informatie te verkrijgen en de applicaties te voorzien van extra kennis die nodig is om de tekst te begrijpen. De lezer kan een meer gedetailleerde inleiding op het onderwerp van dit proefschrift vinden in Hoofdstuk 1. Daar geven we een overzicht van de literatuur en introduceren we alle noodzakelijke concepten om het gepresenteerde werk beter te begrijpen.

We beginnen dit proefschrift (Hoofdstuk 2) met een radicaal andere, *entity-centric* kijk op de informatie in tekst. We stellen dat, in plaats van individuele vermeldingen in tekst te gebruiken om hun betekenis te begrij-

pen, we applicaties moeten bouwen die zouden werken in termen van entiteitsconcepten. Deze entiteitgestuurde benadering houdt in dat alle vermeldingen die naar dezelfde entiteit verwijzen (e.g., "Gent") in coreferentiecluster worden gegroepeerd en de rest van de taken (e.g., . relatieextractie, entiteitskoppeling, etc.) op clusterniveau worden uitgevoerd. Deze benadering heeft het voordeel dat de informatie van alle entiteitsvermeldingen die naar één enkele entiteit in het document verwijzen, wordt benut. Als gevolg hiervan vereist de entiteitsgerichte benadering een weergave op documentniveau van de tekst. Helaas heeft de NLP-gemeenschap geen evaluatie- en trainingsbronnen (dwz datasets) geproduceerd die deze focus op documentniveau zouden hebben voor meerdere taken tegelijk. We pakken deze onderzoekskloof aan door een DWIE-dataset (Deustche Welle Information Extraction) te introduceren waarin we vier verschillende taken op entiteitsniveau annoteren: coreferentieresolutie, entiteitskoppeling, relatie-extractie en benoemde entiteitherkenning. We laten verder zien hoe deze taken elkaar aanvullen in een gezamenlijk informatie-extractiemodel.

In het volgende hoofdstuk van dit proefschrift (Hoofdstuk 3), presenteren we een meer gedetailleerd model over hoe de entiteitsgerichte benadering kan worden gebruikt voor de taak *entity linking*. De *entity linking* bestaat uit het toewijzen van het anker *mentions* in tekst aan doel *entities* die beschrijven ze in een Knowledge Base (KB) (e.g., Wikipedia). In ons werk laten we zien dat deze taak kan worden verbeterd door te overwegen entiteitskoppeling uit te voeren op het coreferentieclusterniveau in plaats van op elk van de vermeldingen afzonderlijk. Door deze aanpak te volgen, is ons gezamenlijke model in staat om de informatie van alle kernvermeldingen tegelijk te gebruiken bij het kiezen van de kandidaat-entiteit. Als gevolg hiervan leidt dit tot consistentere voorspellingen tussen vermeldingen die naar hetzelfde concept verwijzen, met name een verbetering van de prestaties op hoekgevallen die bestaan uit impopulaire vermeldingen.

Ons volgende idee wordt beschreven in Hoofdstuk 4 van dit proefschrift. Daar hanteren we een iets andere benadering met entiteiten: in plaats van puur tekstuele informatie te gebruiken om informatie-extractietaken op te lossen, zoals relatie-extractie, bestuderen we ook hoe de informatie van entiteiten uit Knowledge Base kan worden geïntegreerd. We bereiken een aanzienlijke verbetering van alle geëvalueerde taken door informatie te injecteren van zowel Wikipedia als Wikidata KB's. Bovendien, terwijl de taken die we aanpakken zijn geannoteerd en gedefinieerd op *named entity*-niveau, is de informatie die we in onze tekst injecteren afkomstig van alle bestaande entiteiten die zijn gedefinieerd in de geteste KB's. We vinden dat deze techniek zonder toezicht nog steeds de entiteiten kan detecteren die relevanter zijn voor een bepaalde tekst.

Ten slotte wordt de laatste entiteitgerelateerde bijdrage van dit proefschrift beschreven in Hoofdstuk 5. Daar gaan we nog een stap verder en analyseren we de evolutie van de entiteiten vanuit een tijdsperspectief. Om dit te bereiken, creëren we een nieuwe dataset die bestaat uit 10 jaarlijkse

snapshots van Wikipedia-entiteiten van 2013 tot 2022. We bestuderen verder hoe de taak *entity linking* wordt beïnvloed door (i) wijzigingen van bestaande entiteiten in de tijd, en (ii) creatie van nieuwe opkomende entiteiten.. Verder beperken we onze analyse niet tot het domein van *named bodies*, maar nemen we alle bestaande entiteiten en concepten op die in Wikipedia zijn gedefinieerd. Onze analyse toont een voortdurende afname van de prestaties in de tijd, wat aangeeft dat de entiteiten uit latere versies van Wikipedia moeilijker te ondubbelzinnig zijn dan entiteiten uit eerdere versies. Bovendien laten we zien dat de prestatiedaling vooral scherp is voor entiteiten die aanvullende nieuwe kennis nodig hebben (e.g., nieuwe entiteiten met betrekking tot de COVID-19-pandemie) waarvoor het model niet vooraf is getraind.

Daarnaast omvat dit proefschrift ander onderzoekswerk dat is gepubliceerd in vooraanstaande tijdschriften en conferenties die geen verband houden met het centrale onderwerp van dit proefschrift. Daarom stellen we in appendix Hoofdstuk A voor om terugkerende neurale netwerken te gebruiken om de structuur op vergelijkingsbomen na te bootsen om wiskundige wereldproblemen op te lossen. Met onze aanpak laten we een aanzienlijke verbetering zien. Verder beschrijven we in appendix Hoofdstuk B onze bijdrage aan de gedeelde taak van CLPsych 2018, waarbij we competitieve resultaten behalen met behulp van een ensemble bestaande uit meerdere modellen om depressie en angst te voorspellen in tekstuele enquêtes.

Summary

Artificial Intelligence (AI) has huge impact on our daily lives with applications such as voice assistants, facial recognition, chatbots, autonomously driving cars, etc. Natural Language Processing (NLP) is a cross-discipline of AI and Linguistics, dedicated to study the understanding of the text. This is a very challenging area due to unstructured nature of the language, with many ambiguous and corner cases. For example, sentences with multiple meanings such as "The goat is ready to eat." are extremely hard to interpret for computers (and even for humans) without additional contextual knowledge. Yet, in recent years we have witnessed a rapid progress in the field of NLP with highly useful applications such as automatic text translation, conversational agents, fake news and hate speech detection in social media, among others.

In this thesis, we address a very specific area of NLP which tackles the understanding of *entities* in text. The concept of *entity* is very ambiguous and can be subject to different interpretations depending on a specific application and area of study. The most classical use in Natural Language Processing is to refer to *named entities* denoting real-world or fictitious objects that are represented with proper names. Typical examples include organizations (e.g., "Google", "Ghent University"), people (e.g., "Lionel Messi", "Joe Biden"), fictional characters (e.g., "Batman", "Superman"), locations (e.g., "Ghent", "Denmark"), among others. These entity denotations in text are used to be connected to a richer Knowledge Bases (KB) such as Wikipedia. The use of these KBs is beneficial to get additional information and provide the applications with extra knowledge needed to understand the text. The reader can find a more detailed introduction to the topic of this thesis in Chapter 1. There, we give an overview of the literature and introduce all the necessary concepts to better understand the presented work.

We start this thesis in Chapter 2 with proposing a radically different, *entity-centric* view on the information in text. We argue that, instead of using individual mentions in text to understand their meaning, we need to build applications that would operate in terms of entity concepts. This entity-centric approach involves grouping all the mentions referring to the same entity (e.g., "Ghent") in coreference cluster and perform the rest of the tasks (e.g., relation extraction, entity linking, etc.) on a cluster level.

This approach has the advantage of leveraging the information across all the entity mentions referring to a single entity in the document at once. As a consequence, the entity-centric approach requires a document-level view on the text. Unfortunately, the NLP community has produced no evaluation nor training resources (i.e., datasets) that would have this document-level focus for multiple information extraction tasks. We tackle this research gap by introducing DWIE (Deustche Welle Information Extraction) dataset in which we annotate four different tasks on entity level: coreference resolution, entity linking, relation extraction, and named entity recognition. We further demonstrate the interdependence of these tasks in a joint information extraction model.

In the following Chapter 3, we present a more detailed model on how the entity-centric approach can be used for *entity linking* task. The *entity linking* consists in mapping the anchor *mentions* in text to target *entities* that describe them in a Knowledge Base (KB) (e.g., Wikipedia). In our work, we showcase that this task can be improved by considering performing entity linking on the coreference cluster level instead of on each of the mentions individually. By adopting this approach, our joint model is able to use the information of all the coreferent mentions at once when choosing the candidate entity. As a result, this leads to more consistent predictions among mentions referring to the same concept, especially boosting the performance on corner cases consisting of unpopular mentions.

Our next idea is described in Chapter 4 of this thesis. There, we adopt a slightly different approach involving entities: instead of using purely textual information to solve information extraction tasks such as relation extraction, we also study how the information of entities from a Knowledge Base can be integrated. We achieve significant improvement on all of the evaluated tasks by injecting information both from Wikipedia, as well as from Wikidata KBs. Furthermore, while the tasks we are tackling are annotated and defined on *named entity* level, the information we inject in our text comes from all the existing entities defined in the experimented KBs. We find that this unsupervised technique is still able to detect the entities that are more relevant for a particular text.

Finally, the last entity-related contribution of this thesis is described in Chapter 5. There, we go one step further and analyze the evolution of the entities from temporal perspective. In order to achieve this, we create a new dataset which consists of 10 yearly snapshots of Wikipedia entities from 2013 until 2022. We further study how *entity linking* task is affected by (i) changes of existing entities in time, and (ii) creation of new emerging entities. Furthermore, we do not restrict our analysis to the realm of *named entities*, but incorporate all existing entities and concepts defined in Wikipedia. Our analysis showcases a continual decrease of performance over time, indicating that the entities from later versions of Wikipedia are harder to disambiguate than entities from earlier versions. Additionally, we demonstrate that this decrease of performance is exacerbated on en-

tities requiring additional new knowledge (e.g., new entities related to COVID-19 pandemic) for which the model was not pre-trained.

Additionally, this thesis includes other research work published in top-tier journals and conferences not related to the central topic of this thesis. Thus, in Appendix A we propose to use recursive neural networks to mimic the structure of equation trees to solve mathematical word problems. We showcase a significant improvement using our approach. Furthermore, in Appendix B we describe our contribution to the CLPsych 2018 shared task where we achieve competitive results using an ensemble consisting of multiple models to predict depression and anxiety in textual surveys.

1 Introduction

Natural Language Processing (NLP) has recently gained a lot of attention in society. Formally, NLP is a subfield of Linguistics and Artificial Intelligence (AI) concerned with automatic processing of textual data by computers. It spans a wide range of research areas with high societal impact. For example, research in *information extraction* (IE) [1–4] on extracting the most relevant information from textual documents, supports the construction of robust search engines such as Google that allow millions of people to find useful information on internet. Research in *text classification* allows to automatically divide incoming e-mail in different categories, identify fraudulent profiles in e-commerce sites, detect hate speech in social platforms such as Facebook, etc. The latest advances in *conversational agents* [5], allow to assist people in all variety of daily tasks such as getting medical assistance [6], online shopping [7] and cooking [8, 9]. Development in *psycholinguistics* enables models to accurately predict the chances a person may suffer from depression [10, 11], anxiety [12] or even the inclination to commit suicide [13, 14]. Recent developments in *fake news* [15, 16] and *stance* [17, 18] detection facilitate users to find trustworthy and unbiased information online. This list of NLP areas with applications is far from exhaustive, but should give the reader a good idea of the breadth and high impact of the research in NLP.

In this thesis, we focus on *information extraction* (IE), the sub-field of

NLP that studies the extraction of structured information from unstructured text. Concretely, we study how the information about the *entities* described in text can be extracted and used to solve a number of IE tasks. One can think of entities as concepts that can represent any physical or abstract object. The description of these entities is usually collected in encyclopedias such as Wikipedia,[1] Fandom,[2] DBPedia,[3] etc. Our works in [3] (Chapter 2) and [19] (Chapter 3) deal with a specific type of entities denominated *named entities* that include names of people (e.g., "Joe Biden", "Lionel Messi", "Galileo Galilei"), places (e.g., "Belgium"), organizations (e.g., "Google", "Microsoft", "Ghent University"), etc. As a rule of thumb, named entities are usually written with the first letter in uppercase (i.e., are proper nouns). In the first part of our work we explore the importance of thinking in terms of entities that transcends the written entity *mentions* (i.e., words that refer to a specific entity) in text (Chapters 2 and 3). Furthermore, in [20] (Chapter 4), we focus on exploring how additional entity information can enrich other IE tasks such as coreference resolution (Section 1.1.2), named entity recognition (Section 1.1.1), and relation extraction (Section 1.1.3). Finally, in [21] (Chapter 5) we propose a fundamentally different, *evolutionary* view on the entity linking (Section 1.1.4) task. There, we introduce a new TempEL dataset which consists of entity linking annotations grouped in 10 yearly snapshots. Our experimental results showcase a continual temporal decrease in performance of the EL task: the biggest drop is observed for new entities that require additional world knowledge, non-existing during the pre-training phase of the models.

In this introductory chapter we describe (i) the various information extraction tasks this thesis is about (Section 1.1), (ii) the learning approaches used to train the models that are relevant to this thesis (Sections 1.2–1.4), and (iii) a highlight of the main contributions of our work (Section 1.6) with the list of produced publications (Section 1.7).

1.1 Information extraction tasks

In this thesis, we focus on *information extraction* (IE), which includes tasks used to extract structured information from unstructured text. This structured information is a result of solving multiple diverse tasks on a given piece of text [1, 22–26], many of which go beyond the scope of the current work. In this thesis, we focus on the tasks that allow to identify entities described in text as well as the semantic relations between these entities (see

[1] https://www.wikipedia.org/
[2] https://www.fandom.com/
[3] https://www.dbpedia.org/

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	**Mt. Sanitas** is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states	**Palo Alto** is raising the fees for parking.

Figure 1.1: Example of named entities. Source: [34].

entity-centric approach in Section 1.1.4 for details). The first task consists in identifying and classifying the named mentions such as "Meghan Markle" in the example of our DWIE dataset in Fig. 1.7. This task is called *named entity recognition* (NER) [27–29], and is further described in Section 1.1.1. Once the mentions have been identified, we proceed to group them in clusters, each one referring to one specific entity. For example, the mentions "Meghan" and "Meghan Markle" in Fig. 1.7 are clustered together since they both refer to the same person. This task is known as *coreference resolution* [30, 31], and is further described in Section 1.1.2. We further identify semantic relations between the clusters, such as the relation *spouse_of* between the clusters representing *Meghan* and *Harry*. This task is known as *relation extraction* [32, 33], described in Section 1.1.3. Finally, we connect the mention clusters to the respective entities in encyclopedias (formally known as Knowledge Bases or KB) such as Wikipedia. This task is known as *entity linking* and is further detailed in Section 1.1.4. The remainder of this subsection describes the main characteristics as well as the main datasets used to evaluate the performance of the models for each of these tasks.

1.1.1 Named entity recognition

The task of named entity recognition (NER) consists in finding and classifying named entity mentions in text. A *named entity* mention is a proper name referring to real world objects such as countries, organizations, universities, etc. Named entities tend to be written with the first letter uppercased (see Fig. 1.1). This definition is commonly extended to include mentions denoting dates, times and numerical expressions (e.g., prices). The most widely used dataset to evaluate NER is CoNLL-2003 [35], and consists of 35,089 annotated named entity mentions splitted in four entity types: person (PER), location (LOC), organization (ORG), and miscellaneous (MISC). Other extensively used NER datasets are WNUT 2017 [36], Ontonotes v5 [37] and Few-NERD [38], to mention a few.

Barack Obama nominated *Hillary Rodham Clinton* as *his* secretary of state on *Monday*. *He* ...

Figure 1.2: Example of coreference resolution task composed of three coreference mention clusters, depicted with different colors. Thus, the mentions "he" and "He" are coreferent with "Barack Obama". The rest of the mentions ("Hillary Rodham Clinton" and "Monday") form singleton clusters, each composed of a single mention. Adapted from [39].

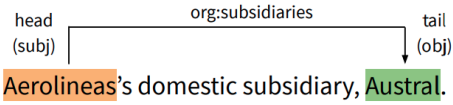


Figure 1.3: Example of relation extraction from TACRED [43] dataset. Source: [44].

1.1.2 Coreference resolution

The coreference resolution task consists in detecting references to the same entity in a text. These references are *mentions* as illustrated in the example of Fig. 1.2, where the mention "He" is coreferent with "Barack Obama". The main dataset to measure the coreference resolution performance is CoNLL-2012 [40] which is part of the OntoNotes corpus [37]. It consists of a set of articles coming from newswire, magazines, broadcast news and conversations, web data, and conversational speech domains. Other datasets in coreference resolution include Task-1 of SemEval 2010 [41] and GAP [42], the latter being a gender-balanced coreference dataset consisting of ambiguous pronouns that have to be resolved to the correct coreferent name.

1.1.3 Relation extraction

The task of relation extraction consists in identifying semantically meaningful relations between two mentions in text. Figure 1.3 illustrates an example from the TACRED dataset [43] of the relation `org:subsidiaries` between the mentions "Aerolineas" and "Austral". This relation denotes that "Austral" (*object* or *tail* of the relation) is a subsidiary of (*relation type*) "Aerolineas" (*head* or *subject* of the relation). The relation types are defined upfront and vary from dataset to dataset. For example, the BC5CDR [45, 46] dataset contains only a single relation type (indicating whether a disease is caused by a particular chemical), while DocRED [47] contains 96 distinct Wikipedia-derived relation types. Other datasets used in relation extraction include ACE 2004 [48], ACE 2005 [49], CoNLL04 [50], SemEval 2010 - Task 8 [51], SciERC [52] and TAC-KBP [53–55].

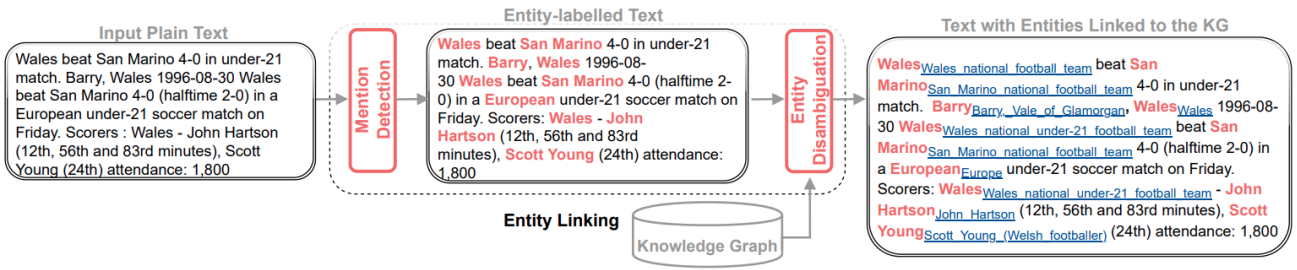


Figure 1.4: Example of entity linking task, which involves two steps: (i) mention detection, and (ii) entity disambiguation. Source: [56].

1.1.4 Entity linking

Entity linking (EL) task consists in mapping a particular mention in text to an entry in a knowledge base (e.g., Wikipedia) which defines the concept (entity) denoted by the mention. Some works [19, 56–59] classify entity linking in two subtasks: mention detection (MD) and entity disambiguation (ED) (see Fig. 1.4), and refer to both of these tasks together as *entity linking*. Yet, most mainstream works [60–64], do not make this distinction and use the concept of *entity linking* as a synonym of entity disambiguation. We go more in detail on these two different settings in Section 1.4 of this chapter, where we describe the difference between *single-task* and *joint* models. Most of the datasets in entity linking use Wikipedia (i.e., wikification; [65]) as the target Knowledge Base to which all the entity mentions are disambiguated.

Most current state-of-the-art EL models [59, 66–69] report on datasets from predominantly the news domain such as AIDA [70], KORE50 [70], AQUAINT [71], ACE 2004, MSNBC [72], N^3 [73], VoxEL [74], and TAC-KBP 2010-2015 [53, 54]. Other frequently used datasets include the web-based IITB [75] and OKE 15/16 [76], as well as the tweet-based Derczynski [77]. Additionally, larger yet automatically annotated datasets such as WNED-WIKI and WNED-CWEB [78] have been also widely adopted. Finally, a number of resources such as the domain-specific biomedical MedMentions [79], the zero-shot ZeShEL [62], and the multi-task DWIE [3] (see Chapter 2) and $AIDA^+$ [19] datasets have been recently introduced. Many of the mentioned datasets are further covered by entity linking evaluation frameworks such as GERBIL [80, 81] and KILT [82] that provide a common interface to evaluate the models.

1.2 Machine learning methodology

In this section we will provide an introductory description of machine learning methodologies to solve the tasks explained in Section 1.1. Due to the sheer amount of architectures to solve the described tasks, we limit our discussion to the ones that are relevant to the main contributions of this thesis. First, we describe the *span-based information extraction* (Section 1.2.1) approach used in the models presented in Chapters 2-4 to detect candidate mention spans in the text. Next, we describe the backbone of the graph propagation algorithm to transfer the local contextual information between these mention spans (see Section 1.2.2). The use of such graph algorithm allows to boost the performance of entity linking tasks by efficiently exchanging information between the mentions spread across the document in Chapter 2. In Section 1.2.3 we describe a novel technique used to efficiently retrieve similar documents given a query commonly known as *dense passage retrieval (DPR)*. We use this method in Chapter 5 of this

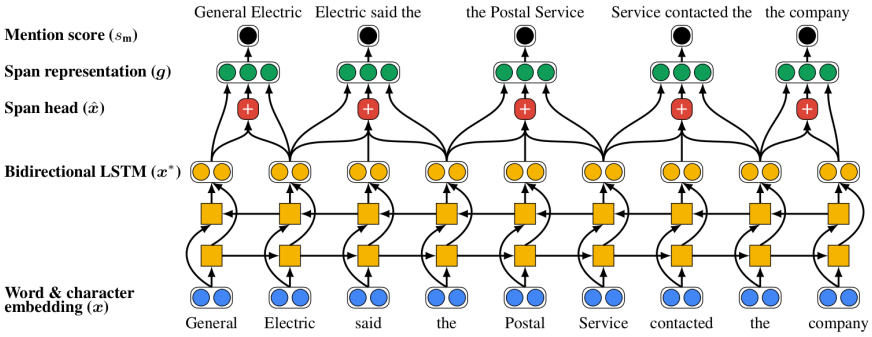


Figure 1.5: Illustration of span-based approach in Coreference Resolution task. Source: [84].

thesis to efficiently retrieve candidate entities given the mention context when tackling temporal entity linking task. Recent work [59, 61, 83], evidences that such fast DPR algorithm is the key backbone component to achieve state of the art results in entity linking task, reason why we use it as a baseline in the Chapter 5 of this thesis. Finally, in Section 1.2.4 we describe two main external knowledge sources, each one containing millions of entities that we use to link to the mentions in text. The first one represents textual-based knowledge bases such as Wikipedia, where each entity is described using plain text. For example, the description of the entity referring to *Meghan Markle* is located at the following Wikipedia url: https://en.wikipedia.org/wiki/Meghan,_Duchess_of_Sussex. The second knowledge source is a structured knowledge graph (KG). Concretely, in this thesis we use Wikidata as such structured knowledge repository which consists of entities, their attributes, and relations between them. The distinction between these two knowledge sources is particularly crucial in the context of Chapter 4, where we combine representations of both of them to obtain major boost in performance as compared to using each of these sources separately.

1.2.1 Span-based information extraction

The *span-based* information-extraction approach has been popularized by the work of [84]. The authors propose, instead of using traditional sequential models such as the ones based on Conditional Random Fields [85–87], to work on all possible token spans of up to a specified length.[4] Such a span-driven approach has a number advantages compared to the sequential models. First, they allow to straightforwardly model the loss function for the information extraction tasks, backpropagating towards the pruner

[4]The length of 10 is enough to cover more than 99% of the mentions in most of the current information extraction datasets.

and the internal LSTM weights. This is because the loss is directly calculated on candidate mention spans and not on intermediate LSTM states as in sequential models. Furthermore, the textual spans provide a natural component to perform further graph propagation as will be explained in Section 1.2.2. Finally, our observations suggest that the span-based models allow to recover more mentions of texts, specially the longer ones consisting of multiple tokens. Figure 1.5 illustrates how the span representation is calculated by concatenating left and right BiLSTM-based token representations. The total number of spans per document of maximum width L can be calculated as:

$$|S| = \sum_{k=1}^{w_{\max}} |T| - k + 1 \; = w_{\max} \left(|T| - \frac{w_{\max} - 1}{2} \right), \tag{1.1}$$

where T represents the total number of tokens in the document, and $w_{\max}$ is the maximum span width. In order to avoid memory overflow, the resulting $|S|$ spans are pruned using a *pruner* component to a manageable fraction of all the tokens in the document. Next, a separate model for a particular IE task is used independently or jointly with the pruner model to predict. The success of span-based approaches [52, 84, 88–91] has also been demonstrated in BERT based models. Thus, [92] introduce SpanBERT, a BERT model pre-trained directly on spans instead of tokens (as is the case of BERT) in text. This model has been successfully used as the backbone to achieve state-of-the-art results in numerous information extraction tasks [93–95]. We use span-based models in our work described in Chapters 2, 3 and 4 of this thesis. Furthermore, we use SpanBERT as the pre-trained model in the architecture described in Chapter 3.

1.2.2 Graph propagation mechanisms

The span representations obtained from the tokens (see Fig. 1.5) can be formalized as follows:

$$\mathbf{g}_i^0 = [\mathbf{e}_l; \mathbf{e}_r; \boldsymbol{\psi}_{r-l}] \tag{1.2}$$

Where $\mathbf{g}_i^0$ is the representation for span s_i, ranging from token l to token r, by concatenating their respective BiLSTM states $\mathbf{e}_l$ and $\mathbf{e}_r$ with an embedding $\boldsymbol{\psi}_{r-l}$ for the span width $w_i = r - l$.

These representations only depend on the underlying BiLSTM states which are inefficient in retaining the context information located further than 50 tokens away [96]. Recent work tackled this problem by using *graph propagation techniques* [97, 98] in span-based models. These techniques are also referred to with the term of *higher order inference (HOI)* [88, 93], and consist in iteratively propagating contextual information between spans. More formally, the propagation operation can be defined as follows:

$$\mathbf{g}_i^{t+1} = \mathbf{f}_x^t(s_i) \odot \mathbf{g}_i^t + \left(1 - \mathbf{f}_x^t(s_i)\right) \odot \mathbf{u}_x^t(s_i), \tag{1.3}$$

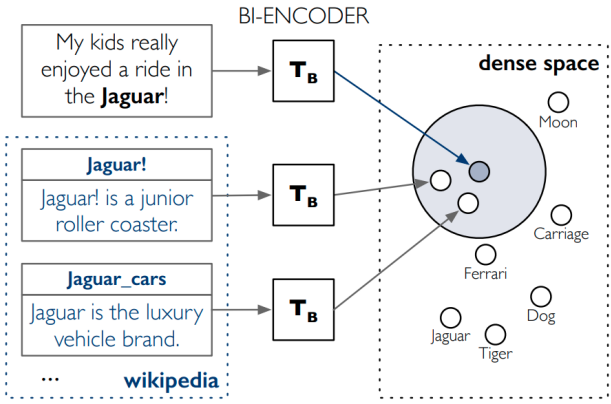


Figure 1.6: Illustration of Dense passage retrieval (DPR) applied on entity linking task. Source: [61].

where the n-dimentional vector $\mathbf{f}_x^t(s_i)$, can be interpreted as a gating vector that acts as a switch between the current span representations $\mathbf{g}_i^t \in \mathbb{R}^n$, and the update span vector $\mathbf{u}_x^t(s_i) \in \mathbb{R}^n$. The various graph propagation methods differ in how $\mathbf{u}_x^t(s_i)$ is calculated. In our work on entity-centric joint information extraction described in Chapter 2, we introduce our own task-independent attention-based graph propagation technique (`AttProp`).

Finally, recently there has been an extensive study evaluating the gains of using HOI on BERT-based (as opposed to BiLSTM-based as described above) coreference resolution models [93]. The authors conclude that, while the coreference resolution models experience additional gains with the incorporation of HOI techniques, it is minimal compared to the gains when using HOI on top of ELMo [99] or LSTM [100] token representation techniques.

1.2.3 Dense passage retrieval

The dense passage retrieval (DPR) concept was introduced by [101] and is used in our work on temporal entity linking (Chapter 5). Concretely, DPR consists in using *dense representations* to match a *query* text with *passages*. The example in Fig. 1.6 showcases the mechanism behind DPR for the entity linking task. Concretely, a set of Wikipedia pages ("Jaguar!", "Jaguar_-cars") are encoded in the same dense space as the mentions ("Jaguar" on top) linked to them. Next, a dot product operation is used to match a query (entity mention with the context) to the Wikipedia entity descriptions in a single dense space. More formally, the similarity operation is defined as follows:

$$\text{sim}(q, p) = E_Q(q)^T E_P(p) \tag{1.4}$$

Where E_Q module encodes the text q in a fixed d-dimensional vector. Similarly, E_P module encodes the candidate passage p into another d-dimensional vector. Most commonly, E_Q and E_P are BERT-based pre-trained encoders, fine-tuned on a specific task (e.g., entity linking).

1.2.4 External knowledge sources

The entities are defined in multiple types of knowledge bases (KBs). A well-known knowledge base is Wikipedia. Each of the entities is described in the corresponding Wikipedia page. For example, the entity *Ghent University* is described on the Wikipedia page https://en.wikipedia.org/wiki/Ghent_University. Yet, KBs such as Wikipedia provide only a textual description of entities. Inherently, computers are not adapted to interpret information in this unstructured textual format. As a result, the research community has recently shown a growing interest in representing the information in a structured manner by means of knowledge graphs (KGs) [102–104]. In a KG, a particular node representing an entity is connected to other nodes as well as associated with certain attributes that describe it. One of the most well-known Knowledge Graphs is Wikidata [105].[5] This knowledge graph interconnects millions of entities[6] using edges of different types. For example, the entity *Ghent University* is connected to the entity *Belgium* by the edge of type *country*, and with the node *William I of the Netherlands* by the edge of type *founded by*. In our work described in Chapter 4 we rely on both the textual Wikipedia KB as well as the structured Wikidata KG to obtain robust entity representations that are used to inject additional knowledge in information extraction models.

1.3 Entity-centric approach

The term *entity-centric* is widely used and is key to understand some of the main contributions of this thesis. The main goal of entity-centric approaches is to encourage to develop of models that reason in terms of entities (concepts) instead of on individual entity mentions in text. One specific type of entities we work with in Chapters 2–4 are named entities, which include all entities denoted with proper names (e.g., names of people, companies, countries, etc.). Figure 1.7 illustrates an example from the entity-centric DWIE dataset, which will be introduced formally in Chapter 2. The left part of the figure depicts the text that is annotated. The right part of the figure illustrates the corresponding structured entity-centric representation where each of the information extraction (IE) annotations are done on entity level. Each such entity may have multiple coreferent

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

[6] At the moment of current writing, Wikidata contains more than 97 million entities.

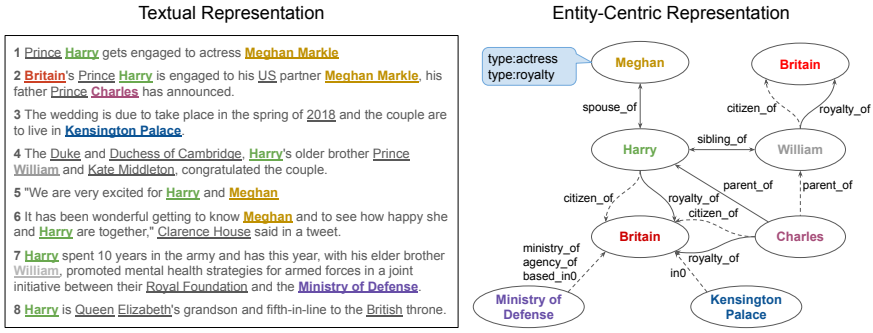


Figure 1.7: An example taken from the DWIE dataset (see Chapter 2) to illustrate the *entity-centric* approach. Each of the entity mentions in the full *document text* of the left is identified and clustered into entities represented in the graph in the right part of the figure. The color of the mentions in text indicates the entity they represent. The remaining annotations such as entity and relation types as well as entity linking (not shown in the figure), are performed on entity level (i.e., are *entity-centric*).

entity mentions in the text. This approach allows to summarize the information of the whole document (e.g., entity and relation types) in a single graph. Furthermore, the document-level perspective of entity-centric annotations also enables to extract information that is not explicitly mentioned in text, but rather can be deduced from the content of the document (represented by dashed arrows in Fig. 1.7). This contrasts with mention-driven [48, 49, 53–55, 106–108] annotations that rely on specific and explicit textual triggers. In this thesis, we are interested in the following entity-centric annotations:

1. **Coreferent entity mentions**: the coreferent entity mentions are grouped in entity mention *clusters*, each one representing a single entity (right part of Fig. 1.7). This clustering of mentions is addressed by *coreference resolution* IE task (see Section 1.1.2), and is tackled by models introduced in Chapters 2–4 of this thesis.

2. **Relation types**: indicate semantically meaningful relations between identified entities. Two entities can be connected by multiple relation types (i.e., the relation annotations are *multilabel*), such as *based_in* and *ministry_of* between entities *Ministry of Defense* and *Britain* in Fig. 1.7. The prediction of these types is addressed by the *relation extraction* IE task (see Section 1.1.3), and is tackled by models introduced in Chapters 2 and 4 of this thesis.

3. **Entity types**: describe the main characteristics of a particular entity. Similar to relation types, the entity types are multilabel (i.e., a particular entity can be associated with multiple entity types). For ex-

ample, in Fig. 1.7 the entity *Meghan* is described with entity types *type:actress* and *type:royalty*. The predictions of entity types is addressed by the named entity recognition (NER) IE task (see Section 1.1.1), and is tackled by models introduced in Chapters 2 and 4.

4. **Entity linking** (not shown in the Fig. 1.7): each of the entities are connected (linked) to the respective entity in the Wikipedia Knowledge Base. For example, the entity Meghan of the right part of Fig. 1.7 is linked to the page *Meghan,_Duchess_of_Sussex* in Wikipedia. The prediction of these links is addressed by the *entity linking* IE task (see Section 1.1.4), and is tackled by models introduced in Chapter 3 using an entity-centric approach, and in Chapter 5 in a more traditional mention-driven setting (i.e., predicting entity links for each of the mentions separately).

1.4 Learning in NLP tasks

In this section we will describe three approaches to solve information extraction (IE) tasks relevant to our thesis. We will use the example in Fig. 1.8, which depicts two subtasks necessary to fully solve the entity-linking task (i.e., with unannotated documents given as input) (see also Section 1.1.4). The first subtask is *mention detection* (MD), to detect all the mentions in text to be linked to a Knowledge Base. In most datasets [3, 19, 70, 73, 109], these mentions are limited to named entity mentions (i.e., proper nouns). The second subtask is *entity disambiguation* (ED), whose goal is to link each of the detected mentions by MD to entities in the Knowledge Base (KB). Both of these subtasks are needed to solve entity linking (EL) task starting from an unannotated plain document input (i.e., setups depicted in Fig. 1.8(a) and Fig. 1.8(c)).

1.4.1 Single task learning

Traditionally, IE architectures tackling multiple (sub-)tasks were solved using a pipelined approach as depicted in the example in Fig. 1.8(a). For instance, [70, 110–112] propose an entity linking pipeline that first performs mention detection and then entity disambiguation. While straightforward to implement, this setup suffers from at least two drawbacks: (i) sequential error accumulation of models executed in the pipeline, and (ii) inability to leverage possible inter-relations between tasks (e.g., knowing that a particular textual span can be linked to a KB can help the mention detection component).

Figure 1.8(b) depicts another single task learning setting. A set of ground truth annotations (e.g., a selection of ground truth mentions in text) are given to the model as input, and it only has to perform a specific task (i.e.,

Figure 1.8: Figure showcasing an example of *entity detection* and *entity disambiguation* information extraction (IE) tasks in order to illustrate three IE learning setups: (a) pipelined setup where separate models are used sequentially for each of the tasks in order to produce the desired output given a *plain document* (i.e., without annotations) as input, (b) single task setup, where the ground truth annotations (i.e., mention(s) to disambiguate in the showcased example) are already given as input to the model, (c) *end-to-end* joint setup where a single model is trained and evaluated jointly on multiple tasks necessary to produce the desired output given a *plain document* as input.

entity disambiguation in Fig. 1.8(b)). While not solving the entity linking task completely, this approach has at least two advantages: (i) it alleviates the study of the performance of a single component responsible for a particular task, and (ii) it opens the possibility to filter specific ground truth annotations to be fed to the input model. This latter point is exploited in recent entity disambiguation works [62, 63, 113, 114] to test the performance of state-of-the-art entity linking models on more challenging entity mentions (e.g., entity mentions linked to unpopular entities). Yet, this may also present the disadvantage of not reflecting the performance in a *real-world* scenario, which is dominated by trivial mentions (e.g., mentions linked to most popular entities) as demonstrated by [78]. In Chapter 5, we use this setup to create temporally evolving *entity disambiguation* dataset where we filter out the trivial mentions (e.g., mentions whose surface form is the same as that of the title of linked entity) in order to focus on most challenging cases.

1.4.2 Joint learning

More recently, there has been a shift towards creating end-to-end models able to be trained and evaluated on multiple tasks jointly. These models share a single neural net architecture, and typically a single loss function is used during their training. The main advantage of such models is an easier deployment since they do not require coupling of multiple components in a pipeline (see Fig. 1.8(a)). Consequently, they do not suffer from error propagation from one model to the next one, characteristic of the pipelined approach. For instance, recent entity linking architectures [19, 58, 59, 68, 83, 115] model both the mention detection (MD) and entity disambiguation (ED) tasks jointly as illustrated Fig. 1.8(c). This contrasts with entity linking models focusing exclusively on the ED task [60–64, 66, 67, 116], taking as input the ground truth mentions as illustrated in Fig. 1.8(c). Additionally, related work [3, 19, 87, 117–119] has also shown an improvement in performance when combining multiple IE tasks in a joint model. This is explained by the interdependence between tasks, where the information from one task can benefit other task(s). For example, knowing the type of a particular entity mention (NER task) can help the model to restrict the entities this mention can be potentially linked to in a KB (EL task) [117].

From a more detailed perspective, we distinguish between *multi-task* and *joint* learning. The *multi-task* learning is characterized by using one or more related tasks from single or separate datasets to train the model. Normally these tasks share some commonalities which act as regularizers to the shared weights of neural model [120] or label spaces [121]. A typical example are language models such as BERT [122] that are pre-trained on predicting tokens, which has a big impact on a huge number of natural language processing tasks such as coreference resolution [92], question answering [123], relation extraction [124], etc. In this thesis, we use purely multi-tasking approach in Chapters 2 and 4 by combining the summing the losses of each of the tasks to obtain the final loss. Empirically, this setup has positive effect with a boost in performance for relation extraction task in Chapter 2. Furthermore, we use the term of *joint* to refer to multi-task models strictly trained on a single dataset, such as the models in Chapters 2 and 4. We also include in the category of *joint* architectures that not necessarily share the neural network weights, but rather connect different tasks in structured way. This is the example of the joint model introduced in Chapter 3 where we connect the coreference and entity linking tasks in a single structured task using a single loss function.

In this thesis, we use *joint* learning to perform cluster-level predictions inherent to the entity-centric approach (see Section 1.3) on datasets such as DWIE (Chapter 2) and AIDA+ (Chapter 3). Concretely, in Chapter 2 we experiment with a joint loss which consists of a sum of losses for each of the tasks the model is trained on. We discover that this setup, besides al-

lowing to make cluster-level predictions using a single end-to-end model, also leads to a significant improvement in performance on the relation extraction task. This indicates that in the proposed joint model, the relation extraction task benefits from the information contained in the rest of the tasks such as NER and coreference resolution (coref). In Chapter 3, we go one step further and frame entity linking (EL) and coref tasks as a single structured task. This allows us to have a single loss term for both of the tasks. We show that this joint approach leads to a general improvement of both coref and EL tasks of up to +5% F1-score. Furthermore, our joint model is also able to solve corner cases of EL task with an improvement of up to +50% in accuracy compared to the standalone EL model.

1.5 Temporal outline of the research

The content of the current PhD thesis with the main focus on entity-centric information extraction was not evident at the beginning of the PhD (almost five years ago). Chronologically, our first work [10] is described in Appendix B, and tackles the shared task in CLPsych 2018 workshop. The proposed architecture achieves competitive results in predicting the various metrics used to measure depression and anxiety based on the content of textual surveys. Next, towards the end of 2018, our interest shifted to investigate the limits and shortcomings of machine learning models to reason on the facts described in Wikipedia entities. Concretely, we focused on studying the results of then state-of-the-art models used in recently introduced FEVER [15] shared task challenge. This challenge consists in verification of textual claims using entity descriptions from Wikipedia. Our analysis of the results showed the difficulty of all the models when reasoning with sentences involving numbers. This inspired us to pursue the work [125] where we propose an innovative structural approach to solve arithmetic word problems which involve multiple numbers, given a textual description of a problem (see Appendix A).

In parallel, during 2018 and 2019, we started annotating DWIE (Deutsche Welle corpus for Information Extraction) dataset as part of CPN project[7]. This resulted in our work [3] described in Chapter 2. The main challenge we faced when working on this project was to design an architecture that would exploit the complementary information across the mentions referring to the same entity (entity-centric approach) in the document. In our baseline model presented in Chapter 2 we succeed in harnessing this interrelated cross-mention information passing by applying graph propagation techniques, which resulted in a significant boost in performance. The ideas for our next two works described in Chapters 3–4 were conceived almost simultaneously in 2020. The first idea [19] described in Chapter 3 con-

[7] https://www.projectcpn.eu/

sists in framing entity linking and coreference resolution tasks in a single structural architecture with a single loss function. Conversely, the idea [20] presented in Chapter 4 intends to exploit the entity information in external Knowledge Bases to use it jointly with the textual mentions entities to boost the performance of various information extraction tasks.

Finally, the work [21] described in Chapter 5 is a result of a joint effort between CopeNLU[8] and T2K[9] research groups and was done during the research visit to University of Copenhagen under the supervision of prof. Isabelle Augenstein in spring of 2022. The idea of working with dynamically evolving entity linking task originated and was shaped during numerous online brainstorm meetings in the course of the winter of 2021-2022. Our main goal was to tackle what, based on our experience when working on Chapters 2–4, we thought was one of the main limitations of the currently available entity linking datasets: they are bound to entities and mentions created at a specific point in time. As a result, these datasets are unable to measure the effect of temporal evolution on entity linking task. Our first plan to tackle this problem was to create a dataset by splitting the news documents in our DWIE dataset (Chapter 2) according to their publication year, linking the mentions to the corresponding Wikipedia version. Yet, this would have produced a very low number of annotated mentions in each of the temporal snapshots, with a high popularity bias (i.e., dominated by most popular entities). As a consequence, we decided to make use of a much larger human-annotated corpus, namely Wikipedia, in order to produce a more challenging large-scale temporally evolving entity-linking dataset: TempEL, described in Chapter 5.

1.6 Research contributions

In this section, we present an overview of the main research contributions of this thesis. We organize the addressed research problems in chapters, each one tackling clearly defined research questions. Table 1.1 summarizes the information extraction tasks and contributions of each of the chapters. In Chapters 2–4 we explore how an entity-centric approach can further boost the performance of information extraction tasks compared to baseline mention-centric architectures, with special focus on multi-task joint models (Section 1.4.2). Conversely, in Chapter 5 we explore the performance of EL solutions for entities as they evolve over larger timeframes Below, we provide a summary of the contributions of each of the chapters:

- In Chapter 2 we propose a radically different, *entity-centric* view on the information in text. We argue that, instead of using individual

[8] https://www.copenlu.com/
[9] https://ugentt2k.github.io/

Chapter	Task	Contribution
2	Multi-task information extraction.	Define multi-tasking IE dataset and baselines.
3	Joint coreference and entity linking.	Joint coreference and entity linking architecture that conceives both tasks in a single structured representation.
4	Named entity recognition, coreference resolution and relation extraction.	New neural architecture that combines textual spans with entity representations from external Knowledge Bases.
5	Entity linking.	New dataset that allows to track the temporal evolution of entities in Wikipedia and the impact of temporal shift on entity linking task.

Table 1.1: Overview of the contributions presented in this thesis.

mentions in text to understand their meaning, we need to build applications that would operate in terms of entity concepts. This entity-centric approach involves grouping all the mentions referring to the same entity (e.g., "Ghent") in a single coreference cluster and perform the rest of the tasks (e.g., relation extraction, entity linking, etc.) on this cluster level. Our approach has the advantage of leveraging the information across all the entity mentions referring to a single entity in the document at once. As a consequence, the entity-centric approach requires a document-level view on the text. Yet, the NLP community has produced no evaluation and training resources (i.e., datasets) that would have this document-level focus for multiple tasks at once. We tackle this research gap by introducing DWIE (Deustche Welle Information Extraction) dataset in which we annotate four different tasks on entity level: coreference resolution, entity linking, relation extraction, and named entity recognition. We further demonstrate how these tasks complement each other in a joint information extraction model.

- In Chapter 3, we develop an entity-centric architecture to make entity linking predictions directly on the entity cluster level instead of on each of the entity mentions separately. To do so, we frame the coreference (coref) and entity linking (EL) tasks as a single structured task. This contrasts with previous attempts to join coref+EL tasks [126–128], where both of the models are trained separately and additional logic is required to merge the predictions of coref and EL tasks. Concretely, in this chapter we contribute with: (i) 2 architectures for joint entity linking (EL) and corefence resolution, (ii) an ex-

tended AIDA dataset [70], adding new annotations of linked and NIL coreference clusters, (iii) experimental analysis on 2 datasets where our joint coref+EL models achieve up to +5% F1-score on both tasks compared to standalone models. We also show up to +50% in accuracy for hard cases of EL where entity mentions lack the correct entity in their candidate list.

- In Chapter 4, we explore how the entity knowledge contained in external Knowledge Bases (KBs) can be injected in text to further enhance the performance of three information extraction (IE) tasks: named entity recognition, coreference resolution, and relation extraction. Furthermore, we analyze what KB representation is more beneficial for these IE tasks: either *KB-graph* trained on Wikidata, or *KB-text* trained directly on Wikipedia. We particularly contribute with (i) a first span-based end-to-end architecture incorporating KB knowledge in a joint entity-centric setting, exploiting unsupervised entity linking (EL) to select KB entity candidates, (ii) exploration of prior- and attention-based mechanisms to combine the EL candidate representations into the model, (iii) assessment of the complementarity of KB-graph and KB-text representations, and (iv) consistent gains of up to 5% F1-score when incorporating KB knowledge in 3 document-level IE tasks evaluated on 2 different datasets.

- In Chapter 5 we propose a fundamentally different, *evolutionary* view on the entity linking (see Section 1.1.4) task. There, we introduce a new, TempEL dataset which consists of Wikipedia entity linking annotations grouped in 10 yearly snapshots. We further contribute with a study of how *entity linking* task is affected by (i) changes of existing entities in time, and (ii) creation of new emerging entities. Our experimental results showcase a continual temporal decrease in performance of the EL task, with the biggest drop for new entities that require additional world knowledge non-existing during the pre-training phase of the models.

Additionally, this thesis appendices' include other research work published in top-tier journals and conferences not related to the central contribution summarized in Table 1.1. Thus, in [125] (see Appendix A) we propose to use recursive neural networks to mimic the structure of equation trees to solve mathematical word problems. We showcase a significant improvement using our approach over baselines. Furthermore, in [10] (see Appendix B) we describe our contribution to CLPsych 2018 shared task where we achieve competitive results using an ensemble consisting of multiple models to predict depression and anxiety in textual surveys.

1.7 Publications

The research output obtained during this PhD has been published in scientific journals and presented at a series of international conferences and workshops. The following list provides an overview of these publications.

1.7.1 Publications in international journals (listed in the Science Citation Index[10])

[3] **K. Zaporojets**, J. Deleu, T. Demeester, and C. Develder, *DWIE: An entity-centric dataset for multi-task document-level information extraction*. Information Processing & Management. 58: 102563, 2021. (**acceptance rate:** 11%)

[125] **K. Zaporojets**, G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, *Solving Arithmetic Word Problems by Scoring Equations with Recursive Neural Networks*. Expert Systems with Applications. 174: 114704, 2021. (**acceptance rate:** 12%)

1.7.2 Publications in international conferences

[21] **K. Zaporojets**, LA. Kaffee, J. Deleu, T. Demeester, C. Develder, I. Augenstein, *TempEL: A Dataset to Evaluate Temporal Effect on Entity Linking Task*. 2022 Conference on Neural Information Processing Systems Datasets and Benchmarks Track: NeurIPS, 2022.

[129] **K. Zaporojets**, J. Deleu, Y. Jiang, T. Demeester, and C. Develder, *Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution*. 2022 Conference on the Association for Computational Linguistics: ACL, 2022. pp. 778-784. **acceptance rate:** 25.2%

[20] **S. Verlinden***, **K. Zaporojets***, J. Deleu, T. Demeester, and C. Develder, *Injecting Knowledge Base Information into End-to-End Joint Entity and Relation Extraction and Coreference Resolution*. Findings of the Association for Computational Linguistics: ACL-IJCNLP, 2021. pp. 1952-1957. **acceptance rate:** 34.9%

* Equal contribution

[10] **K. Zaporojets**, L, Sterckx, J. Deleu, T. Demeester, C. Develder, *Predicting psychological health from childhood essays: the UGent-IDLab CLPsych*

[10]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: *"A1 publications are articles listed in the Science Citation Index, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper."*

2018 shared task system. 5th Annual Workshop on Computer Linguistics and Clinical Psychology (CLPsych 2018) at NAACL-HLT 2018. pp. 119-125.

1.7.3 Publications in international journals and conferences (not included in this thesis)

[14] S. Bitew, G. Bekoulis, J. Deleu, L. Sterckx, **K. Zaporojets**, T. Demeester, and C. Develder, *Predicting Suicide Risk from Online Postings in Reddit – The UGent-IDLab submission to the CLPysch 2019 Shared Task A*. 6th Ann. Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2019) at NAACL-HLT, 2019. pp. 158-161.

[8] Y. Jiang, **K. Zaporojets**, J. Deleu, T. Demeester, and C. Develder, *Recipe instruction semantics corpus (RISeC): resolving semantic structure and zero anaphora in recipes*. 2020 Conference of the Asia-Pacific Chapter of the Association Computational Linguistics and 10th International Joint Conference on Natural Language Processing: AACL-IJCNLP 2020. pp. 821-826.

[9] Y. Jiang, **K. Zaporojets**, J. Deleu, T. Demeester, and C. Develder, *CookDial: A dataset for task-oriented dialogs grounded in procedural documents*. Applied Intelligence Journal 2022. pp. 1-19.

References

[1] J. L. Martinez-Rodriguez, A. Hogan, and I. Lopez-Arevalo. *Information extraction meets the semantic web: a survey*. Semantic Web, 11(2):255–335, 2020.

[2] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, et al. *Clinical information extraction applications: a literature review*. Journal of biomedical informatics, 77:34–49, 2018.

[3] K. Zaporojets, J. Deleu, C. Develder, and T. Demeester. *DWIE: An entity-centric dataset for multi-task document-level information extraction*. Information Processing & Management, 58(4):102563, 2021. Available from: https://arxiv.org/abs/2009.12626.

[4] F. Corcoglioniti, M. Dragoni, M. Rospocher, and A. P. Aprosio. *Knowledge Extraction for Information Retrieval*. In Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains-Volume 9678, pages 317–333, 2016.

[5] S. Hussain, O. Ameri Sianaki, and N. Ababneh. *A survey on conversational agents/chatbots classification and design techniques*. In Workshops of the International Conference on Advanced Information Networking and Applications, pages 946–956. Springer, 2019.

[6] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. Lau, et al. *Conversational agents in healthcare: a systematic review*. Journal of the American Medical Informatics Association, 25(9):1248–1258, 2018.

[7] R. Bavaresco, D. Silveira, E. Reis, J. Barbosa, R. Righi, C. Costa, R. Antunes, M. Gomes, C. Gatti, M. Vanzin, et al. *Conversational agents in business: A systematic literature review and future research directions*. Computer Science Review, 36:100239, 2020.

[8] Y. Jiang, K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Recipe instruction semantics corpus (RISeC): resolving semantic structure and zero anaphora in recipes*. In AACL-IJCNLP 2020, the 1st Conference of the Asia-Pacific Chapter of the Association Computational Linguistics and 10th International Joint Conference on Natural Language Processing, pages 821–826. Association for Computational Linguistics (ACL), 2020.

[9] Y. Jiang, K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *CookDial: a dataset for task-oriented dialogs grounded in procedural documents*. Applied Intelligence, pages 1–19, 2022.

[10] K. Zaporojets, L. Sterckx, J. Deleu, T. Demeester, and C. Develder. *Predicting psychological health from childhood essays: the UGent-IDLab CLPsych 2018 shared task system.* In 5th Annual Workshop on Computer Linguistics and Clinical Psychology (CLPsych 2018) at NAACL-HLT 2018, pages 119–125. Association for Computational Linguistics, 2018.

[11] A. Trifan, R. Antunes, S. Matos, and J. L. Oliveira. *Understanding depression from psycholinguistic patterns in social media texts*. In European Conference on Information Retrieval, pages 402–409. Springer, 2020.

[12] N. C. Jacobson, D. Lekkas, R. Huang, and N. Thomas. *Deep learning paired with wearable passive sensing data predicts deterioration in anxiety disorder symptoms across 17–18 years*. Journal of Affective Disorders, 282:104–111, 2021.

[13] M. M. Tadesse, H. Lin, B. Xu, and L. Yang. *Detection of suicide ideation in social media forums using deep learning*. Algorithms, 13(1):7, 2019.

[14] S. K. Bitew, I. Bekoulis, J. Deleu, L. Sterckx, K. Zaporojets, T. Demeester, and C. Develder. *Predicting suicide risk from online postings in Reddit: the UGent-IDLab submission to the CLPysch 2019 Shared Task A*. In CLPsych2019, the 6th Annual Workshop on Computational Linguistics and Clinical Psychology at NAACL-HLT 2019, pages 158–161. Association for Computational Linguistics (ACL), 2019.

[15] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. *FEVER: a Large-scale Dataset for Fact Extraction and VERification*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), pages 809–819, 2018. Available from: https://aclanthology.org/N18-1074.

[16] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal. *FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information*. arXiv preprint arXiv:2106.05707, 2021.

[17] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva. *Stance Detection with Bidirectional Conditional Encoding*. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 876–885, 2016.

[18] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel. *A simple but tough-to-beat baseline for the Fake News Challenge stance detection task*. arXiv preprint arXiv:1707.03264, 2017.

[19] K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution*, 2021. arXiv:2108.13530.

[20] S. Verlinden, K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Injecting Knowledge Base Information into End-to-End Joint Entity and Relation Extraction and Coreference Resolution*. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1952–1957, 2021.

[21] K. Zaporojets, L.-A. Kaffee, J. Deleu, T. Demeester, C. Develder, and I. Augenstein. *TempEL: Linking Dynamically Evolving and Newly Emerging Entities*.

[22] D. Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[23] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh. *A Survey on Open Information Extraction*. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3866–3878, 2018.

[24] Z. Nasar, S. W. Jaffry, and M. K. Malik. *Information extraction from scientific articles: a survey*. Scientometrics, 117(3):1931–1990, 2018.

[25] R. Grishman. *Information extraction*. IEEE Intelligent Systems, 30(5):8–15, 2015.

[26] S. Sarawagi. *Information extraction*. Now Publishers Inc, 2008.

[27] D. Nadeau and S. Sekine. *A survey of named entity recognition and classification*. Lingvisticae Investigationes, 30(1):3–26, 2007.

[28] V. Yadav and S. Bethard. *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, 2018.

[29] J. Li, A. Sun, J. Han, and C. Li. *A survey on deep learning for named entity recognition*. IEEE Transactions on Knowledge and Data Engineering, 2020.

[30] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu. *Anaphora and coreference resolution: A review*. Information Fusion, 59:139–162, 2020.

[31] N. Stylianou and I. Vlahavas. *A neural entity coreference resolution review*. Expert Systems with Applications, 168:114466, 2021.

[32] S. Pawar, G. K. Palshikar, and P. Bhattacharyya. *Relation extraction: A survey*. arXiv preprint arXiv:1712.05191, 2017.

[33] S. Kumar. *A survey of deep learning methods for relation extraction*. arXiv preprint arXiv:1705.03645, 2017.

[34] D. Jurafsky. *Speech and Language Processing*. 2018.

[35] E. F. T. K. Sang and F. De Meulder. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 142–147, 2003.

[36] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham. *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, 2017.

[37] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al. *Ontonotes release 5.0 ldc2013t19*. Linguistic Data Consortium, Philadelphia, PA, 23, 2013.

[38] N. Ding, G. Xu, Y. Chen, X. Wang, X. Han, P. Xie, H. Zheng, and Z. Liu. *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, 2021.

[39] A. Rahman and V. Ng. *Supervised models for coreference resolution*. In Proceedings of the 2009 conference on empirical methods in natural language processing, pages 968–977, 2009.

[40] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. *CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes*. In Proceedings of the 2012 Conference on Computational Natural Language Learning, pages 1–40, 2012.

[41] M. Recasens, L. Màrquez, E. Sapena, M. A. Martí, M. Taulé, V. Hoste, M. Poesio, and Y. Versley. *Semeval-2010 task 1: Coreference resolution in multiple languages*. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 1–8, 2010.

[42] K. Webster, M. Recasens, V. Axelrod, and J. Baldridge. *Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns*. Transactions of the Association for Computational Linguistics, 6:605–617, 2018.

[43] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning. *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, 2017.

[44] C. Alt, A. Gabryszak, and L. Hennig. *TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task*. arXiv preprint arXiv:2004.14855, 2020.

[45] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu. *BioCreative V CDR task corpus: a resource for chemical disease relation extraction*. Database, 2016, 2016.

[46] C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu. *Overview of the BioCreative V chemical disease relation (CDR) task*. In Proceedings of the 5th BioCreative Challenge Evaluation Workshop, 2015.

[47] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. *DocRED: A Large-Scale Document-Level Relation Extraction Dataset*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 764–777, 2019.

[48] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel. *The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation*. In Proceedings of the 2004 International Conference on Language Resources and Evaluation Workshop on Linguistics, pages 837–840, 2004.

[49] C. Walker, S. Strassel, J. Medero, and K. Maeda. *ACE 2005 multilingual training corpus*. Linguistic Data Consortium, Philadelphia, 57, 2006.

[50] D. Roth and W.-t. Yih. *A linear programming formulation for global inference in natural language tasks*. Technical report, Illinois Univ at Urbana-Champaign Dept of Computer Science, 2004.

[51] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. *SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals*. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 33–38, 2010.

[52] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi. *Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3219–3232, 2018.

[53] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. *Overview of the TAC 2010 knowledge base population track*. In Proceedings of the 2010 Text Analysis Conference, pages 3–3, 2010.

[54] H. Ji, J. Nothman, B. Hachey, and R. Florian. *Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking*. In Proceedings of the 2015 Text Analysis Conference, 2015.

[55] H. Ji, X. Pan, B. Zhang, J. Nothman, J. Mayfield, P. McNamee, C. Costello, and S. I. Hub. *Overview of TAC-KBP2017 13 Languages Entity Discovery and Linking.* In Proceedings of the 2017 Text Analysis Conference, 2017.

[56] O. Sevgili, A. Shelmanov, M. Arkhipov, A. Panchenko, and C. Biemann. *Neural Entity Linking: A Survey of Models based on Deep Learning*. arXiv preprint arXiv:2006.00575, 2020.

[57] O.-E. Ganea and T. Hofmann. *Deep Joint Entity Disambiguation with Local Neural Attention*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 2619–2629, 2017. Available from: https://www.aclweb.org/anthology/D17-1277.

[58] N. Kolitsas, O.-E. Ganea, and T. Hofmann. *End-to-End Neural Entity Linking*. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 519–529, 2018. Available from: https://www.aclweb.org/anthology/K18-1050/.

[59] W. Zhang, W. Hua, and K. Stratos. *EntQA: Entity Linking as Question Answering*. arXiv preprint arXiv:2110.02369, 2021.

[60] D. Rao, P. McNamee, and M. Dredze. *Entity linking: Finding extracted entities in a knowledge base*. In Multi-Source, Multilingual Information Extraction and Summarization, pages 93–115. Springer, 2013. Available from: https://doi.org/10.1007/978-3-642-28569-1_5.

[61] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. *Zero-shot entity linking with dense entity retrieval*. arXiv preprint arXiv:1911.03814, 2019.

[62] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee. *Zero-Shot Entity Linking by Reading Entity Descriptions*. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3449–3460, 2019.

[63] Y. Onoe and G. Durrett. *Fine-Grained Entity Typing for Domain Independent Entity Linking.* In AAAI, pages 8576–8583, 2020.

[64] J. Raiman. *DeepType 2: Superhuman Entity Linking All You Need Is Type Interactions*. In Proceedings of the 2022 Conference on Artificial Intelligence (AAAI 2022), 2022. Available from: https://www.aaai.org/AAAI22Papers/AAAI-2612.RaimanJ.pdf.

[65] R. Mihalcea and A. Csomai. *Wikify! Linking documents to encyclopedic knowledge*. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 233–242, 2007.

[66] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto. *Global Entity Disambiguation with Pretrained Contextualized Embeddings of Words and Entities*. arXiv preprint arXiv:1909.00426, 2020. Available from: https://arxiv.org/abs/1909.00426.

[67] L. Orr, M. Leszczynski, S. Arora, S. Wu, N. Guha, X. Ling, and C. Re. *Bootleg: Chasing the tail with self-supervised named entity disambiguation*. arXiv preprint arXiv:2010.10363, 2020.

[68] N. De Cao, G. Izacard, S. Riedel, and F. Petroni. *Autoregressive Entity Retrieval*. In Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021), 2020. Available from: https://arxiv.org/abs/2010.00904.

[69] N. De Cao, W. Aziz, and I. Titov. *Highly Parallel Autoregressive Entity Linking with Discriminative Correction*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7662–7669, 2021. Available from: https://aclanthology.org/2021.emnlp-main.604, doi:10.18653/v1/2021.emnlp-main.604.

[70] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. *Robust disambiguation of named entities in text*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pages 782–792, 2011. Available from: https://www.aclweb.org/anthology/D11-1072/.

[71] D. Milne and I. H. Witten. *Learning to link with wikipedia*. In Proceedings of the 17th ACM conference on Information and knowledge management, pages 509–518, 2008.

[72] L. Ratinov, D. Roth, D. Downey, and M. Anderson. *Local and global algorithms for disambiguation to wikipedia*. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pages 1375–1384, 2011.

[73] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both. *N^3-A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format*. In Proceedings of the Ninth

International Conference on Language Resources and Evaluation (LREC'14), pages 3529–3533, 2014.

[74] H. Rosales-Méndez, A. Hogan, and B. Poblete. *VoxEL: a benchmark dataset for multilingual entity linking*. In Proceedings of the 2018 International Semantic Web Conference (ISWC 2018), pages 170–186, 2018. Available from: https://doi.org/10.1007/978-3-030-00668-6_11.

[75] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. *Collective annotation of Wikipedia entities in web text*. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 457–466, 2009.

[76] A. G. Nuzzolese, A. L. Gentile, V. Presutti, A. Gangemi, D. Garigliotti, and R. Navigli. *Open knowledge extraction challenge*. In Semantic Web Evaluation Challenges, pages 3–15. Springer, 2015.

[77] L. Derczynski, D. Maynard, G. Rizzo, M. Van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. *Analysis of named entity recognition and linking for tweets*. Information Processing & Management, 51(2):32–49, 2015.

[78] Z. Guo and D. Barbosa. *Robust named entity disambiguation with random walks*. Semantic Web, 9(4):459–479, 2018.

[79] S. Mohan and D. Li. *MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts*. In Automated Knowledge Base Construction (AKBC), 2018.

[80] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, et al. *GERBIL: general entity annotator benchmarking framework*. In Proceedings of the 24th international conference on World Wide Web, pages 1133–1143, 2015.

[81] M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo. *Gerbil–benchmarking named entity recognition and linking consistently*. Semantic Web, 9(5):605–625, 2018.

[82] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, et al. *Kilt: a benchmark for knowledge intensive language tasks*. arXiv preprint arXiv:2009.02252, 2020.

[83] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, and A. Pierleoni. *ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking*. arXiv preprint arXiv:2207.04108, 2022.

[84] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. *End-to-end Neural Coreference Resolution*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 188–197, 2017. Available from: https://www.aclweb.org/anthology/D17-1018.

[85] A. McCallum and W. Li. *Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons*. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pages 188–191, 2003.

[86] C. Sutton, A. McCallum, et al. *An introduction to conditional random fields*. Foundations and Trends® in Machine Learning, 4(4):267–373, 2012.

[87] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *Joint entity recognition and relation extraction as a multi-head selection problem*. Expert Systems with Applications, 114:34–45, 2018.

[88] K. Lee, L. He, and L. Zettlemoyer. *Higher-Order Coreference Resolution with Coarse-to-Fine Inference*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), pages 687–692, 2018. Available from: https://www.aclweb.org/anthology/N18-2108.

[89] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi. *A general framework for information extraction using dynamic span graphs*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 3036–3046, 2019. Available from: https://www.aclweb.org/anthology/N19-1308/.

[90] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi. *Entity, Relation, and Event Extraction with Contextualized Span Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pages 5788–5793, 2019.

[91] L. He, K. Lee, O. Levy, and L. Zettlemoyer. *Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling*. In Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics, pages 364–369, 2018.

[92] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. *SpanBERT: Improving pre-training by representing and predicting spans*. Transactions of the Association for Computational Linguistics (TACL 2020), 8:64–77, 2020. Available from: https://www.aclweb.org/anthology/2020.tacl-1.5.

[93] L. Xu and J. D. Choi. *Revealing the Myth of Higher-Order Inference in Coreference Resolution*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 8527–8533, 2020. Available from: https://doi.org/10.18653/v1/2020.emnlp-main.686.

[94] L. Xu and J. D. Choi. *Modeling Task Interactions in Document-Level Joint Entity and Relation Extraction*. arXiv preprint arXiv:2205.01909, 2022.

[95] W. Wu, F. Wang, A. Yuan, F. Wu, and J. Li. *CorefQA: Coreference Resolution as Query-based Span Prediction*. In Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics (ACL 2020), pages 6953–6963, 2020. Available from: https://www.aclweb.org/anthology/2020.acl-main.622.

[96] U. Khandelwal, H. He, P. Qi, and D. Jurafsky. *Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context*. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 284–294, 2018.

[97] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. *A comprehensive survey on graph neural networks*. IEEE Transactions on Neural Networks and Learning Systems, pages 1–21, 2020.

[98] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. *Graph neural networks: A review of methods and applications*. AI Open, 1:57–81, 2020.

[99] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. In Proceedings of NAACL-HLT, pages 2227–2237, 2018.

[100] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural computation, 9(8):1735–1780, 1997.

[101] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. *Dense Passage Retrieval for Open-Domain Question Answering*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, 2020.

[102] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip. *A survey on knowledge graphs: Representation, acquisition, and applications*. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[103] C. Gutiérrez and J. F. Sequeda. *Knowledge graphs*. Communications of the ACM, 64(3):96–104, 2021.

[104] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou. *A retrospective of knowledge graphs*. Frontiers of Computer Science, 12(1):55–74, 2018.

[105] D. Vrandečić and M. Krötzsch. *Wikidata: a free collaborative knowledgebase*. Communications of the ACM, 57(10):78–85, 2014.

[106] J. Ellis, J. Getman, and S. M. Strassel. *Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results*. In Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology, pages 17–18, 2014.

[107] J. Ellis, J. Getman, D. Fore, N. Kuster, Z. Song, A. Bies, and S. M. Strassel. *Overview of Linguistic Resources for the TAC KBP 2015 Evaluations: Methodologies and Results.* In Proceedings of the 2015 Text Analysis Conference, 2015.

[108] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma. *From light to rich ere: annotation of entities, relations, and events*. In Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 89–98, 2015.

[109] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. *KORE: keyphrase overlap relatedness for entity disambiguation*. In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 545–554, 2012.

[110] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. *Dexter: an open source framework for entity linking*. In Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval, pages 17–20, 2013.

[111] M. Van Erp, G. Rizzo, and R. Troncy. *Learning with the Web: Spotting Named Entities on the Intersection of NERD and Machine Learning.* In # MSM, pages 27–30. Citeseer, 2013.

[112] F. Piccinno and P. Ferragina. *From TagME to WAT: a new entity annotator*. In Proceedings of the first international workshop on Entity recognition & disambiguation, pages 55–62, 2014.

[113] V. Provatorova, S. Vakulenko, E. Kanoulas, K. Dercksen, and J. M. van Hulst. *Named Entity Recognition and Linking on Historical Newspapers: UvA. ILPS & REL at CLEF HIPE 2020.* In CLEF (Working Notes), 2020.

[114] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy. *Named Entity Disambiguation for Noisy Text*. In Proceedings of the 2017 Conference on Computational Natural Language Learning, pages 58–68, 2017.

[115] S. Broscheit. *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 2019), pages 677–685, 2019. Available from: https://www.aclweb.org/anthology/K19-1063.

[116] E. Barba, L. Procopio, and R. Navigli. *ExtEnD: Extractive Entity Disambiguation*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2478–2488, 2022.

[117] P. H. Martins, Z. Marinho, and A. F. Martins. *Joint Learning of Named Entity Recognition and Entity Linking*. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 190–196, 2019.

[118] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *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, 2018.

[119] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. *Latent multi-task architecture learning*. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4822–4829, 2019.

[120] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. *Natural language processing (almost) from scratch*. Journal of machine learning research, 12(ARTICLE):2493–2537, 2011.

[121] I. Augenstein, S. Ruder, and A. Søgaard. *Multi-Task Learning of Pairwise Sequence Classification Tasks over Disparate Label Spaces*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1896–1906, 2018.

[122] M. Joshi, O. Levy, L. Zettlemoyer, and D. S. Weld. *BERT for Coreference Resolution: Baselines and Analysis*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 5807–5812, 2019. Available from: https://www.aclweb.org/anthology/D19-1588.

[123] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. *QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 535–546, 2021. Available from: https://aclanthology.org/2021.naacl-main.45.

[124] N. Zhang, X. Chen, X. Xie, S. Deng, C. Tan, M. Chen, F. Huang, L. Si, and H. Chen. *Document-level Relation Extraction as Semantic Segmentation*. In IJCAI, 2021.

[125] K. Zaporojets, G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *Solving arithmetic word problems by scoring equations with recursive neural networks*. Expert Systems with Applications, 174:114704, 2021.

[126] H. Hajishirzi, L. Zilles, D. S. Weld, and L. Zettlemoyer. *Joint coreference resolution and named-entity linking with multi-pass sieves*. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), pages 289–299, 2013. Available from: https://aclanthology.org/D13-1029/.

[127] S. Dutta and G. Weikum. *C3EL: A joint model for cross-document co-reference resolution and entity linking*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pages 846–856, 2015. Available from: https://doi.org/10.18653/v1/d15-1101.

[128] R. Angell, N. Monath, S. Mohan, N. Yadav, and A. McCallum. *Clustering-based Inference for Biomedical Entity Linking*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 2598–2608, 2021. Available from: https://doi.org/10.18653/v1/2021.naacl-main.205.

[129] K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution*. arXiv preprint arXiv:2108.13530, 2021.

2

DWIE: an Entity-Centric Dataset for Multi-Task Document-Level Information Extraction

In this chapter we introduce 'Deutsche Welle corpus for Information Extraction', a newly created multi-task dataset that combines four main Information Extraction (IE) annotation subtasks: (i) Named Entity Recognition (NER), (ii) Coreference Resolution, (iii) Relation Extraction (RE), and (iv) Entity Linking. Furthermore, we propose a radically different, entity-centric view on the information in text. We argue that, instead of using individual mentions in text to understand their meaning, we need to build applications that would operate in terms of entity concepts. This approach has the advantage of leveraging the information across all the entity mentions referring to a single entity in the document at once. As a consequence, all the annotations on DWIE are done on coreference concept level. Each of the concepts can group one or more mentions that refer to the same entity in the Knowledge Base. We further demonstrate how these tasks complement each other in a joint information extraction model.

⋆ ⋆ ⋆

K. Zaporojets, J. Deleu, C. Develder and T. Demeester

Information Processing & Management, 2021.

Abstract This paper presents DWIE, the 'Deutsche Welle corpus for Information Extraction', a newly created multi-task dataset that combines four main Information Extraction (IE) annotation subtasks: (i) Named Entity Recognition (NER), (ii) Coreference Resolution, (iii) Relation Extraction (RE), and (iv) Entity Linking. DWIE is conceived as an *entity-centric* dataset that describes interactions and properties of conceptual entities on the level of the complete document. This contrasts with currently dominant *mention-driven* approaches that start from the detection and classification of named entity mentions in individual sentences. Further, DWIE presented two main challenges when building and evaluating IE models for it. *First*, the use of traditional mention-level evaluation metrics for NER and RE tasks on entity-centric DWIE dataset can result in measurements dominated by predictions on more frequently mentioned entities. We tackle this issue by proposing a new entity-driven metric that takes into account the number of mentions that compose each of the predicted and ground truth entities. *Second*, the document-level multi-task annotations require the models to transfer information between entity mentions located in different parts of the document, as well as between different tasks, in a joint learning setting. To realize this, we propose to use graph-based neural message passing techniques between document-level mention spans. Our experiments show an improvement of up to 5.5 F_1 percentage points when incorporating neural graph propagation into our joint model. This demonstrates DWIE's potential to stimulate further research in graph neural networks for representation learning in multi-task IE. We make DWIE publicly available at https://github.com/klimzaporojets/DWIE.

2.1 Introduction

Information Extraction (IE) plays a fundamental role as a backbone component in many downstream applications. For example, an application such as question answering may be improved by relying on relation extraction (RE) [1, 2], coreference resolution [3, 4], named entity recognition (NER) [5, 6], and entity linking (EL) [7, 8] components. This also holds for other applications such as personalized news recommendation [9–11], fact checking [12, 13], opinion mining [14], semantic search [15], and conversational agents [16]. The last decade has shown a growing interest in IE datasets suitably annotated for developing multi-task models where each of the tasks (e.g., NER, RE, etc.) would benefit from the interaction with (an)other task(s) [17–21], to boost their performance. However, the currently widely used IE datasets to build such multi-task models exhibit three major limitations. *First*, the annotation schema adopted in most of these datasets is mention-driven, focusing on annotating elements (e.g., relations, entity types) that involve specific entity mentions explicitly mentioned in the text. This produces very localized annotations (e.g., sentence-

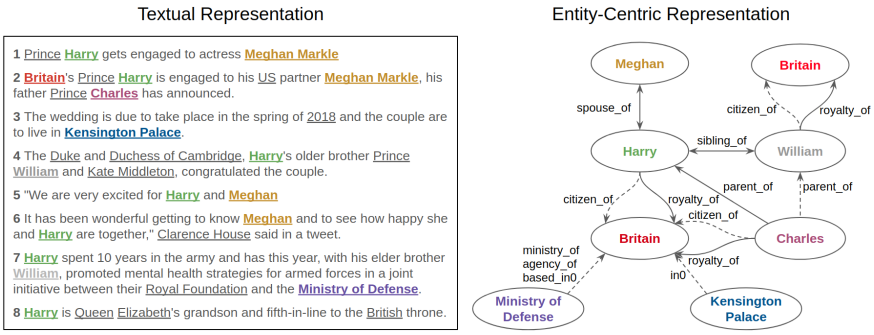


Figure 2.1: An example from the DWIE dataset with entity mentions underlined. We show 8 of the 29 entities in the graph on the right. It illustrates the relations that can be derived from the content of the article. The relations that are explicitly mentioned in the text (trigger-based) are depicted by solid arrows. Conversely, the relations that are implicit and/or need the whole document context (document-based) to be derived are represented by dashed arrows.

based relations between entity mentions) that do not reflect meaning that can be inferred on a more general document-level. *Second*, the number of annotated extraction tasks in most of the IE datasets is rather limited. Most of them focus on a single or at most a few different tasks. Furthermore, some other datasets, including the well-known TAC-KBPs [22–26], use different non-overlapping corpora for each of the tracks that group a few related tasks. Consequently, current models addressing multiple IE tasks together often use multi-tasking (with different datasets per task) rather than really joint modeling approaches. *Finally*, the annotation of currently widely used IE datasets is driven by either relying on a priori defined annotation schemas [27–32] or on distantly supervised labeling techniques [33–37]. In consequence, the resulting annotations are not necessarily representative of the actual information contained in the annotated corpus.

In this work, we tackle the aforementioned limitations of IE datasets by introducing a new dataset named DWIE. It consists of 802 general news articles in English, selected randomly from a corpus collected from Deutsche Welle[1] between 2002 and 2018, as part of the CPN project.[2] We focus on annotating four main IE tasks: (i) Named Entity Recognition (NER), (ii) Coreference Resolution, (iii) Relation Extraction (RE), and (iv) Entity Linking.[3] Figure 2.1 shows an example snippet from the DWIE corpus. We adopt an *entity-centric* approach where all annotations (i.e., for NER, RE

[1] https://www.dw.com
[2] https://www.projectcpn.eu
[3] The linking is done to Wikipedia version 20181115.

and Entity Linking tasks) are made on the entity[4] level. Each of the entities is composed by the coreferenced entity mentions from the entire document (e.g., the entity *Meghan* in Fig. 2.1 clusters the entity mentions "Meghan Markle" and "Meghan" across the whole document). This entity-centric approach contrasts with mention-driven annotations in widely used IE datasets [24, 27, 29, 31–33, 38] where the annotation process is biased towards considering only local explicit textual evidence to annotate elements such as relations and entity types (e.g., the relation spouse_of⟨*Meghan*, *Harry*⟩ that can be extracted from the 1st sentence in Fig. 2.1). Consequently, our DWIE dataset paves the way for research on more complex document-level reasoning that goes beyond only the local textual context directly surrounding individual entity mentions. For example, consider the relation ministry_of⟨*Ministry of Defense*, *Britain*⟩ in Fig. 2.1: while the text of the document does not directly state such a relation, it can be deduced from a more general document-level entity-centric vision of the article, i.e., combining the information involving the entities *Ministry of Defense* and *Harry* in sentence 7 with the one involving *Britain* and *Harry* in sentence 2. Finally, the entity-centric approach provides entity linking annotations that are consistent across the document: by clustering mentions of the same entity, and then providing links to the Wikidata KB (or NIL if the entity does not appear there) for the whole cluster at once, we limit annotation errors or accidental inconsistencies (in the linking itself, but also in terms of NER labels). To our knowledge, DWIE is the first dataset with this level of conceptual consistency over the considered information extraction tasks. We therefore expect that the dataset will play a key role in advancing research exploring potential benefits of (i) entity-level information extraction in terms of reducing potential inconsistent decisions (within EL across multiple mentions, as well as across multiple tasks), and (ii) using entity-centric information stored in a KB to complement the otherwise exclusively text-dependent IE tasks such as NER, RE, and coreference resolution.

Additionally, we use a bottom-up, data-driven annotation approach where we manually define our annotations (e.g., in terms of the entity and relation types) to maximally reflect the information of the corpus at hand. Currently dominant datasets are driven by distant supervision and executed top-down, by which we mean that the selection of entity and relation types is a priori defined and limited in coverage (i.e., the raw data potentially contains other types that thus remain un-annotated). Conversely, we do not a priori limit the entity and relation types to annotate, but adopt a bottom-up approach driven by the data itself. Our proposed bottom-up approach encompasses a three-pass annotation procedure where we use the first exploratory annotation pass to derive the main annotation types (annotation schema) from the corpus, and the next two passes to perform schema-driven annotations and refine them by carrying out an

[4] Also referred to as *entity cluster* or just *cluster*.

additional parallel annotation of the corpus for fixing errors inferred from inter-annotator inconsistencies.

Besides the dataset itself, we also contribute empirical modeling results to address the aforementioned IE tasks. Our goal is to study two important properties that are inherent to DWIE. The first key property is the need for *long-range contextual information sharing* to make document-level predictions involving entities whose mentions are located in different parts of the document. The second key property involves the *joint interaction between tasks* where the information obtained in one task can help to solve another task. For example, in Fig. 2.1 knowing the types of entities (which involves NER and coreference tasks) *Britain* and *Kensington Palace* can boost the performance of the relation extraction task by limiting the number of possible relation types between these two entities (e.g., ministry_of but not citizen_of). In order to study the impact of these two phenomena inherent to our DWIE dataset on the final results, we experiment with neural graph-based models [39–41]. These models allow message passing between local contextual encodings, making it possible to measure the impact of local contextual information sharing both on a more general document level and across the tasks. Furthermore, previous work already has shown the positive effect of using graph-based information passing techniques on single tasks [20, 42], and between tasks [18, 21, 43, 44] on mention-driven datasets. We expand this work even further by extending these models to be used on the entity-centric, document-level DWIE dataset. More specifically, we experiment with both single-task (Section 2.4.5) as well as joint (Section 2.4.2) models to study the effect of contextual information propagation in single task and joint settings. Additionally, for the NER and RE tasks, we propose a new entity-centric evaluation metric that not only considers the predictions on separate entity mentions (as is done in related IE datasets), but also accounts for the impact of the predictions on entity cluster level.

In summary, the main objective that we address in the current paper is to introduce an entity-centric multi-task IE dataset that covers different related tasks on a document level as well as provides a connection with external structured knowledge (through the entity linking task). Furthermore, we aim to explore how neural graph-based models can boost the performance by enabling local contextual information propagation across the document (single-task models) and between different tasks (joint models). The results presented in this paper suggest that, while challenging, DWIE opens up new possibilities of research in the domain of joint entity-centric information extraction methods. The main contributions of our work are that:

(1) We construct a self-contained dataset (Section 2.3) with joint annotations for four basic information extraction tasks (NER, entity linking, coreference resolution, and RE), that provide entity-centric document-level annotations (as opposed to typical mention-driven sentence-

level annotations for, e.g., RE) connecting unstructured (text) and structured (KB) information sources.

(2) We introduce a data-driven, bottom-up three-pass annotation approach complemented by context-based logical rules to build such dataset (Section 2.3).

(3) We propose a new evaluation metric for the NER and RE tasks (Section 2.5), in line with the entity-centric nature of DWIE.

(4) We extend the competitive graph-based neural IE model DyGIE [21] for the four IE tasks in DWIE (Section 2.4) and provide source code for NER, coreference resolution, and RE. Furthermore, we introduce a new latent attention-driven `AttProp` graph propagation method and show its advantages in both single and joint model settings. The experimental results (Section 2.6) demonstrate the potential of such neural graph based models.

2.2 Related work

This section summarizes the overview of related datasets (Section 2.2.1), and explores the differences between our newly created DWIE and other similar datasets widely used by the scientific community. The main qualitative differences are presented in Table 2.1, while the quantitative comparison is provided in Table 2.2. Next, we describe the current trends in IE to solve the tasks included in DWIE, and compare them to our proposed approach (Section 2.2.2). Finally, we discuss currently used metrics to evaluate model performance on IE datasets and introduce some challenges in applying them to measuring the performance on DWIE (Section 2.2.3).

2.2.1 Related datasets

Most of IE datasets have focused on a single task, making it very challenging to develop systems that jointly train for different annotation subtasks on a single corpus. Well-known single-task datasets include (i) *for NER:* CoNLL-2003 [45] and WNUT 2017 [46], (ii) *for relation extraction:* Semeval-2010 T8 [32], TACRED [31] and FewRel [33], (iii) *for entity linking:* IITB [47], CoNLL-YAGO [48], and WikilinksNED [49], and (iv) *for coreference resolution:* CoNLL-2012 [50] and GAP [51]. Conversely, in this work we propose a multi-task dataset as a single corpus annotated with different information extraction layers: named entities, mention clustering in entities (i.e., coreference), relations between entity clusters of mentions, and entity linking. We further complement our dataset with additional tasks such as document classification and keyword extraction. It is worth noting

	Core Tasks				Doc-Based				Entity-Centric				Unaided				
Dataset	**NER**	**Coreference**	**Relations**	**Linking**	**Coreference**	**Relations**	**Multi-label Rel**	**Keywords**	**Classification**	**Multi-label Ent**	**Relations**	**Linking**	**NER**	**Coreference**	**Relations**	**Linking**	**Open**
DWIE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TAC-KBP [22, 24, 26]	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗
BC5CDR [52, 53]	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✓
MUC-7 [54]	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓	✗	✗
SciERC [38]	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓
DocRED [34]	✓	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
Rich ERE [29, 55]	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗
ACE 2005 [28]	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗
OntoNotes 5.0 [56, 57]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
ScienceIE [30]	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	✗	✓
FewRel [33]	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
GENIA [58]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
AIDA CoNLL-YAGO [48]	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
SemEval 2010 T8 [32]	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
NYT [35]	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
ACEtoWiki [59]	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
WNUT 2017 [46]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
CoNLL-2003 [45]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
TACRED [31]	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

Table 2.1: Qualitative comparison of the datasets. We divide our comparison in five groups: (i) *Core Tasks* represent the main subtasks covered in DWIE, (ii) *Doc-Based* indicates whether different subtasks are annotated on the document-level, (iii) *Entity-Centric* indicates which annotations are done with respect to entity clusters (✓) as opposed to individual mentions (✗), (iv) *Unaided* specifies whether the annotation process was completely manual (✓) or with some form of distant supervision (✗), and (v) *Open* indicates whether the dataset is freely available.

that our *coreference* annotations differ from the widely adopted CoNLL-2012 [50] scheme in two aspects: (i) we retain singleton entities composed by only one mention as a valid entity cluster, (ii) we only cluster proper nouns, leaving out nominal and anaphoric expressions.

Furthermore, most prominent efforts to produce jointly annotated datasets have focused on using a *top-down* annotation approach. This method involves an a priori defined annotation schema that drives the process of selection and labeling of the corpus. The de facto datasets used in most of the joint learning baselines such as ACE 2005 [27, 28], TAC-KBPs [22–26] and Rich ERE [29] use this annotation approach. More specifically, during the creation of the ACE 2005 dataset, the annotators initially tagged candidate documents as "good" or "bad" depending on the estimated number and types of entities present in each one. In subsequent annotation stages, only "good" documents were fully annotated and included in the final dataset. Similarly, during the creation of the TAC-KBP datasets, the annotators focused on producing annotations evenly distributed among three

Dataset	# Tokens	Entities			Relations			Linking	
		# Mentions	# Entity clusters	# Entity types	# Relation mentions	# Relation clusters	# Relation types	# Mention KB Links	# Cluster KB Links
NYT	5,765,332	1,388,982	-	-	142,823	-	52	1,388,982	-
TACRED	3,866,863	-	-	-	21,784	-	42	-	-
TAC-KBP[5]	3,053,336	6,495	3,750	-	-	-	-	3,818	2,094
OntoNotes 5.0	2,088,832	161,783	136,037	-	-	-	-	-	-
FewRel[6]	1,397,333	114,213	112,000	-	58,267	56,000	80	114,213	112,000
DocRED	1,018,297	132,392	98,610	6	155,535	50,503	96	-	-
MUC-4	717,798	14,196	-	13	-	-	-	-	-
GENIA	554,346	56,743	10,728	5	2,337	-	2	-	-
DWIE	501,095	43,373	23,130	311	317,204	21,749	65	28,482	13,086
BC5CDR	343,175	29,271	10,326	2	47,813	3,116	1	29,562	10,326
CoNLL-2003	301,418	35,089	-	4	-	-	-	-	-
CoNLL-YAGO	301,418	34,929	-	-	-	-	-	34,929	-
ACE 2005	259,889	54,824	37,622	51	8,419	7,786	18	-	-
ACEtoWiki	259,889	-	-	-	-	-	-	16,310	-
SEval 2010 T8	207,307	21,434	-	-	6,674	-	9	-	-
ACE 2004	185,696	29,949	12,507	43	5,976	5,525	24	-	-
WNUT 2017	101,857	3,890	-	6	-	-	-	-	-
ScienceIE	99,580	9,946	9,536	3	638	-	1	-	-
SciERC	65,334	8,094	1,015	6	2,687	-	7	-	-

Table 2.2: Numerical comparison of DWIE and well-known IE datasets. Note that some datasets (including DWIE) use an entity-centric approach, organizing entity mentions in entity clusters, and annotating entities, relations, and linking on the cluster level. Hence, we provide both mention-level as well as cluster-level (if a particular dataset supports it) statistics.

entity types (PERs, ORGs, and GPEs) by annotating only the documents that contained a minimum number of entities related to event types. In the case of Rich ERE, the documents to tag were prioritized by the event trigger word density calculated per 1,000 tokens, thus focusing only on content with a high number of previously defined key event-related tokens. Furthermore, other IE-related datasets [30–34] use similar pre-filtering techniques in order to select the text to be annotated. As a consequence, the corpus and annotations in these datasets tend to be biased and likely not representative of the language used in the different input domains. Conversely, we adopt a radically different *bottom-up* approach where we derive the annotations (e.g., entity classification types, relation types) from the data itself. This bottom-up data-driven procedure guarantees that the annotations in DWIE are representative of the document corpus information and reflects the particularities of the language used in its journalistic domain. Furthermore, it better represents the properties that are inherently present in written corpora, e.g., the long-tail distribution of different annotation types.

Finally, from the perspective of the necessary evidence to annotate a particular entity type or relation, we propose to make a distinction for the currently existing datasets between *trigger-based* and *document-based* annotations (see *Doc-Based* comparison group in Table 2.1). The *trigger-based*

[5]The EDL track only of TAC-KBP 2010.

[6]Numbers based on publicly available train and development sets.

datasets require that a particular relation or entity type should only be annotated if it is supported by an explicit reference in a text. For example, in Fig. 2.1 there is a concrete reference of the relation between "Meghan" and "Harry" in form of triggers such as "gets engaged" in sentence 1 and "The wedding" in sentence 2. Most of the traditionally used jointly annotated datasets such as ACE 2005 [27, 28], TAC-KBPs [22–26] and Rich ERE [29], as well as others, including FewRel [33], OntoNotes [56, 57], TACRED [31], SemEval 2010 Task 8 [32] and SciERC [38], are *trigger-based*. The disadvantage of such an approach is that it only captures the most simple cases of relations and entity types that are explicitly mentioned in the text. As a general rule, this also limits the datasets to cover only the relations between entity mentions (i.e., the annotation process is mention-driven) that appear within a single or at most few adjacent sentences where the relation trigger occurs (see Fig. 2.2 in Section 2.3 for a more detailed illustration of this phenomenon). However, as we move to a broader *document-based* interpretation, it is common to find relations that are not explicitly mentioned in text. Thus, in our example of Fig. 2.1 the relation between "Ministry of Defense" and "Britain" is not explicitly indicated in the text. However, after reading the whole article we can infer relations such as ministry_of, agency_of and based_in between these two entities. This document-level reasoning makes it essential to adopt an entity-centric approach (see *Entity-Centric* comparison group in Table 2.1) where each *entity* comprises one or more entity *mentions*, and the annotations (i.e., *relations*, *entity tags* and *entity linking* in DWIE) are made on the entity level, thus abstracting from specific *mention-driven* triggers.

2.2.2 Recent advances in information extraction

In the last couple of years, the advances in joint modeling have been accompanied by an ever increasing interest in the use of graph-based neural networks [39–41]. Initially, this approach has been applied to improve the performance of the single coreference resolution task by transferring document-level contextual information between coreferenced entity mention spans [20, 42]. Most recently, these graph propagation techniques have been successfully used in a joint setting [18, 21, 43, 44] by performing graph message passing updates between the shared spans across different tasks. However, while successful on mention-driven datasets such as ACE 2005 [28] and NYT [35], as far as we are aware, the advantages of these techniques have not yet been investigated in an entity-centric document-level setting. We fill this gap by extending the neural graph-based model initially proposed by [21] to be used on DWIE (see Section 2.4). More specifically, we explore the effect of performing document-level coreference (`CorefProp`) [20, 21] and relation-driven (`RelProp`) [21] graph message passing updates between the spans. Additionally, we introduce a new latent attention-based graph propagation method (`AttProp`) and com-

pare it to previously proposed task-driven graph propagation methods (`CorefProp` and `RelProp`).

2.2.3 Metrics and evaluation

Current dominant IE systems consider *mention-level* scoring of NER as well as RE components when reporting on datasets such as CoNLL-2003 [60–64], OntoNotes [61, 65, 66], ACE 2004 [67–69], ACE 2005 [18, 21, 69], TACRED [31, 70, 71], and SelEval 2010-Task 8 [72–74] among others. In contrast, the DWIE dataset is entity-centric where all the annotations are done on the entity cluster level. Consequently, adopting a purely mention-based evaluation approach can lead to a dominance of the score by predictions on entities composed by many mentions as opposed to entities composed by only few ones. Conversely, a purely cluster-level evaluation would be overly strict, requiring correct prediction of relation/entity types as well as an exact match of the predicted entity clusters. To tackle this problem, we propose a new scoring method that combines entity mention-level and cluster-level evaluation, while avoiding the pitfalls of either method alone (see Section 2.5).

2.3 Annotation process

In this work we introduce our *bottom-up* data-driven annotation approach. Our main goal is to get an annotation schema that reflects the types of entities and relations that are effectively mentioned throughout the corpus to maximally capture the information it contains. Therefore, we derive the annotation schema from the corpus itself, adopting three annotation passes that are detailed next: (i) *exploratory pass*, (ii) *schema-driven pass*, and (iii) *inter-annotator refinement*. Each pass encompasses substeps to cover all IE subtasks: (i) mention annotation (i.e., the entities and their types), (ii) coreference resolution, (iii) relation extraction on the entity level (i.e., clustering all mentions referring to the same entity), and (iv) entity linking (again, on the entity level, providing the same link for all clustered mentions).

2.3.1 Exploratory pass

The first annotation pass aims to discover the annotation structure (i.e., annotation schema) to be used on the corpus, in particular the types to use for named entity recognition (NER) and relation extraction (RE) tasks. Three annotators are involved in this step to provide annotations on the mention level: one expert annotator and two paid students. However, no parallel annotation is done and the role of the expert annotator is to annotate part of the corpus, as well as instructing and supervising the paid annotators.

Entity Tag	Description	Example
ENTITY	All nominal named entities.	"UK court rules WikiLeaks' Assange should be extradited to Sweden"
location	Entities referring to a particular geographical location.	"Libya is one of Germany's strongest trading partners in northern Africa."
organization	Organizations such as companies, governmental organizations, etc.	"According to the report, Amazon would pay the same level of royalty fees as Apple."
person	Entities referring to people in general such as politicians, artists, sport players, etc.	"With Ramires out, Drogba could start as striker, with Torres moving to the wing."
misc	Miscellaneous entity types such as names of work of arts, treaties, product names, etc.	"According to the director's own words, The Post is a 'patriotic film'."
event	Events such as sport competitions, summits, etc.	"Last year's Champions League final drew a crowd of just 14,303."
ethnicity	Entity type used to identify different ethnic groups.	"Attempt to assimilate Uyghurs into dominant Han Chinese culture."
VALUE	Values in general such as time, money, etc.	"It ended the 2014 fiscal year 45 million euros ($51 million) in the red."
OTHER	Includes the nominal variations of entity types (e.g., includes variations of country names such as "German", which is a variation of "Germany").	"Franco-German 'war child' granted German citizenship."

Table 2.3: *Descriptions* and *Examples* (with entity mentions underlined) of each of the most granular entity classes in DWIE (*ENTITY*, *VALUE* and *OTHER*) in the *type* tag hierarchy. Additionally, for the type *ENTITY*, we describe and give examples of each of its direct subtypes (*location*, *organization*, *person*, *misc*, *event* and *ethnicity*).

No a priori fixed schema is followed, but we ask the annotators to be as consistent as possible during the process. More specifically, the annotators are free to define their own entity and relation types for the NER and RE tasks that reflect the contents of the articles as long as they comply with the following generic guidelines:

1. **Named Entities:** any physical or abstract object (e.g., "Washington", "Jeff Davis","Nobel Prize", "Lisbon Treaty", etc.) that can be denoted with a proper noun. Entities are usually upper-cased in the text, although values such as money and time can also be included. Use short and specific entity types (e.g., person, organization, etc.) to classify entities, the types can be overlapping (a single entity can have multiple types).

2. **Relations:** identify meaningful relations between entities. The type of a relation should be specific and reflect the type of the connected entities as well as the semantic meaning of the relation. For example, instead of using a generic "located in" relation for entities located in a particular country, we can divide it in "based in country" for organizations that are based in a country, "city located in country" for cities located in the country, etc. The types of the relations should have short names, ideally not exceeding 15 characters.

By not constraining the annotation process to specific entity and relation types, we ensure that our annotations are representative of the actual information contained in the annotated corpus.

2.3.2 Schema-driven pass

The main goal of this step is to create a consistent annotation schema for (i) named entity types and (ii) relation types based on the annotations made in the *exploratory pass*. As a first step, we identify the classification *tags* to be assigned to **entities**. We divide these tags in five main categories: *type*, *topic*, *iptc*, *slot*, and *gender* (see Table 2.14). Our *type* tag is organized in a hierarchical structure (see Table 2.11 in 2.7), making it easier to extend our annotations to more granular subtypes. Table 2.3 defines and provides examples of each of the top *type* tags in the entity type hierarchy (*ENTITY*, *VALUE* and *OTHER*) as well as the direct subtypes of *ENTITY*. The *topic* tag allows to assign topics (e.g., politics, culture, education, etc.) to the entities and it complements the *type* tag (see Table 2.12). The *iptc* tag is used for the universally defined IPTC news categories based on a media taxonomy (https://iptc.org/standards/subject-codes/). The *slot* tag is used for additional categorization that is transversal to different entity types. One example of this is the slot *interviewee* that can be assigned to any person (entity of type *person*) interviewed in a particular article.[7] Finally, the *gender* tag is used

[7] Other possible slot values are: *keyword*, *head*, *death*, *interviewer* and *expert*.

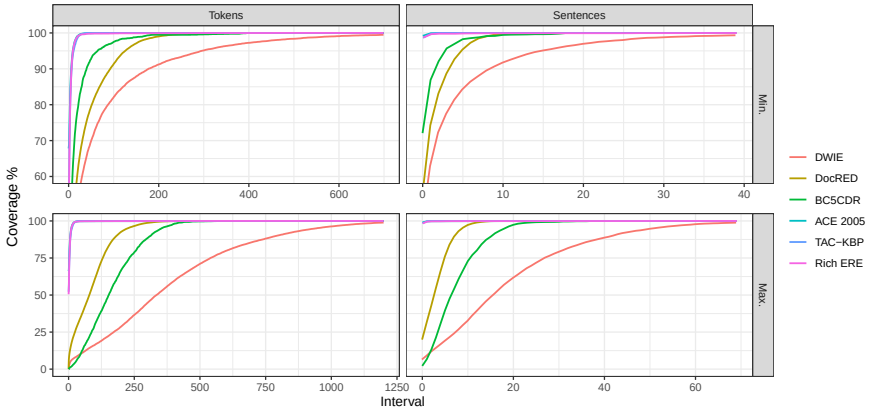


Figure 2.2: Comparison of the coverage of the % of relations with increasing interval in tokens (left) and interval in sentences (right). The graph at the top illustrates the relations coverage measuring the minimum distance between entities (closest mentions). Conversely, the graph at the bottom shows the coverage measuring the maximum distance between entity mentions. In both graphs, we note that the distance between the related mentions in our dataset is higher than in other widely used datasets.

to indicate the gender of the entities that refer to people. By defining these multiple overlapping tag types, we realize that the entity classification is multi-label by nature and thus allows different complementary entity tags to be assigned to a particular entity.[8] This contrasts with prevailing single-label multi-class datasets such as ACE 2005 [27, 28], TAC-KBPs [23–26], Rich ERE [29], WNUT 2017 [46] and CoNLL-2003 [45].

For our **relation** annotations, we focus on annotating relations between entities themselves (cf. *document-based entity-centric* approach). Our adopted approach allows us to think concept-wise and come up not only with relations that are explicitly stated, but also those that can be implicitly inferred from the text. As a result, our dataset includes relations whose connected mentions are located further apart in the document. This can be seen in Fig. 2.2, where we compare the *minimum* (*Min.*) and *maximum* (*Max.*) distances between the mentions of the two entities connected by a relation for various mention-driven (Rich ERE[9], TAC-KBP[10], and ACE 2005) and entity-centric (DocRED, BC5CDR, and the final version of our DWIE dataset) RE datasets. We note how other datasets that define the relation in terms of entities (BC5CDR and DocRED) require a higher number of token and sentence spans to cover all the relations in the respective

[8] The average number of labels per entity is 4.0 in our DWIE dataset.
[9] We use the Rich ERE dataset from the LDC2015E29 and LDC2015E68 catalogs.
[10] We use the TAK-KBP 2017 dataset from the LDC2017E54 and LDC2017E55 catalogs.

Relation Type	Description	Example
based_in0	Relations between organizations and the countries they are based in, ex: based_in0⟨*Uni of Cologne*, *Germany*⟩	"Now he's back in Germany carrying on with his cancer research at the University of Cologne."
in0	Relations between geographic locations and the countries they are located in, ex: in0⟨*Athens*, *Greece*⟩	"The murder of a left-wing activist in Athens has shaken up Greece and inspired a backlash."
citizen_of	Relations between people and the country they are citizens of, ex: citizen_of⟨*Guerrero*, *Peru*⟩	"Even as a teenager, Guerrero played for the national side in his native Peru."
based_in0-x	Relations between organizations and the nominal variations of the countries they are based in, ex: based_in0-x⟨*SPD*, *German*⟩	"SPD denies 'green light' for new German government, but keeps options open"
citizen_of-x	Relations between people and the nominal variations of the countries they are citizens of, ex: citizen_of-x⟨*Assange*, *Australian*⟩	"Australian national Assange said the accusations were politically motivated."

Table 2.4: *Descriptions* and *Examples* of the top 5 most occurring relation types in DWIE. The entity mentions involved in the relations are underlined.

dataset: entity-centric relations very often involve mentions located in different sentences in the document that refer to those entities. This is not the case for mention-driven trigger-based relations as in the TAC-KBP, Rich ERE and ACE 2005 datasets, where the annotation bias is towards finding explicitly mentioned relations, often involving entity mentions in a single sentence.

Similarly as with entity tags, we organize our relation annotations using multi-label types (see Table 2.15 for details). Table 2.4 gives some examples from the DWIE corpus for the top 5 most occurring relation types (a detailed list can be consulted in Table 2.16). For reasons of space, the examples only involve relations between entities whose mentions occur in a single sentence; for an example involving document-level relations we refer to Fig. 2.1.

Task	Before Refinement	After Refinement
Named Entity	0.8497	0.8703
Named Entity Detection	0.9665	0.9673
Named Entity Classification	0.8812	0.9026
Coreference	0.9302	0.9324
Entity Linking	0.9280	0.9320
Relation	0.6594	0.8729
Relation Detection	0.7686	0.8727
Relation Classification	0.8118	0.9666

Table 2.5: The inter-annotation agreement Cohen's kappa scores for all the different annotation tasks *before* and *after* the dataset refinement used to analyze and correct the discrepancies between the parallel annotations.

Additionally, we define logical rules to automatically guarantee the consistency of the relations and their types. The following is an example,

$$\mathsf{based\_in2}\langle X, Z\rangle \wedge \mathsf{in0}\langle Z, Y\rangle \implies \mathsf{based\_in0}\langle X, Y\rangle \tag{2.1}$$

reflecting the knowledge that if an organization X is based in a city Z (relation based_in2), and that this city Z is located in the country Y (relation in0), the fact that company X is also located in that country (relation based_in0) is valid as well. The goal of this step is mainly consistency of the annotations, but it implies that an effective predictor would need to perform some form of reasoning to correctly predict all relations in the dataset. A complete list of logical rules is provided in 2.7.

2.3.3 Inter-annotator refinement

In order to assess and further improve the quality of our dataset we re-annotate a 100 randomly selected news articles (12.5% of the articles used in the previous annotation rounds) from scratch. This work is done by a second independent expert annotator. The annotations in this pass are performed by following the already defined annotation schema based on the annotation process in the *exploratory* and *schema-driven* passes. We use this second annotated subset to measure the inter-annotator agreement and subsequently determine the parts of the dataset that still need to be improved. Table 2.5 compares the kappa scores before and after this refinement pass for each of the tasks (see 2.7 for details on how the kappa score is calculated). We observe that, after the refinement, all of the kappa scores are above 0.85, which is considered a 'strong' [75] to 'almost perfect' [76] agreement.

Note that the revisions were seeded by and evaluated on the subset of 100 re-annotated articles. However, we argue that the inter-annotator re-

finement improved the annotation consistency of the entire dataset, given that the reviewed entity and relation types are used in more than 99.4% of all annotations in DWIE.

2.4 Model architecture

In this section we introduce the end-to-end architecture used to compare the performance of models trained on the separate tasks with the models that are trained jointly for multiple tasks on the DWIE dataset. The main component of our approach is the use of Graph Neural Networks [39–41, 77], relying on propagation techniques in both single-task and joint setups. More specifically, we implement span-based graph message passing on coreference (`CorefProp`) [20, 21] and relation levels (`RelProp`) [21]. Additionally, we introduce a latent attentive propagation method (`AttProp`) which is not driven by annotations of any task in particular and, as a result, can be freely applied to any task or joint combination of tasks. The interconnection between the different components of our model architecture is depicted in Fig. 2.3. It is based on the *span-based architecture* introduced in [19], which supports training on the space of all entity spans simultaneously, dynamically updating span representations by using the graph propagation approach (further detailed in Section 2.4.4). Recent works have shown that this idea has the potential for improved effectiveness (albeit at a higher computational cost) [18, 20, 21, 78], compared to more traditional sequence-labeling approaches [60, 79–81]. More concretely, the use of a span-based approach where all the spans are shared between the individual task modules avoids the cascading of errors from the entity mention identification module (*entity scorer* in Fig. 2.3) to the rest of the tasks.

The most similar architecture to ours in using joint span-based neural graph IE is DyGIE [21] and its successor DyGIE++ [43]. Our model is described in detail below, but here we already list the aspects in which it differs from these models:

1. We introduce the graph propagation technique `AttProp` (see Section 2.4.4), which is not directly conditioned on a particular task and can be used in single-task (for each of the tasks) as well as joint settings.

2. We define a coreference architecture that, unlike previous work in span-based coreference resolution [19, 20], allows to also account for singleton entities in the DWIE dataset (see Sections 2.4.2.2 and 2.4.5.2) by using an additional *pruner loss*, which turns out essential for the single model focusing on end-to-end coreference resolution.

3. Due to the document-level nature of DWIE, we run graph propagations on the whole document. This contrasts with a sentence-based

approach adopted initially in the DyGIE/DyGIE++ architectures. It also drives some changes such as the use of a single *pruner* (see Section 2.4.1) to extract spans used in coreference and RE modules. Similarly, instead of applying the shared BiLSTM sentence by sentence as in [21] and [43], we do it on the entire document, in order to allow capturing cross-sentence dependencies for document-level relations and entity clusters in DWIE.

4. We add an additional decoding step (see Section 2.4.3) needed to transform mention-based predictions for RE and NER tasks into entity-based ones, as required by the entity-centric nature of DWIE, and propose corresponding evaluation metrics (see Section 2.5).

5. Finally, we make changes in the loss and prediction components to support multi-label classification (in NER and RE) as required in DWIE.

2.4.1 Span-based representation

The input to our model consists of document-level annotation instances. Each document D from the considered document collection $\mathcal{D}$ is represented by its sequence T of tokens. These tokens are represented internally as a concatenation of GloVe [82] and character embeddings [79]. We also experiment with additionally concatenating BERT [83] contextualized embeddings. Since BERT is run on a sub-token level, to the representation of each token we only concatenate the BERT-based representation of the first sub-token, as originally proposed by [83]. This input is fed into a BiLSTM layer in order to obtain the output token representations by concatenating the forward and backward LSTM hidden states. The BiLSTM outputs for the considered document D are written on the token level as $\mathbf{e}_i \in \mathbb{R}^m$ $(i = 1, \ldots, |T|)$. These are converted into *span representations*. The set of all possible spans for D, up to maximum span width $w_{\max}$ (which is a hyperparameter of the model), is written as $S = \{s_1, \ldots, s_{|S|}\}$. The number of spans can be calculated as follows,

$$|S| = \sum_{k=1}^{w_{\max}} |T| - k + 1 \; = w_{\max} \left(|T| - \frac{w_{\max} - 1}{2} \right) \tag{2.2}$$

We obtain the representation $\mathbf{g}_i^0$ for span s_i, ranging from token l to token r, by concatenating their respective BiLSTM states $\mathbf{e}_l$ and $\mathbf{e}_r$ with an embedding $\boldsymbol{\psi}_{r-l}$ for the span width $w_i = r - l$

$$\mathbf{g}_i^0 = [\mathbf{e}_l; \mathbf{e}_r; \boldsymbol{\psi}_{r-l}] \tag{2.3}$$

As seen from Eq. (2.2), the number of possible spans scales approximately linearly with the maximum span width $w_{\max}$, as well as the doc-

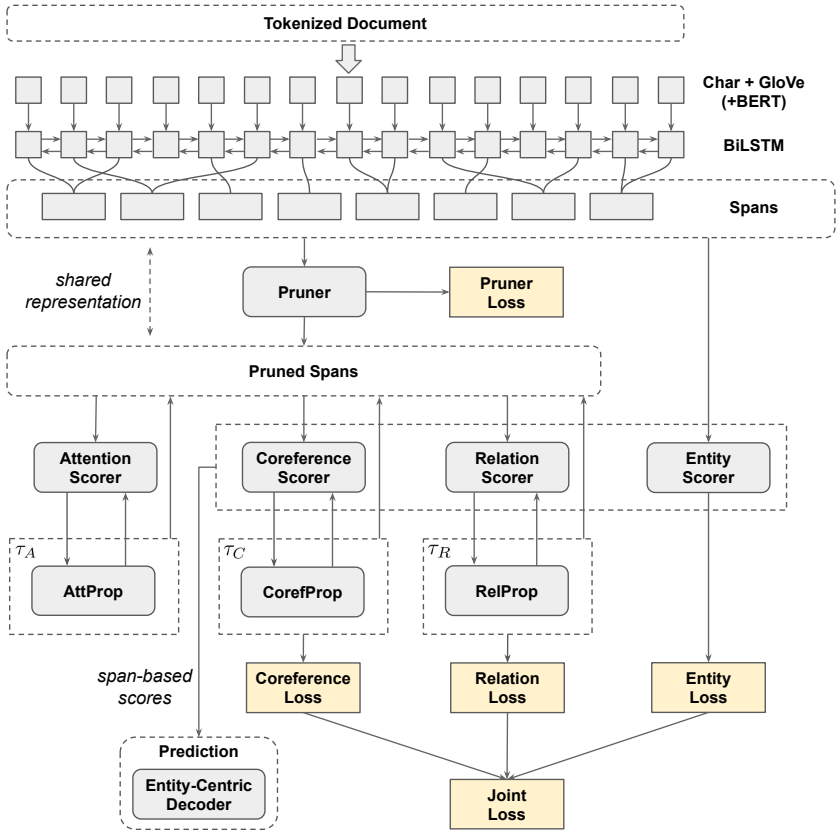


Figure 2.3: Architecture of our model; the span-oriented approach makes it possible to execute *coreference* (Section 2.4.2.2) and *relation* (Section 2.4.2.3) scorers independently from *entity scorer* (Section 2.4.2.1). However, a pruning step (described in Section 2.4.1) is needed in order to limit the memory required to perform matrix operations on span representations involved in graph propagation (`AttProp`, `CorefProp`, `RelProp`)(Section 2.4.4) as well as in the attention, coreference and relation scorer modules. The *pruned spans* share the same representation with the rest of the *spans* (*shared representation*). This way, the update in span representations caused by the graph propagation modules also affects the *entity scorer*. Our `AttProp` graph propagation method runs independently from coreference, relation, and entity scorers, enabling its use in combination with any task. Finally, the *entity-centric decoder* (Section 2.4.3) uses the entity clusters predicted by the *coreference scorer* to convert the span-based predictions from the *relation* and *entity* scorers to entity-centric ones.

ument length $|T|$ (assuming $w_{\max} \ll |T|$). This leads to a strongly in-

creased set of spans, as compared to previous works where $|S|$ scales with the length of individual sentences rather than entire documents [21, 43]. In order to mitigate the required memory of our model, we use a shared *pruner* to reduce S to a smaller set P of candidate spans to be used by the coreference and RE scorers and in the graph propagation modules (see further). The choice of using a single pruner contrasts with similar work in [21] and [43] where two separate pruners are used, one for the relation task, and another for coreference. Our design choice is based on the fact that both of these tasks use the same document-level entity mentions. This contrasts with datasets used in [21] and [43] where, while the coreference is defined on the document-level, the relations are sentence-based.

Finally, we use graph propagation to iteratively refine the pruned spans representations. Three graph propagation mechanisms are compared in the experiments. Our own contribution is the attention-based graph propagation method `AttProp`, where the span representations are updated in τ_A iterations. Alternatively, τ_C iterations of `CorefProp` [20, 21] can be performed, or τ_R iterations of `RelProp` [21].

The span representation of a particular span s_i after iteration t is denoted as $\mathbf{g}_i^t$ in our notation. The details of graph propagation are explained in Section 2.4.4. Note that in theory several of these graph propagation techniques could be accumulated, but in our setting the benefits thereof in terms of model effectiveness were minor, at a significantly higher computational cost. Therefore, in our experiments, we only compare models without graph propagation with models applying a single form of graph propagation. To keep the sections introducing the models clear, we will write τ to denote the number of propagations in general (which could be 0, or any of τ_A, τ_C or τ_R, depending on the chosen experiments and considered model components).

2.4.2 Joint model for entity recognition, coreference resolution, and relation extraction

In this section, we present the joint model including recognition of entity mentions as belonging to L_T types (introduced as NER), the clustering of the entity mentions into entities (coreference resolution), and identifying relations between entities, all on the document level. The building blocks responsible for the three subtasks are discussed next, as well as the total loss of the joint model. The details of the graph propagation mechanisms are then provided further on (Section 2.4.4).

2.4.2.1 Entity mention module

All spans s_i (up to width w_{max}) of the considered document[11] are scored by feeding their representation (starting from Eq. (2.3) and potentially updated after τ graph propagation iterations) into the feed-forward neural network (FFNN) written as $\boldsymbol{\mathcal{F}}_{\text{mention}}$, with as many outputs as there are entity types:

$$\boldsymbol{\Phi}^{\tau}_{\text{mention}}(s_i) = \boldsymbol{\mathcal{F}}_{\text{mention}}(\mathbf{g}_i^{\tau}). \tag{2.4}$$

Throughout this section, we will maintain the same notation of $\boldsymbol{\mathcal{F}}(\mathbf{x})$ to denote a FFNN that takes as input a vector $\mathbf{x}$ and produces a vector of scores, and $\mathcal{F}(\mathbf{x})$ to refer to a FFNN with a scalar output.

The probability of each label being valid for the considered span is modeled by component-wise application of a sigmoid ($\sigma(x) = 1/(1 + e^{-x})$) to these scores $\boldsymbol{\Phi}^{\tau}_{\text{mention}}(s_i) \in \mathbb{R}^{L_T}$ (with L_T the number of entity tags). The log probability of the ground truth mention labels for all spans of document D is given by

$$\begin{aligned}\log P_{\text{mention}}\left(E^*|G^{\tau}\right) = \sum_{i=1}^{|S|}\sum_{l=1}^{L_T} I_{i,l}\log\sigma(\boldsymbol{\Phi}^{\tau}_{\text{mention}}(s_i)_l) + (1 - I_{i,l}) \\ \log\left(1 - \sigma\left(\boldsymbol{\Phi}^{\tau}_{\text{mention}}(s_i)_l\right)\right),\end{aligned} \tag{2.5}$$

in which E^* represents the set of ground truth mention labels for all spans in the document, and $I_{i,l} \in \{0, 1\}$ is the ground truth indicator label for mention tag l of span s_i. G^{τ} denotes the set of all considered span representations for the current document. The superscript τ reflects the fact that, in case graph propagation is applied, the subset of $|P|$ representations (for the spans retained after pruning) have been updated over τ iterations. By summing over all entity types ($l = 1, \ldots, L_T$), we account for the fact that a particular span can have multiple associated entity tags (i.e., the considered NER task is multi-label). At inference time, spans get assigned those entity types for which the corresponding score $\boldsymbol{\Phi}^{\tau}_{\text{mention}}(s_i) > 0$. Note that not all valid entity mentions necessarily get an entity type assigned: if the relation extractor determines that a span is part of a relation, it effectively becomes an entity mention, even if none of the pre-defined types is considered applicable by the entity scorer.

2.4.2.2 Coreference module

While the entity scoring is performed on all span representations S, this is not possible for the coreference and relation scorers, due to memory limitations. The latter scorers predict on *pruned spans*, as shown in Fig. 2.3. How

[11] For convenience, the subscript D indicating the current document is left out in the equations of this section.

the pruner is trained jointly with the model, is described in Section 2.4.2.4. In order to avoid confusion by introducing additional notations, we list the spans in the pruned set P as $s_1, \ldots, s_{|P|}$, according to their original order in the text.

The module for coreference resolution is based on pairwise scoring of the pruned spans from P. Following ideas from [19–21, 38], for any span s_j, scores with respect to each of the preceding (also referred to as 'antecedent') spans s_i $(i \leq j)$ in the document are calculated with a neural network $\mathcal{F}_{\text{coref}}$:

$$\Phi^{\tau}_{\text{coref}}(s_i, s_j) = \mathcal{F}_{\text{coref}}\left([\mathbf{g}_i^{\tau}; \mathbf{g}_j^{\tau}; \mathbf{g}_i^{\tau} \odot \mathbf{g}_j^{\tau}; \boldsymbol{\varphi}_{i,j}]\right). \tag{2.6}$$

This expression scores the compatibility between spans s_i and s_j, taking as input the concatenation of their respective span representations (after τ propagation iterations), their component-wise product, and an embedding $\boldsymbol{\varphi}_{i,j}$ representing their distance in terms of the number of ordered candidate spans from s_i to s_j.

In order to deal with non-coreferent or incorrect spans, previous work in span-based coreference [19, 20] defines a dummy antecedent ϵ to which all non-coreferent or invalid spans point. While this approach is effective in datasets that do not contain singleton entity clusters, such as OntoNotes-based CoNLL-2012 [50], it does not allow to distinguish between valid singleton entity mentions and invalid mention spans. This makes it unsuitable to use on DWIE, since it contains singleton entity clusters, consisting of a single mention. In fact, 66.4% of the entity clusters in DWIE are singletons. Furthermore, the current official CoNLL-2012 evaluation script[12] based on [84] accounts for scenarios where either the dataset or the predicted mentions are singletons, which has a direct impact on the established B-CUBED [85] and CEAF_e [86] coreference scores. In order to tackle the singleton entity cluster detection in our coreference model, we propose to start from $\Phi^{\tau}_{\text{coref}}(s_j, s_j)$[13] as a self-coreference span score. By applying the correct target in the coreference loss, it allows indicating that either the span s_j is not a valid mention, or that it is a valid mention that is not co-referenced with any antecedent span.

The log probability of the ground truth coreference labels of document D is given by

$$\log P_{\text{coref}}\left(C^* | G^{\tau}\right) = \sum_{j=1}^{|P|} \log \frac{\sum\limits_{s^* \in S_j^*} \exp\left(\Phi^{\tau}_{\text{coref}}(s^*, s_j)\right)}{\sum\limits_{i=1}^{j} \exp\left(\Phi^{\tau}_{\text{coref}}(s_i, s_j)\right)}. \tag{2.7}$$

[12] https://github.com/conll/reference-coreference-scorers

[13] This would be replaced with $\Phi^{\tau}_{\text{coref}}(\epsilon, s_j)$ in the *dummy-based* formulation defined in [19].

The set of ground truth coreference labels is indicated as C^*. The summation over j represents the contribution to the log likelihood of the correct antecedent labels for each span s_j in the pruned set P. The individual terms in the right-hand side correspond to the log probability of the correct antecedent labels for a particular span s_j. In the denominator, the summation ranges from the first span, up to span s_j itself (i.e., for the self-coreference score), but not beyond it (given that only *antecedents* in the sorted sequence of pruned spans are considered). The numerator contains the contributions from the potentially multiple ground truth antecedents for span s_j. This stems from the fact that multiple antecedent mentions may belong to the same cluster as s_j, which all contribute to the probability of the correct antecedent labels. The set of ground truth antecedents corresponding to span s_j is written S_j^*.

At inference time, the highest scoring antecedent for span s_j (including s_j itself) is picked. Due to the idea of only predicting *antecedents*, picking any of the ground truth antecedents leads to the correct mention clusters [19, 20, 87, 88].

2.4.2.3 Relation module

Similar to the coreference module (Eq. (2.6)), we score span pairs using an FFNN

$$\mathbf{\Phi}^{\tau}_{\text{relation}}(s_i, s_j) = \boldsymbol{\mathcal{F}}_{\text{relation}}\left([\mathbf{g}_i^{\tau}; \mathbf{g}_j^{\tau}; \mathbf{g}_i^{\tau} \odot \mathbf{g}_j^{\tau}; \boldsymbol{\varphi}_{i,j}]\right), \tag{2.8}$$

where $\boldsymbol{\varphi}_{i,j}$ is again the distance embedding as introduced in Section 2.4.2.2. $\mathbf{\Phi}^{\tau}_{\text{relation}}(i,j) \in \mathbb{R}^{L_R}$ is a vector representing relation span pair scores for each of the L_R possible relation types between spans s_i and s_j.

The log probability of the ground truth relation labels of document D is given by

$$\begin{aligned}\log P_{\text{relation}}\left(R^*|G^{\tau}\right) = \sum_{i,j=1}^{|P|}\sum_{l=1}^{L_R} I_{i,j,l} \log \sigma(\mathbf{\Phi}^{\tau}_{\text{relation}}(s_i, s_j)_l) + \\ (1 - I_{i,j,l}) \log\left(1 - \sigma(\mathbf{\Phi}^{\tau}_{\text{relation}}(s_i, s_j)_l)\right),\end{aligned} \tag{2.9}$$

in which R^* represents the set of ground truth relation labels for all combination of pruned span pairs in the document, and $I_{i,j,l} \in \{0, 1\}$ is the ground truth indicator label for relation type l of the span pair (s_i, s_j). Note that all $|P|^2$ pruned span pairs are considered, since the order of the spans in the relation matters (unlike the coreference case). By summing over all possible relation types L^R, we account for the fact that a particular relation between two spans can be multi-label (which is the case for more than 30% of relations, as shown in Table 2.15).

Since this model is run in parallel with the coreference module, it is used to predict relations only between entity mentions and not entity clusters. During inference, candidate relations are accepted when $\mathbf{\Phi}^{\tau}_{\text{relation}}(s_i, s_j)_l > 0$.

2.4.2.4 Span pruner

The span pruner is an FFNN, denoted $\mathcal{F}_{\text{pruner}}$, that scores all spans s_i based on their initial representation $\mathbf{g}_i^0$, after which only the highest scoring spans are retained in the pruned span set P. In our experiments P contains the top $0.2\,|T|$ highest scoring spans, which covers more than 98% of all the ground truth mention spans in the DWIE dataset. We represent the pruner score for span s_i as

$$\Phi_{\text{pruner}}\left(s_i\right) = \mathcal{F}_{\text{pruner}}\left(\mathbf{g}_i^0\right). \tag{2.10}$$

Several strategies can be used to train the pruner. One option is to directly optimize the probability of the pruner to detect the spans of correct entity mentions. With S^* the set of spans with at least one ground truth entity type, and $I_i \in \{0, 1\}$ an indicator for whether $s_i \in S^*$, the corresponding log likelihood can be written as

$$\begin{aligned}\log P_{\text{pruner}}\left(S^* | G^0\right) = \sum_{i=1}^{|S|} I_i \log \sigma\left(\Phi_{\text{pruner}}(s_i)\right) + \\ (1 - I_i) \log\left(1 - \sigma\left(\Phi_{\text{pruner}}(s_i)\right)\right),\end{aligned} \tag{2.11}$$

leading to a separate pruner loss term. Alternatively, the pruner can be trained indirectly by adapting the mention score from Eq. (2.4), the coreference score from Eq. (2.6) or the relation score from Eq. (2.8) as follows:

$$\tilde{\Phi}^{\tau}_{\text{mention}}\left(s_i\right) = \Phi^{\tau}_{\text{mention}}\left(s_i\right) + \Phi_{\text{pruner}}\left(s_i\right) \tag{2.12}$$

$$\tilde{\Phi}^{\tau}_{\text{coref}}\left(s_i, s_j\right) = \Phi^{\tau}_{\text{coref}}\left(s_i, s_j\right) + \Phi_{\text{pruner}}\left(s_i\right) \tag{2.13}$$

$$\tilde{\Phi}^{\tau}_{\text{relation}}\left(s_i, s_j\right) = \Phi^{\tau}_{\text{relation}}\left(s_i, s_j\right) + \Phi_{\text{pruner}}\left(s_i\right) \tag{2.14}$$

for use in the expressions Eq. (2.5), Eq. (2.7) and Eq. (2.9), respectively. As such, higher pruner scores would directly correspond to higher mention or coreference scores, and lead to a meaningful ranking of spans according to pruner scores. All three strategies seem to work on a similar level, but for the presented joint model experiments, we use the indirect training through the coreference module, as in Eq. (2.13). Note that we did not experiment with training the pruner through the relation module, because it would be trained only on those spans involved in relations, which is a mere subset of all valid mentions.

2.4.2.5 Joint model

We perform joint training in order to explore the degree to which the graph propagation techniques (see Section 2.4.4) affect related tasks in DWIE. For instance, we expect that performing a coreference propagation can have a positive impact on the NER task. We hypothesize that enriching the entity spans with broader contextual information coming from other mention spans in the cluster, can improve the effectiveness of the entity module. Furthermore, given the entity-centric nature of DWIE, the mention-based predictions for NER and RE have to be grouped in coreference clusters (see section 2.4.3 for details), which makes it necessary to execute these tasks jointly with the coreference task.

The joint loss for each document D is a weighted sum of the individual loss functions of the subtasks:

$$\mathcal{L}_D^{\text{joint}} = \sum_{(E^*,C^*,R^*)} \lambda_E \log P_{\text{mention}}\left(E^*|G^\tau\right) + \lambda_C \log P_{\text{coref}}\left(C^*|G^\tau\right) + \lambda_R \log P_{\text{relation}}\left(R^*|G^\tau\right), \tag{2.15}$$

in which λ_E, λ_C, and λ_R are hyperparameters of the joint model.

2.4.3 Decoding and prediction

Unlike previous datasets used in span-based predictions [27, 28, 38, 89] where the relation and entity extraction are done on the mention-level, DWIE is an entity-centric dataset. During inference, this requires an additional decoding step to cluster the mention-based span-dependent predictions into entity-centric ones. The component responsible for this decoding in the proposed architecture is the *entity-centric decoder* (see Fig. 2.3). The pseudo-code in Algorithm 1 summarizes the steps performed by this component. First, the decoder receives as *input* the predicted span clusters (*p_cl*), entity mentions (*p_men*) and relations between spans (*p_rel*) obtained from the scores calculated in Eq. (2.13), Eq. (2.4) and Eq. (2.8), respectively. Next, the predicted entity mentions are connected with the respective clusters by using the dictionary C that maps mention spans to cluster ids (lines 3–12 in Algorithm 1). Specifically, each of the entity clusters is assigned the union of the entity types predicted for any of the mention spans inside the cluster (line 11 in Algorithm 1). If the predicted entity mention can not be located inside the predicted clusters, a new singleton cluster is added (lines 5–6 in Algorithm 1). Finally, all the pairwise predicted relations on the mention level (*p_rel*) between members of two different clusters are assigned as predicted relations between the (cluster-level) entities (lines 13–20 in Algorithm 1). Similarly as with entity mentions, the dictionary C is used to map the mention spans (*span_h* and *span_t*) of a particular relation type *rel_type* to the corresponding cluster ids. Furthermore,

the relations added between two clusters are the union of all the relations predicted between any pair of mentions inside these clusters (line 18 in Algorithm 1).

2.4.4 Graph propagation mechanisms

In order to evaluate the impact of graph-based propagation of contextual information between the spans, we propose `AttProp`, and reimplement the `CorefProp` and `RelProp` graph propagation algorithms. [20] proposed the gated graph propagation update function for use on coreference resolution, which was then successfully applied in a joint multi-task setting by [21, 43]. The graph propagation equations are written as:

$$\mathbf{f}_x^t(s_i) = \sigma\left(\boldsymbol{\mathcal{F}}_x([\mathbf{g}_i^t; \mathbf{u}_x^t(s_i)])\right), \tag{2.16}$$

$$\mathbf{g}_i^{t+1} = \mathbf{f}_x^t(s_i) \odot \mathbf{g}_i^t + \left(1 - \mathbf{f}_x^t(s_i)\right) \odot \mathbf{u}_x^t(s_i), \tag{2.17}$$

where in our case $x \in \{A, C, R\}$ denotes `AttProp`, `CorefProp`, and `RelProp`, respectively. The n-dimentional vector $\mathbf{f}_x^t(s_i)$, produced by the single-layer FFNN $\boldsymbol{\mathcal{F}}_x$ can be interpreted as a gating vector that acts as a switch between the current span representations $\mathbf{g}_i^t \in \mathbb{R}^n$, and the update span vector $\mathbf{u}_x^t(s_i) \in \mathbb{R}^n$. The various graph propagation methods differ in how $\mathbf{u}_x^t(s_i)$ is calculated.

`CorefProp` — The coreference confidence score between span s_i and s_j for propagation iteration t is denoted as $P_C^t(s_i, s_j)$ and calculated as follows,

$$P_C^t(s_i, s_j) = \frac{\exp\left(\tilde{\Phi}_{\text{coref}}^t(s_i, s_j)\right)}{\sum_{i'=1}^{j} \exp\left(\tilde{\Phi}_{\text{coref}}^t(s_{i'}, s_j)\right)}, \tag{2.18}$$

in which $i' \in \{1, \ldots, j\}$ refers to all antecedent spans $s_{i'}$ to span s_j in the pruned span set. Note that the coreference scores according to Eq. (2.13) are used. This means the confidence scores not only reflect whether the considered spans are compatible, but also whether the individual spans are likely to be retained by the pruner as potential entity mentions. In order to perform a `CorefProp` graph iteration, the span update vector $\mathbf{u}_C^t(i) \in \mathbb{R}^n$ is first calculated as a weighted average of the current representation of span s_j and all of its antecedents

$$\mathbf{u}_C^t(s_j) = \sum_{i=1}^{j} P_C^t(s_i, s_j)\, \mathbf{g}_i^t, \tag{2.19}$$

in which the weighting coefficients quantify the coreference compatibil-

Input: predicted clusters (*p_cl*), entity mentions (*p_men*) and relations between mentions (*p_rel*):

1. *p_cl* is a dictionary (map) that maps cluster ids to mention spans
2. *p_men* is list of tuples ⟨predicted span, predicted tag⟩
3. *p_rel* is list of tuples ⟨predicted head span, predicted relation, predicted tail span⟩

Output: clusters (*p_cl*), decoded entities (*d_ent*) and relations between entities (*d_rel*)

```
1:  Initialize d_ent, d_rel ← empty dictionary (map)
2:  C ← transformed p_cl that maps spans to cluster ids
    ▷ Decode entity mentions (p_men) to entities (d_ent) (lines 3–12)
3:  for span, tag in p_men do
4:      if span not in C.keys() then
5:          C[span] ← new concept id
6:          p_cl[C[span]] ← list([span])
7:      end if
8:      if C[span] not in d_ent.keys() then
9:          d_ent[C[span]] ← empty set
10:     end if
11:     d_ent[C[span]].add(tag)
12: end for
    ▷ Decode relations between mentions (p_rel) to relations between entities (d_rel) (lines 13–20)
13: for span_h, rel_type, span_t in p_rel do
14:     if (span_h in C.keys()) and (span_t in C.keys()) then
15:         if ⟨C[span_h], C[span_t]⟩ not in d_rel.keys() then
16:             d_rel[⟨C[span_h], C[span_t]⟩] ← empty set
17:         end if
18:         d_rel[⟨C[span_h], C[span_t]⟩].add(rel_type)
19:     end if
20: end for
```

Algorithm 1: Entity-centric decoder for the *Joint* model.

ity of the corresponding span with s_j. After that, the update equations Eq. (2.16) and Eq. (2.17) are applied.

`RelProp` — Similarly as with `CorefProp`, a relation span update vector is calculated as formalized next,

$$\mathbf{u}_R^t(s_j) = \sum_{i=1}^{|P|} \left(\mathbf{A}_R \, f\left(\mathbf{\Phi}^t_{\text{relation}}(s_i, s_j)\right)\right) \odot \mathbf{g}_i^t, \tag{2.20}$$

where $\mathbf{A}_R \in \mathbb{R}^{n \times L_R}$ is a trainable projection tensor, and f is a non-linear activation function (ReLU). Similarly as in Eq. (2.19), the update vector can be interpreted as a weighted sum of all span representations, with the additional expressiveness stemming from the projection matrix $\mathbf{A}_R$ in accounting for the relation scores.

`AttProp` — In order to measure the impact of the 'supervised' `CorefProp` and `RelProp` propagation techniques described by equations (2.18)-(2.20) above, we introduce a latent attentive propagation. Unlike `CorefProp` and `RelProp` that are driven by the task-specific confidence propagation scores $\Phi^t_{\text{coref}}(s_i, s_j)$ and $\mathbf{\Phi}^t_{\text{relation}}(s_i, s_j)$, `AttProp` is influenced only by latent attention weights between all the pruned spans P calculated as follows,

$$\Phi^t_{\text{att}}(s_i, s_j) = \mathcal{F}_{\text{att}}\left([\mathbf{g}_i^t; \mathbf{g}_j^t; \mathbf{g}_i^t \odot \mathbf{g}_j^t; \boldsymbol{\varphi}_{i,j}]\right), \tag{2.21}$$

where $\boldsymbol{\varphi}_{i,j}$ is the distance feature embedding function between spans s_i and s_j, and $\Phi^t_{\text{att}}(s_i, s_j)$ is the attention score between these spans. This score is normalized with a softmax to get the $P^t_A(s_i, s_j)$ confidence score

$$P^t_A(s_i, s_j) = \frac{\exp\left(\Phi^t_{\text{att}}(s_i, s_j)\right)}{\sum_{j'=1}^{|P|} \exp\left(\Phi^t_{\text{att}}(s_i, s_{j'})\right)}. \tag{2.22}$$

The span update vector $\mathbf{u}^t_A(s_i) \in \mathbb{R}^n$ is calculated as a weighted sum of all the P span representations as opposed to only antecedents in `CorefProp`

$$\mathbf{u}^t_A(s_i) = \sum_{j=1}^{|P|} P^t_A(s_i, s_j)\, \mathbf{g}_j^t. \tag{2.23}$$

2.4.5 Single task models

In this section we shortly describe independent baseline models for the three individual core tasks under study in this paper, as training these models not entirely corresponds to merely minimizing the corresponding loss term from the total loss Eq. (2.15).

2.4.5.1 Single entity recognition model

The single-task NER model is designed for detecting and correctly labeling the individual entity spans, and is based on Eq. (2.5). However, even for the single models, the graph propagation mechanism `AttProp` may be useful, but for that the pruner needs to be jointly trained with the model. This is obtained by augmenting the mention loss $-\log P_{\text{mention}}(E^*|G^\tau)$ with the pruner loss $-\log P_{\text{pruner}}\left(S^*|G^0\right)$ according to Eq. (2.11).

2.4.5.2 Single coreference resolution model

The single-task end-to-end coreference model needs to detect mentions and correctly cluster them. Here again, the standard coreference loss $-\log P_{\text{coref}}\left(C^*|G^\tau\right)$ according to Eq. (2.13) and Eq. (2.7) is extended with the pruner loss $-\log P_{\text{pruner}}(S^*|G^0)$. This turned out essential for correctly predicting the singleton clusters.

2.4.5.3 Single relation extraction model

The single relation extraction model is trained to detect mentions as well as the correct pairwise relations between mentions (i.e., without the coreference step). In order to train the pruner as well, the standard relation score is extended as described in Eq. (2.14) before calculating the loss $-\log P_{\text{relation}}\left(R^*|G^\tau\right)$ based on Eq. (2.9).

2.5 Entity-centric metrics

Unlike the currently widespread datasets that use a mention-driven approach to annotate named entities [45, 46, 90, 91], relations [26, 27, 29, 30, 38, 58, 91] and entity linking [35, 48, 59], DWIE is entirely entity-centric. As explained before, we group entity mentions s_i referring to the same entity into clusters C_k. While we can, and will, adopt the traditional coreference measures as defined by [84] to judge this cluster formation, the NER and relation extraction (RE) evaluation (using precision, recall and F_1) can be done either on (i) mention level, or (ii) entity (cluster) level. The first option however would have the metrics being dominated by the more frequently occurring entities, while the second would penalize mistakes in the clustering (since partially correctly identified clusters would be seen as completely incorrect). This is illustrated in Fig. 2.4 and the corresponding performance metrics in Table 2.6, where scenarios 1 and 2 highlight the effect of making labeling mistakes on the cluster level for different sizes, and scenario 3 highlights the pessimistic view of hard entity-level metrics in case of clustering mistakes. Note that we indicate the mention-level metrics with subscript m, while the (hard) entity-level metrics will have subscript with e.

		Mention-Level			Hard Ent-Level			Soft Ent-Level		
		$\mathbf{Pr_m}$	$\mathbf{Re_m}$	$\mathbf{F_{1,m}}$	$\mathbf{Pr_s}$	$\mathbf{Re_s}$	$\mathbf{F_{1,s}}$	$\mathbf{Pr_e}$	$\mathbf{Re_e}$	$\mathbf{F_{1,e}}$
(a) NER	Gr. Truth	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Scenario 1	0.143	0.100	0.118	0.600	0.500	0.545	0.600	0.500	0.545
	Scenario 2	0.931	0.900	0.915	0.600	0.500	0.545	0.600	0.500	0.545
	Scenario 3	1.000	0.900	0.947	0.333	0.500	0.400	1.000	0.944	0.971
(b) RE	Gr. Truth	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Scenario 1	1.000	0.027	0.053	1.000	0.500	0.667	1.000	0.500	0.667
	Scenario 2	1.000	0.973	0.986	1.000	0.500	0.667	1.000	0.500	0.667
	Scenario 3	0.983	0.783	0.872	0.000	0.000	0.000	0.889	0.889	0.889

Table 2.6: Comparison of different metrics for the example scenarios depicted in Fig. 2.4, for **(a)** NER and **(b)** relation extraction.

Because the (hard) entity-level metrics in our opinion overly penalize clustering mistakes (cf. scenario 3), we propose a variant of entity-level evaluation which we term *soft* entity-level metrics (denoted by subscript s). Basically, instead of adopting a binary count of 1 (all mentions correct) or 0 (as soon as a single mention is missed) on an entity cluster level, we rather count the fraction of its mentions that are correctly labeled. This is illustrated in the formula part of Fig. 2.4(a) for NER, and below we present the adopted formulas in detail. Note that in case clusters are completely predicted correctly, the soft entity-level metrics are the same as hard entity-level metrics (and thus avoid the metric being dominated by frequent mentions, as in the mention-level case).

The formal definition of the metrics depends on counting true positives $tp_p(l)$ and $tp_g(l)$, false positives $fp(l)$, and false negatives $fn(l)$ for a particular NER tag/relation type l, which are specified in Eq. (2.24)–(2.25). These and other notation definitions are summarized in Table 2.8. Further, note that we define two true positives for a particular label l, because of the potential difference between predicted and ground truth clusters: $tp_p(l)$ sums fractions of *predicted* clusters and is used to calculate the precision $\mathrm{Pr_s}$ in Eq. (2.26), while $tp_g(l)$ considers *ground truth* clusters and is used for the recall $\mathrm{Re_s}$ in Eq. (2.26). This allows us to preserve the *cluster-based* relationships between true positives, false positives and false negatives as described for expressions $tp_p(l)+fp(l)$ and $tp_g(l)+fn(l)$ in Table 2.7. Thus our soft entity-level metrics are still cluster-based, while accounting for the mention-level predictions.

$$tp_p(l) = \sum_{C_p \in P_C(l)} \frac{|C_p \cap G_M(l)|}{|C_p|}, \quad tp_g(l) = \sum_{C_g \in G_C(l)} \frac{|C_g \cap P_M(l)|}{|C_g|} \tag{2.24}$$

$$fp(l) = |P_C(l)| - tp_p(l), \qquad fn(l) = |G_C(l)| - tp_g(l) \tag{2.25}$$

Our soft entity-level precision, recall and $\mathrm{F_1}$ metrics are formally de-

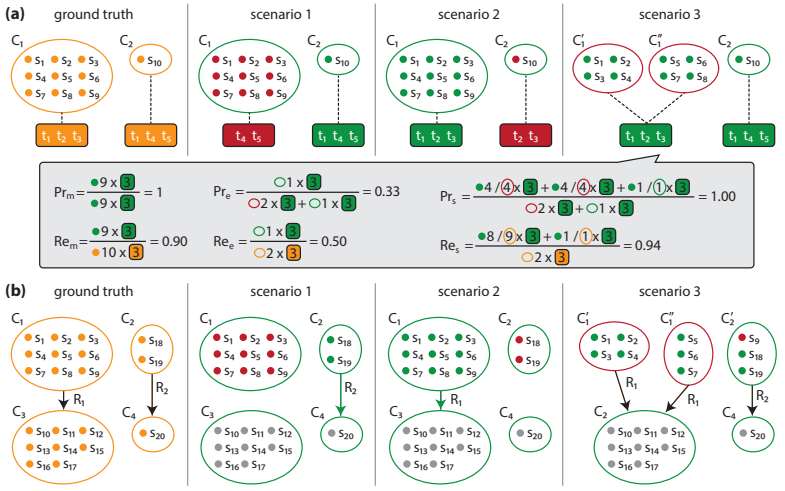


Figure 2.4: Illustration of entity prediction scenarios for **(a)** NER and **(b)** relation extraction, with large clusters (C_1, C_3) and smaller ones (C_2, C_4). *Scenario 1* erroneously labels the large one, *scenario 2* incorrectly labels the small one, *scenario 3* incorrectly splits up the large one and makes a mistake for one of its mentions, s_9. The formulas in the grey box illustrate the calculation of mention-level (Pr_m, Re_m), hard entity-level (Pr_e, Re_e) and soft entity-level (Pr_s, Re_s) precision and recall for NER in *scenario 3*. Note that in (b), the mention dots are colored for correct (green) and incorrect (red) relation heads only.

Expression	(a) Meaning for NER	(b) Meaning for RE
$tp_p(l) + fp(l)$	Number of *predicted* entity clusters with tag l.	Number of *predicted* relations of type l between entity clusters.
$tp_g(l) + fn(l)$	Number of *ground truth* entity clusters with tag l.	Number of *ground truth* relations of type l between entity clusters.

Table 2.7: The relations between the weighted true positives by the size of predicted ($tp_p(l)$) and ground truth ($tp_g(l)$) entity clusters allows us to achieve the constraints needed for the denominators of precision ($tp_p(l) + fp(l)$) and recall ($tp_p(l) + fn(l)$) functions (Eq. (2.26)) in terms of the number of entity clusters.

fined as follows, where L refers to either the number of all possible tags for NER or the number of all possible relation types for RE:

$$\mathrm{Pr_s} = \frac{\sum_{l=1}^{L} tp_p(l)}{\sum_{l=1}^{L} tp_p(l) + fp(l)}, \qquad \mathrm{Re_s} = \frac{\sum_{l=1}^{L} tp_g(l)}{\sum_{l=1}^{L} tp_g(l) + fn(l)}, \qquad \mathrm{F_{1,s}} = \frac{2 \cdot \mathrm{Pr_s} \cdot \mathrm{Re_s}}{\mathrm{Pr_s} + \mathrm{Re_s}} \tag{2.26}$$

2.6 Experimental results

2.6.1 Experimental setup

We train and evaluate our model as described in Section 2.4 on three tasks: NER, coreference, and relation extraction (RE) independently and jointly. We experiment with three main model variations:

1. **Single**: Experiments on individual tasks by training with the respective loss functions as described in Section 2.4.5.

2. **Joint**: Experiments jointly on all three tasks using pre-trained *GloVe representations*[14] concatenated to character embeddings in the shared input layer (see Fig. 2.3). For training we use the joint loss defined in Section 2.4.2.

3. **Joint+BERT**: as in the *Joint* setting, experiments jointly on all three tasks, but using pre-trained $\mathrm{BERT_{BASE}}$ embeddings[15] concatenated to the GloVe and character embeddings. We use an input window

[14] http://nlp.stanford.edu/data/glove.840B.300d.zip
[15] https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip

Symbol	(a) Meaning for NER	(b) Meaning for RE
$P_C(l)$	Set of predicted entity clusters with tag l.	Set of predicted relations of type l between the predicted entity clusters.
$C_p \in P_C(l)$	Set of predicted entity mentions for a particular entity cluster in $P_C(l)$.	Set of relations between the predicted entity mentions for a particular pair of related entity clusters in $P_C(l)$.
$G_C(l)$	Set of ground truth entity clusters annotated with tag l.	Set of ground truth relations of type l between the ground truth entity clusters.
$C_g \in G_C(l)$	Set of ground truth entity mentions for a particular entity cluster in $G_C(l)$.	Set of relations between the ground truth entity mentions for a particular pair of related entity clusters in $G_C(l)$
$P_M(l)$	Set of predicted entity mentions with tag l.	Set of predicted relations of type l between the predicted entity mentions.
$G_M(l)$	Set of ground truth entity mentions annotated with tag l.	Set of ground truth relations of type l between the ground truth entity mentions.
$tp_p(l)$	Number of true positive predictions of tag l on mentions re-weighted by predicted cluster sizes.	Number of true positive predictions of relation type l between mentions re-weighted by the number of mention level relations between the connected pairs of predicted clusters.
$tp_g(l)$	Number of true positive mention level predictions of tag l re-weighted by ground truth cluster sizes.	Number of true positive predictions of relation type l between mentions re-weighted by the number of mention level relations between the connected pairs of ground truth clusters.
$fp(l)$	Number of false positive mention level predictions of tag l re-weighted by predicted cluster sizes.	Number of false positive predictions of relation type l between mentions re-weighted by the number of mention level relations between the connected pairs of predicted clusters.
$fn(l)$	Number of false negative mentions with ground truth tag l re-weighted by ground truth cluster sizes.	Number of false negative relations of type l between mentions re-weighted by the number of mention level relations between the connected pairs of ground truth clusters.

Table 2.8: Short definition of the symbols and expressions involved in our *soft-entity level* metric formulation in Eq. (2.24)–(2.26) for both NER and RE tasks.

	Coreference F_1				NER F_1			RE F_1		
Model Setup	**MUC**	**CEAF$_e$**	**B^3**	**Avg.**	**F$_{1,m}$**	**F$_{1,e}$**	**F$_{1,s}$**	**F$_{1,m}$**	**F$_{1,e}$**	**F$_{1,s}$**
Single	92.8	90.9	88.2	90.6	85.7	-	-	68.2	-	-
+AttProp	**93.2**	**91.5**	**88.7**	**91.1**	**87.1**	-	-	**71.3**	-	-
+CorefProp	92.8	90.9	88.3	90.7	-	-	-	-	-	-
+RelProp	-	-	-	-	-	-	-	68.2	-	-
Joint	92.5	**90.5**	**87.3**	**90.1**	85.4	71.7	84.4	68.1	46.8	66.5
+AttProp	92.3	90.4	87.3	90.0	87.1	72.9	**86.1**	**72.1**	**50.4**	**72.1**
+CorefProp	92.3	90.3	87.2	89.9	**87.2**	**73.2**	86.0	71.6	50.2	71.0
+RelProp	**92.6**	90.2	86.8	89.9	86.7	72.4	85.2	69.5	48.2	68.8
Joint+BERT	**93.8**	**92.1**	**89.0**	**91.6**	87.6	74.2	86.4	70.6	48.7	68.9
+AttProp	93.2	91.4	88.6	91.1	**88.8**	74.2	**87.7**	72.3	**50.4**	**73.0**
+CorefProp	93.5	91.8	88.7	91.3	88.7	74.4	87.4	**72.7**	50.0	71.9
+RelProp	93.7	91.8	88.7	91.4	88.4	**74.8**	87.0	72.0	49.9	71.4

Table 2.9: Main results of the experiments grouped in three model setups: (i) *Single* models trained individually, (ii) *Joint* model trained using as input GloVe and character embeddings, and (iii) *Joint+BERT* model trained on $\text{BERT}_{\text{BASE}}$ embeddings. To report the results, we use MUC, CEAF_e, B^3 as well as the average (Avg.) of these three metrics for *coreference resolution*. For NER and RE we use mention-level ($F_{1,m}$), hard entity-level ($F_{1,e}$), and soft entity-level ($F_{1,s}$) metrics described in Section 2.5. In bold we mark the best results for each model setup, the best overall results are underlined. Note that the metrics are expressed in percentage points.

size of 250 tokens and concatenate the last 2 hidden layers of BERT to get token representations.

Additionally, for each of the three model setups we experiment with the graph propagation techniques defined in Section 2.4.4. To maximize result consistency, we train each model 5 times and report the average of these 5 results for each of the experiments.

We use a single-layer BiLSTM with forward and backward hidden states of 200 dimensions each. All our FFNNs used to obtain confidence scores ($\mathcal{F}_{\text{pruner}}$, $\mathcal{F}_{\text{coref}}$, $\boldsymbol{\mathcal{F}}_{\text{mention}}$, $\boldsymbol{\mathcal{F}}_{\text{relation}}$, and $\mathcal{F}_{\text{att}}$) have two 150-dimensional hidden layers trained with a dropout of 0.4. We set the maximum span width w_{max} to 5 and the pruner ratio to 0.2 of the total number of tokens in a document. For training, we use Adam with a learning rate of $1e-3$ for 100 epochs with a linear decay of 0.1 starting at epoch 15.

2.6.2 Results and analyses

Table 2.9 gives an overview of the results achieved in *Single* as well as *Joint* and *Joint + BERT* setups. Additionally, Fig. 2.5 illustrates the impact of the

number of graph propagation iterations for each of the span graph propagation methods on the final results.

First, we observe a general improvement in all our *Single* tasks when using graph propagation techniques. More specifically, our proposed latent `AttProp` achieves superior results compared to the relation (`RelProp`) and coreference (`CorefProp`) propagations when added to the *Single* setup. The biggest improvement across iterations (see Fig. 2.5) is for the single RE task mention-level $F_{1,m}$ score with a boost of $\sim$3 percentage points when incorporating `AttProp`. We also observe an improvement of $\sim$1.5 percentage points in $F_{1,m}$ for the NER task and a consistent but smaller improvement of 0.5 F_1 percentage points for the coreference task. These results illustrate the effectiveness of `AttProp` when applied to single task models.

A further improvement in results is achieved by training our model *jointly* (see the Joint setup in Table 2.9 and graphs in Fig. 2.5) for NER and RE tasks. This illustrates that, besides the positive effect of neural graph propagation on single task models, training our model jointly has an additional benefit by exploiting the interaction between tasks. In particular, this effect can be seen for RE, where our *Joint* model achieves a boost in performance of 0.8 percentage points for the mention-level $F_{1,m}$ metric compared to the best result for the *Single* setup. Furthermore, our `AttProp` graph propagation method achieves the best performance on all the metrics for the RE task in the *Joint* setting with up to $\sim$ 5.5 percentage points improvement in our newly proposed $F_{1,s}$ metric. Additionally, we observe a beneficial effect of graph propagation for the NER task in the *Joint* setup with slightly better results for the $F_{1,m}$ metric compared to the *Single* setting. Our `AttProp` technique performs on par with `CorefProp`, outperforming the latter by a small margin in terms of $F_{1,s}$ metric.

Similarly to the *Joint* model variation, we observe benefits when using graph propagation techniques in the *Joint+BERT* models. Table 2.10 illustrates the deltas in performance for the NER and relation extraction tasks. This way, we can see more clearly the difference in impact of our neural message passing methods grouped by the model setup and metric type. First, we observe that the general performance boost from using graph propagation techniques is lower in *Joint+BERT* than in the *Joint* setup. We hypothesize that this effect is due to the fact that BERT itself has a better long-range context extraction due to the attention-based mechanism, which spans the input window as opposed to purely local (non-contextualized) GloVe embeddings used in the *Joint* setting. This is in line with the findings in [92], [43], and [93] that show the advantage of using large BERT input window sizes to produce better IE results. Second, we observe that our `AttProp` method achieves consistently superior performance on our proposed soft entity-level metric $F_{1,s}$, capturing thus better the mention-based predictions as weighted by their cluster sizes. Finally, from Table 2.10(b) we notice that adding BERT to our joint model does not affect the boost in performance caused by the `RelProp` method for relation

		Joint			Joint+BERT		
		$\mathbf{F_{1,m}}$	$\mathbf{F_{1,e}}$	$\mathbf{F_{1,s}}$	$\mathbf{F_{1,m}}$	$\mathbf{F_{1,e}}$	$\mathbf{F_{1,s}}$
(a) NER	Δ `AttProp`	1.69	1.18	**1.67**	**1.16**	−0.02	**1.31**
	Δ `CorefProp`	**1.78**	**1.50**	1.54	1.05	0.20	1.02
	Δ `RelProp`	1.33	0.70	0.75	0.78	**0.56**	0.60
(b) RE	Δ `AttProp`	**3.97**	**3.62**	**5.56**	1.66	**1.69**	**4.05**
	Δ `CorefProp`	3.48	3.45	4.47	**2.02**	1.29	2.95
	Δ `RelProp`	1.35	1.47	2.32	1.37	1.20	2.48

Table 2.10: Deltas of improvement in performance for each of the graph propagation methods (`AttProp`, `CorefProp`, `RelProp`) in F_1 scores for **(a)** NER and **(b)** relation extraction tasks.

extraction. We hypothesize that this is due to the fact that `RelProp` propagation can capture relational semantics that goes beyond BERT's contextual span representation similarity (which mainly drives the positive impact of *Joint+BERT*).

Unlike for the NER and RE tasks, where we observe a consistent positive impact of span graph propagation and joint modeling across all our experiments, the impact on the *coreference* task is not clear. Our experiments on *Single* setup show small, but constant improvement of the Avg.-F_1 score with the number of `AttProp` propagation iterations (see Fig. 2.5). However, in our *Joint* and *Joint+BERT* setups the graph propagation appears to not have any positive impact on Avg.-F_1 coreference scores. We hypothesize that the main reason for this phenomenon lies in the coreference annotations in DWIE: since we only annotate clusters of proper nouns, leaving out the nominal (e.g., "the prime minister") and anaphoric expressions (e.g., "he", "she", "they", etc), there might be little to no additional benefit in propagating information between co-referenced entity mentions, since the representation of proper nouns likely is not much influenced by textual context (e.g., the span "Merkel" can have very similar span representation to "Angela Merkel", gaining nothing in adding contextual graph propagation).

Additionally, we explore in more detail the effect of the number of `AttProp`, `CorefProp`, `RelProp` graph propagation iterations on the final F_1 score of all the tasks in Fig. 2.5. We observe that the number of iterations have a decreasing effect on the improvement of performance for the NER and RE tasks. Furthermore, the positive effect of `CorefProp` and `RelProp` tends to saturate or even become negative after 1 or 2 iterations. This is in line with findings of [21] on other datasets, where the performance peak is usually achieved at 2 graph propagation iterations. For our `AttProp` however, we observe that the positive effect of additional iterations tends to

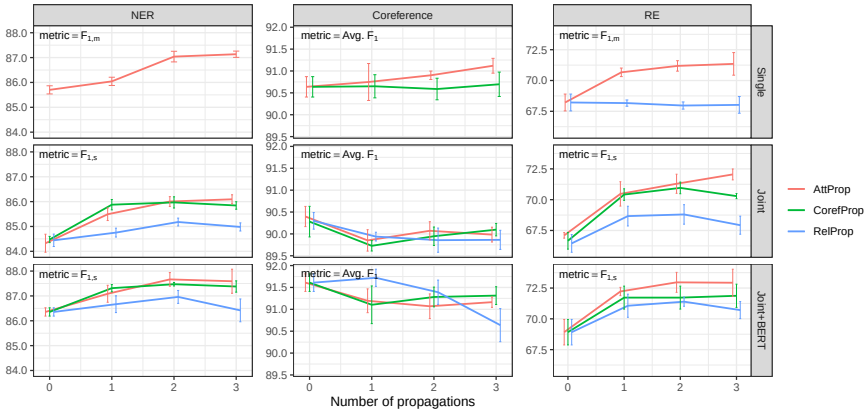


Figure 2.5: Impact of `AttProp`, `CorefProp` and `RelProp` graph propagations on performance metrics for each of the *Single*, *Joint* and *Joint+BERT* model setups. Note the different Y-axis scales.

persist longer, particularly on the *Joint* setup where the positive effect of `AttProp` seems to be still growing after the last iteration (3) in our experiments.

2.7 Conclusion and future work

In this work we introduced DWIE, a manually annotated multi-task dataset that comprises Named Entity Recognition, Coreference, Relation Extraction and Entity Linking as main tasks. We highlight how DWIE is different from the mainstream datasets by focusing on document-level and entity-centric annotations. This also makes the predictions on this dataset more challenging by having not only to consider explicit, but also implicit document-level interactions between entities. Furthermore, we showed how Graph Neural Networks can help to tackle this issue by propagating local contextual mention span information on a document level for a single task as well as across the tasks on the DWIE dataset. We experiment with known graph propagation techniques driven by the scores of the coreference resolution (`CorefProp`) and relation extraction (`RelProp`) components, as well as introduced a new latent task-independent attention-based graph propagation method (`AttProp`). We demonstrated that, without relying on the task-specific scorers, `AttProp` can boost the performance of single-task as well as joint models, performing on par and even outperforming significantly in some scenarios the `RelProp` and `CorefProp` graph propagations.

In future work we will aim to integrate an entity linking component into our joint architecture. As a consequence, we expect to obtain a fur-

ther boost in performance of different tasks included in DWIE by taking advantage of the information coming from Wikipedia 2018, the reference knowledge base for the entity linking annotations. Conversely, we conjecture that the results of the entity linking component can be improved when training it jointly with other tasks, such as NER and coreference resolution. Finally, we plan extending the coreference annotations to include nominal and anaphoric expressions. We expect that including these diverse mention types, whose initial span embedding representation can be different from coreferenced named entities, will make our coreference resolution task more challenging, allowing to investigate further the potential benefits of using graph-based neural networks.

Acknowledgements

Part of the research leading to these results has received funding from (i) the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761488 for the CPN project,[16] and (ii) the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

[16] https://www.projectcpn.eu/

Appendix

Dataset insights

Entity Type	# Entities	% Entities	# Mentions	% Mentions
ENTITY	*13,151*	*56.9%*	*30,719*	*70.8%*
location	4,957	21.4%	11,548	26.6%
gpe	3,965	17.1%	9,830	22.7%
gpe0	2,225	9.6%	6,559	15.1%
gpe2	1,497	6.5%	2,873	6.6%
gpe1	244	1.1%	406	0.9%
regio	479	2.1%	916	2.1%
facility	259	1.1%	385	0.9%
organization	3,434	14.8%	8,165	18.8%
media	659	2.8%	984	2.3%
igo	547	2.4%	1,992	4.6%
so	171	0.7%	912	2.1%
party	381	1.6%	949	2.2%
company	368	1.6%	932	2.1%
sport_team	367	1.6%	1,106	2.5%
governmental_organization	342	1.5%	636	1.5%
agency	228	1.0%	444	1.0%
armed_movement	108	0.5%	374	0.9%
person	3,390	14.7%	8,259	19.0%
politician	1,184	5.1%	3,326	7.7%
head_of_state	380	1.6%	1,271	2.9%
head_of_gov	247	1.1%	673	1.6%
minister	217	0.9%	458	1.1%
sport_player	405	1.8%	844	1.9%
artist	260	1.1%	586	1.4%
politics_per	209	0.9%	457	1.1%
manager	104	0.4%	297	0.7%
offender	75	0.3%	347	0.8%
misc	823	3.6%	1,646	3.8%
work_of_art	174	0.8%	247	0.6%
event	354	1.5%	701	1.6%
sport_competition	183	0.8%	410	0.9%
ethnicity	84	0.4%	242	0.6%
VALUE	*5,903*	*25.5%*	*7,104*	*16.4%*
time	2,907	12.6%	3,608	8.3%
role	2,390	10.3%	2,865	6.6%
money	606	2.6%	631	1.5%
OTHER	*2,724*	*11.8%*	*5,482*	*12.6%*
gpe0-x	1,596	6.9%	3,827	8.8%
footer	413	1.8%	413	1.0%
loc-x	353	1.5%	585	1.3%
religion-x	235	1.0%	486	1.1%
TOTAL	23,130	100.0%	43,373	100.0%

Table 2.11: Statistics depicting the hierarchical structure of entity types described in Section 2.3.2. Only the most frequent entity types/subtypes are shown (% Mentions $> 0.5\%$)

	person	organization	event	location		mi
politics	head_of_gov, head_of_state, minister, politician_regional, politician_local, politician_national, candidate, politician, politics_per, activist, gov_per	politics_institution, politics_org, party, ngo, igo, so, policy_institute, movement, agency, ministry, military_alliance	summit_meeting, scandal, politics_event	politics_facility	politics_misc, project, treaty, re	
culture	character, culture_per, artist, writer, actor, filmmaker, musician, photographer	music_band, culture_org, theatre_org, dance_org	festival, film_festival	culture_facility	art_title, culture_title, exhibiti work_of_art, book_title, film_ theatre_title, musical_title, film music_award, tv_award, colu radio_title, dance_title, opera	
education	teacher, education_per, education_student	education_org		education_facility	education_study	
religion	deity, clergy	religion_org	religious_event	religion_facility	religion, religion_misc	
human	royalty				film_award, book_award, awa sport_award	
conflict	military_personnel, military_rebel	army, military_alliance, armed_movement	war, protest	military_facility	military_equipment, military_	
media	journalist	media				
science	researcher, science_per	research_center			species, research_journal, tech	
sport	sport_player, sport_coach, sport_head, sport_referee, sport_person	sport_team, sport_org	sport_competition	sport_facility	sport_award	
labor	union_head, union_member, union_rep, union_per	union				
business	manager, employee, business_per	company, business_org, brand, trade_fair, market_exchange, advocacy		business_facility	product, market_index, busine	
health	health_per	health_org		health_facility	health_disease, health_drug	
justice	offender, advisor, victim, judge, police_per, justice_per	court, criminal_org, police_org, justice_org		prison	justice_misc, case	
weather			storm			

Table 2.12: Illustration of NER entity types in DWIE. Each cells contains possible entity subtypes (of different hierarchy levels) corresponding to the respective parent entity type (column) and topic (row).

Table 2.13 describes the statistics of linked entities with respect to the total number of entities in each of the *Entity* subtypes. The columns *% Linked Entities* and *% Linked Mentions* indicate the percentage of annotated linked entities and mentions with respect to the total number of annotated entities/mentions in a particular *Entity* type category. Furthermore, we calculate two accuracies on test split when linking the entity mention with the most frequent entity link used either in DWIE: (i) training set of DWIE dataset ("Acc. Prior Train"), or (ii) Wikipedia corpus ("Acc. Prior Wiki"). Overall, using prior linking annotations from Wikipedia gives 9 percentage points better performance (79.0%) than when using train set (70.0%). This difference is explained by the fact that Wikipedia has much larger corpus to calculate the prior linking information from. Nevertheless, we still observe that for some entity types such as *sport_team* and *media* the accuracy based on DWIE training set prior is higher. This suggests the use of domain-specific language to refer to some entities in DWIE not used in a more general Wikipedia domain.

Entity Type	# Linked Entities	% Linked Entities	# Linked Mentions	% Linked Mentions	Acc. Prior Train	Acc. Prior Wiki
LOCATION	*4,863*	*98.1%*	*11,496*	*99.5%*	*85.7%*	*92.9%*
gpe	3,938	99.3%	9,810	99.8%	89.8%	95.6%
regio	456	95.2%	889	97.1%	83.3%	76.3%
facility	229	88.4%	381	99.0%	19.7%	73.8%
waterbody	90	98.9%	145	100.0%	83.3%	91.7%
district	37	94.9%	45	100.0%	33.3%	33.3%
ORGANIZATION	*3,145*	*91.6%*	*8,029*	*98.3%*	*69.8%*	*70.8%*
media	622	94.4%	979	99.5%	81.8%	59.5%
igo	525	96.0%	1,952	98.0%	76.4%	78.8%
party	358	94.0%	897	94.5%	77.5%	66.7%
company	320	87.0%	923	99.0%	67.6%	89.7%
sport_team	366	99.7%	1,105	99.9%	71.0%	47.5%
PERSON	*2,627*	*77.5%*	*8,217*	*99.5%*	*45.7%*	*69.4%*
politician	1,162	98.1%	3,324	99.9%	66.0%	78.1%
sport_player	404	99.8%	843	99.9%	34.4%	71.3%
artist	246	94.6%	567	96.8%	0.0%	29.4%
politics_per	126	60.3%	456	99.8%	23.7%	42.1%
manager	58	55.8%	296	99.7%	22.2%	33.3%
MISC	*607*	*73.8%*	*1,532*	*93.1%*	*58.4%*	*73.4%*
work_of_art	142	81.6%	246	99.6%	0.0%	100.0%
award	72	80.0%	186	94.9%	63.6%	81.8%
treaty	60	74.1%	149	99.3%	66.7%	50.0%
product	50	76.9%	146	98.6%	52.0%	92.0%
species	10	25.0%	14	18.4%	0.0%	100.0%
EVENT	*320*	*90.4%*	*683*	*97.4%*	*49.4%*	*67.1%*
sport_competition	163	89.1%	397	96.8%	64.6%	87.5%
summit_meeting	15	68.2%	37	92.5%	100.0%	100.0%
holiday	21	95.5%	39	97.5%	100.0%	100.0%
history	17	89.5%	30	100.0%	100.0%	100.0%
protest	14	100.0%	22	100.0%	80.0%	100.0%
TOTAL	**13,086**	**56.6%**	**28,482**	**65.7%**	**70.0%**	**79.0%**

Table 2.13: Entity linking statistics, only the top 5 types and subtypes with largest number of linked entities are showed. The *total* is calculated on all the entity types. The accuracy (both for most likely prior links on train and Wiki corpora) is computed on test set.

Entity Tag Category	# Entities	% Entities	# Mentions	% Mentions	# Classes	Labels per Entity
type	21,745	94.0%	43,122	99.4%	174	2.9
topic	7,843	33.9%	18,359	42.3%	14	1.0
iptc	7,059	30.5%	17,195	39.6%	114	1.3
gender	3,352	14.5%	8,200	18.9%	2	1.0
slot	3,232	14.0%	14,983	34.5%	7	1.2
TOTAL	23,130	100.0%	43,373	100.0%	311	4.0

Table 2.14: Main named entity tag categories with statistics of the number and % of covered entities and mentions as well as the number of classes in each and average number of labels per entity cluster.

Table 2.14 illustrates the number of annotated entities and mentions per each tag category (type, topic, iptc, gender and slot). It also showcases the multi-label nature of entity classification task in DWIE, with the average number of labels per entity of 4.0.

Table 2.15 illustrates the number and percentage of related entities and mentions of our dataset grouped by the number of relation labels. It also compares with other entity-centric RE datasets, namely BC5CDR [52, 53] and DocRED [34] datasets.

	DWIE				BC5CDR	DocRED
# Relation labels	# Related ent. pairs	% Related ent. pairs	# Related mention pairs	% Related mention pairs	% Related ent. pairs	% Related ent. pairs
1	12,856	76.32%	112,708	69.40%	100%	92.89%
2	3,101	18.41%	34,948	21.52%	0%	6.82%
3	884	5.25%	14,650	9.02%	0%	0.26%
4	3	0.02%	100	0.06%	0%	0.03%
TOTAL	16,844	100.0%	162,406	100.0%	100.0%	100.0%

Table 2.15: This table groups the number of related pairs in DWIE by the number of assigned relation labels to each of these pairs. We compare with other two entity-centric datasets: BC5CDR and DocRED.

Relation Type	# Related Ent. Pairs	% Related Ent. Pairs	# Related Men. Pairs	% Related Men. Pairs
based_in0	2,361	14.0%	18,771	11.6%
in0	2,120	12.6%	15,810	9.7%
citizen_of	1,969	11.7%	25,752	15.9%
based_in0-x	1,882	11.2%	12,211	7.5%
citizen_of-x	1,844	10.9%	17,049	10.5%
member_of	1,616	9.6%	19,953	12.3%
gpe0	1,569	9.3%	18,110	11.2%
in0-x	1,474	8.8%	8,784	5.4%
agent_of	954	5.7%	15,776	9.7%
head_of	564	3.3%	7,710	4.7%
agency_of	435	2.6%	4,775	2.9%
player_of	401	2.4%	5,692	3.5%
agency_of-x	382	2.3%	2,108	1.3%
head_of_state	380	2.3%	7,986	4.9%
head_of_state-x	343	2.0%	3,853	2.4%
appears_in	294	1.7%	4,555	2.8%
vs	281	1.7%	7,187	4.4%
head_of_gov	273	1.6%	4,015	2.5%
head_of_gov-x	247	1.5%	2,383	1.5%
minister_of	234	1.4%	2,280	1.4%
minister_of-x	213	1.3%	1,629	1.0%
based_in2	185	1.1%	971	0.6%
event_in0	181	1.1%	843	0.5%
part_of	164	1.0%	2,858	1.8%
in2	157	0.9%	1,055	0.6%
created_by	134	0.8%	945	0.6%
agent_of-x	125	0.7%	897	0.6%
award_received	111	0.7%	969	0.6%
institution_of	105	0.6%	2,113	1.3%
ministry_of	81	0.5%	666	0.4%
coach_of	65	0.4%	1,211	0.7%
won_vs	61	0.4%	1,531	0.9%
spouse_of	55	0.3%	599	0.4%
directed_by	44	0.3%	318	0.2%
is_meeting	41	0.2%	968	0.6%
event_in2	40	0.2%	259	0.2%
spokesperson_of	39	0.2%	177	0.1%
plays_in	38	0.2%	330	0.2%
gpe1	35	0.2%	135	0.1%
product_of	31	0.2%	334	0.2%
parent_of	22	0.1%	281	0.2%
child_of	22	0.1%	281	0.2%
based_in1	22	0.1%	376	0.2%
signed_by	20	0.1%	521	0.3%
law_of	16	0.1%	286	0.2%
TOTAL	16,844	100.0%	162,406	100.0%

Table 2.16: Relation type statistics. We compare the number of related entity and mention pairs per relation type. Only the most frequent relation types are shown (% Related Mention Pairs $> 0.1\%$)

Inter-annotator agreement calculations

In order to measure the agreement we use Cohen's kappa coefficient [94], defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.27}$$

where p_o represents the observed agreement between the two annotators and p_e is the expected agreement between the annotators (i.e., agreement by chance). More specifically, in our case we calculate the observed probability p_o as in Eq. (2.28) where N is the number of annotated items, $A_{i,j}$ is the annotation made by annotator i for item j, and $\mathbb{1}\{A_{1,j} = A_{2,j}\}$ returns 1 if $A_{1,j}$ is equal to $A_{2,j}$ and 0 otherwise. Thus, p_o can be interpreted as the fraction of the labels two annotators agree, also called *percent agreement* [75, 95].

$$p_o = \frac{\sum_{j=1}^{N} \mathbb{1}\{A_{1,j} = A_{2,j}\}}{N} \tag{2.28}$$

To calculate the expected agreement probability we use the formulation in Eq. (2.29). It can be interpreted as the probability that both annotators, when randomly distributing all of their label annotations among the items to be annotated, assign the same label to a given item.

$$p_e = \sum_{l=1}^{L} \frac{n_{1,l}}{N} \frac{n_{2,l}}{N} \tag{2.29}$$

In this context, $n_{i,l}$ is the number of items the annotator i annotated with label l and L is the total number of labels. For multi-label annotations where it is possible to assign multiple classes for a particular annotation item (i.e., named entity and relation types), we report a weighted kappa score.

Relation consistency rules

This appendix enumerates the logical predicates used as a consistency check in our dataset.

$$\text{spouse\_of}\langle Y, X\rangle \implies \text{spouse\_of}\langle X, Y\rangle \tag{2.1}$$

$$\text{vs}\langle Y, X\rangle \implies \text{vs}\langle X, Y\rangle \tag{2.2}$$

$$\text{won\_vs}\langle X, Y\rangle \implies \text{vs}\langle X, Y\rangle \tag{2.3}$$

$$\text{won\_vs}\langle X, Y\rangle \implies \text{vs}\langle Y, X\rangle \tag{2.4}$$

$$\text{child\_of}\langle Y, X\rangle \implies \text{parent\_of}\langle X, Y\rangle \tag{2.5}$$

$$\text{parent\_of}\langle Y, X\rangle \implies \text{child\_of}\langle X, Y\rangle \tag{2.6}$$

$$\text{ministry\_of}\langle X, Y\rangle \implies \text{agency\_of}\langle X, Y\rangle \tag{2.7}$$

$$\text{agency\_of-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{agency\_of}\langle X, Y\rangle \tag{2.8}$$

$$\text{agency\_of}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{agency\_of-x}\langle X, Z\rangle \tag{2.9}$$

$$\text{agent\_of-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{agent\_of}\langle X, Y\rangle \tag{2.10}$$

$$\text{agent\_of}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{agent\_of-x}\langle X, Z\rangle \tag{2.11}$$

$$\text{minister\_of}\langle X, Y\rangle \implies \text{agent\_of}\langle X, Y\rangle \tag{2.12}$$

$$\text{head\_of\_gov}\langle X, Y\rangle \implies \text{agent\_of}\langle X, Y\rangle \tag{2.13}$$

$$\text{head\_of\_state}\langle X, Y\rangle \implies \text{agent\_of}\langle X, Y\rangle \tag{2.14}$$

$$\text{citizen\_of-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{citizen\_of}\langle X, Y\rangle \tag{2.15}$$

$$\text{citizen\_of}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{citizen\_of-x}\langle X, Z\rangle \tag{2.16}$$

$$\text{minister\_of-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{minister\_of}\langle X, Y\rangle \tag{2.17}$$

$$\text{minister\_of}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{minister\_of-x}\langle X, Z\rangle \tag{2.18}$$

$$\text{head\_of\_state-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{head\_of\_state}\langle X, Y\rangle \tag{2.19}$$

$$\text{head\_of\_state}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{head\_of\_state-x}\langle X, Z\rangle \tag{2.20}$$

$$\text{head\_of\_gov-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{head\_of\_gov}\langle X, Y\rangle \tag{2.21}$$

$$\text{head\_of\_gov}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{head\_of\_gov-x}\langle X, Z\rangle \tag{2.22}$$

$$\text{in0-x}\langle X, Z\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{in0}\langle X, Y\rangle \tag{2.23}$$

$$\text{in0}\langle X, Y\rangle \wedge \text{gpe0}\langle Z, Y\rangle \implies \text{in0-x}\langle X, Z\rangle \tag{2.24}$$

$$\text{in2}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{in0}\langle X, Y\rangle \tag{2.25}$$

$$\text{in1}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{in0}\langle X, Y\rangle \tag{2.26}$$

$$\text{based\_in2}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{based\_in0}\langle X, Y\rangle \tag{2.27}$$

$$\text{based\_in1}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{based\_in0}\langle X, Y\rangle \tag{2.28}$$

$$\text{agency\_of}\langle X, Y\rangle \wedge \text{gpe0}\langle Y\rangle \implies \text{based\_in0}\langle X, Y\rangle \tag{2.29}$$

$$\text{event\_in2}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{event\_in0}\langle X, Y\rangle \tag{2.30}$$

$$\text{event\_in1}\langle X, Z\rangle \wedge \text{in0}\langle Z, Y\rangle \implies \text{event\_in0}\langle X, Y\rangle \tag{2.31}$$

$$\text{head\_of}\langle X, Y\rangle \implies \text{member\_of}\langle X, Y\rangle \tag{2.32}$$

$$\text{coach\_of}\langle X, Y\rangle \implies \text{member\_of}\langle X, Y\rangle \tag{2.33}$$

$$\text{spokesperson\_of}\langle X, Y\rangle \implies \text{member\_of}\langle X, Y\rangle \tag{2.34}$$

$$\text{member\_of}\langle X, Y\rangle \wedge \text{sport\_player}\langle X\rangle \implies \text{player\_of}\langle X, Y\rangle \tag{2.35}$$

$$\mathsf{mayor\_of}\langle X, Y\rangle \implies \mathsf{head\_of\_gov}\langle X, Y\rangle \tag{2.36}$$

$$\mathsf{directed\_by}\langle X, Y\rangle \implies \mathsf{created\_by}\langle X, Y\rangle \tag{2.37}$$

$$\mathsf{character\_in}\langle X, Y\rangle \wedge \mathsf{played\_by}\langle X, Z\rangle \implies \mathsf{plays\_in}\langle Z, Y\rangle \tag{2.38}$$

$$\mathsf{institution\_of}\langle X, Y\rangle \implies \mathsf{part\_of}\langle X, Y\rangle \tag{2.39}$$

$$\mathsf{based\_in0\text{-}x}\langle X, Z\rangle \wedge \mathsf{gpe0}\langle Z, Y\rangle \implies \mathsf{based\_in0}\langle X, Y\rangle \tag{2.40}$$

$$\mathsf{based\_in0}\langle X, Y\rangle \wedge \mathsf{gpe0}\langle Z, Y\rangle \implies \mathsf{based\_in0\text{-}x}\langle X, Z\rangle \tag{2.41}$$

References

[1] M. Yu, W. Yin, K. S. Hasan, C. dos Santos, B. Xiang, and B. Zhou. *Improved Neural Relation Detection for Knowledge Base Question Answering*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 571–581, 2017.

[2] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. *Language-conditioned graph networks for relational reasoning*. In Proceedings of the IEEE International Conference on Computer Vision, pages 10294–10303, 2019.

[3] Y. Gao, P. Li, I. King, and M. R. Lyu. *Interconnected Question Generation with Coreference Alignment and Conversation Flow Modeling*. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4853–4862, 2019.

[4] S. Bhattacharjee, R. Haque, G. M. de Buy Wenniger, and A. Way. *Investigating query expansion and coreference resolution in question answering on BERT*. In International Conference on Applications of Natural Language to Information Systems, pages 47–59. Springer, 2020.

[5] D. Molla, M. van Zaanen, and D. Smith. *Named Entity Recognition for Question Answering*. In Proceedings of the Australasian Language Technology Workshop 2006, pages 51–58, 2006.

[6] K. Singh, A. S. Radhakrishna, A. Both, S. Shekarpour, I. Lytra, R. Usbeck, A. Vyas, A. Khikmatullaev, D. Punjani, C. Lange, et al. *Why reinvent the wheel: Let's build question answering systems together*. In Proceedings of the 2018 World Wide Web Conference, pages 1247–1256, 2018.

[7] D. Chen, A. Fisch, J. Weston, and A. Bordes. *Reading Wikipedia to Answer Open-Domain Questions*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1870–1879, 2017.

[8] S. Broscheit. *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, 2019.

[9] H. Wang, F. Zhang, X. Xie, and M. Guo. *DKN: Deep knowledge-aware network for news recommendation*. In Proceedings of the 2018 world wide web conference, pages 1835–1844, 2018.

[10] M. Karimi, D. Jannach, and M. Jugovac. *News recommender systems–Survey and roads ahead*. Information Processing & Management, 54(6):1203–1227, 2018.

[11] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo. *Multi-task feature learning for knowledge graph enhanced recommendation*. In The World Wide Web Conference, pages 2000–2010, 2019.

[12] J. Thorne and A. Vlachos. *Automated Fact Checking: Task Formulations, Methods and Future Directions*. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3346–3359, 2018.

[13] X. Zhang and A. A. Ghorbani. *An overview of online fake news: Characterization, detection, and discussion*. Information Processing & Management, 57(2):102025, 2020.

[14] S. Sun, C. Luo, and J. Chen. *A review of natural language processing techniques for opinion mining systems*. Information fusion, 36:10–25, 2017.

[15] P. Cifariello, P. Ferragina, and M. Ponza. *Wiser: A semantic approach for expert finding in academia based on entity linking*. Information Systems, 82:1–16, 2019.

[16] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, et al. *Recipes for building an open-domain chatbot*. arXiv preprint arXiv:2004.13637, 2020.

[17] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *Joint entity recognition and relation extraction as a multi-head selection problem*. Expert Systems with Applications, 114:34–45, 2018.

[18] H. Fei, Y. Ren, and D. Ji. *Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction*. Information Processing & Management, 57(6):102311, 2020.

[19] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. *End-to-end Neural Coreference Resolution*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 188–197, 2017.

[20] K. Lee, L. He, and L. Zettlemoyer. *Higher-Order Coreference Resolution with Coarse-to-Fine Inference*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 687–692, 2018.

[21] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi. *A general framework for information extraction using dynamic span graphs*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3036–3046, 2019.

[22] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. *Overview of the TAC 2010 knowledge base population track*. In Proceedings of the 2010 Text Analysis Conference, pages 3–3, 2010.

[23] J. Ellis, J. Getman, and S. M. Strassel. *Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results*. In Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology, pages 17–18, 2014.

[24] H. Ji, J. Nothman, B. Hachey, and R. Florian. *Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking.* In Proceedings of the 2015 Text Analysis Conference, 2015.

[25] J. Ellis, J. Getman, D. Fore, N. Kuster, Z. Song, A. Bies, and S. M. Strassel. *Overview of Linguistic Resources for the TAC KBP 2015 Evaluations: Methodologies and Results.* In Proceedings of the 2015 Text Analysis Conference, 2015.

[26] H. Ji, X. Pan, B. Zhang, J. Nothman, J. Mayfield, P. McNamee, C. Costello, and S. I. Hub. *Overview of TAC-KBP2017 13 Languages Entity Discovery and Linking.* In Proceedings of the 2017 Text Analysis Conference, 2017.

[27] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel. *The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation*. In Proceedings of the 2004 International Conference on Language Resources and Evaluation Workshop on Linguistics, pages 837–840, 2004.

[28] C. Walker, S. Strassel, J. Medero, and K. Maeda. *ACE 2005 multilingual training corpus*. Linguistic Data Consortium, Philadelphia, 57, 2006.

[29] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma. *From light to rich ere: annotation of entities, relations, and events*. In Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 89–98, 2015.

[30] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum. *SemEval 2017 Task 10: ScienceIE-Extracting Keyphrases and Relations from Scientific Publications*. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 546–555, 2017.

[31] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning. *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, 2017.

[32] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. *SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals*. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 33–38, 2010.

[33] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun. *FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4803–4809, 2018.

[34] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. *DocRED: A Large-Scale Document-Level Relation Extraction Dataset*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 764–777, 2019.

[35] S. Riedel, L. Yao, and A. McCallum. *Modeling relations and their mentions without labeled text*. In Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases, pages 148–163, 2010.

[36] C. Quirk and H. Poon. *Distant Supervision for Relation Extraction beyond the Sentence Boundary*. In Proceedings of the 2017 Conference of the European Chapter of the Association for Computational Linguistics, pages 1171–1182, 2017.

[37] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-t. Yih. *Cross-sentence n-ary relation extraction with graph lstms*. Transactions of the Association for Computational Linguistics, 5:101–115, 2017.

[38] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi. *Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3219–3232, 2018.

[39] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. *Gated graph sequence neural networks*. In Proceedings of the 2016 International Conference on Learning Representations, 2016.

[40] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. *How Powerful are Graph Neural Networks?* In Proceedings of the 2018 International Conference on Learning Representations, 2018.

[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. *A comprehensive survey on graph neural networks*. IEEE Transactions on Neural Networks and Learning Systems, pages 1–21, 2020.

[42] B. Kantor and A. Globerson. *Coreference resolution with entity equalization*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 673–677, 2019.

[43] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi. *Entity, Relation, and Event Extraction with Contextualized Span Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pages 5788–5793, 2019.

[44] T.-J. Fu, P.-H. Li, and W.-Y. Ma. *GraphRel: Modeling text as relational graphs for joint entity and relation extraction*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 1409–1418, 2019.

[45] E. F. T. K. Sang and F. De Meulder. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 142–147, 2003.

[46] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham. *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, 2017.

[47] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. *Collective annotation of Wikipedia entities in web text*. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 457–466, 2009.

[48] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. *Robust disambiguation of named entities in text*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 782–792, 2011.

[49] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy. *Named Entity Disambiguation for Noisy Text*. In Proceedings of the 2017 Conference on Computational Natural Language Learning, pages 58–68, 2017.

[50] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. *CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes*. In Proceedings of the 2012 Conference on Computational Natural Language Learning, pages 1–40, 2012.

[51] K. Webster, M. Recasens, V. Axelrod, and J. Baldridge. *Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns*. Transactions of the Association for Computational Linguistics, 6:605–617, 2018.

[52] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu. *BioCreative V CDR task corpus: a resource for chemical disease relation extraction*. Database, 2016, 2016.

[53] C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu. *Overview of the BioCreative V chemical disease relation (CDR) task*. In Proceedings of the 5th BioCreative Challenge Evaluation Workshop, 2015.

[54] N. Chinchor and E. Marsh. *Muc-7 information extraction task definition*. In Proceeding of the 1998 Message Understanding Conference (MUC-7), pages 359–367, 1998.

[55] J. Aguilar, C. Beller, P. McNamee, B. Van Durme, S. Strassel, Z. Song, and J. Ellis. *A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards*. In Proceedings of the 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 45–53, 2014.

[56] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. *OntoNotes: the 90% solution*. In Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 57–60, 2006.

[57] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al. *Ontonotes release 5.0 ldc2013t19*. Linguistic Data Consortium, Philadelphia, PA, 23, 2013.

[58] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. *GENIA corpus - a semantically annotated corpus for bio-textmining*. Bioinformatics, 19(suppl_1):180–182, 2003.

[59] L. Bentivogli, P. Forner, C. Giuliano, A. Marchetti, E. Pianta, and K. Tymoshenko. *Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia*. In Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, pages 19–27, 2010.

[60] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. *Neural Architectures for Named Entity Recognition*. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 260–270, 2016.

[61] J. P. Chiu and E. Nichols. *Named Entity Recognition with Bidirectional LSTM-CNNs*. Transactions of the Association for Computational Linguistics, 4:357–370, 2016.

[62] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli. *Cloze-driven Pretraining of Self-attention Networks*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pages 5363–5372, 2019.

[63] A. Akbik, T. Bergmann, and R. Vollgraf. *Pooled contextualized embeddings for named entity recognition*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 724–728, 2019.

[64] A. Akbik, D. Blythe, and R. Vollgraf. *Contextual string embeddings for sequence labeling*. In Proceedings of the 2018 International Conference on Computational Linguistics, pages 1638–1649, 2018.

[65] K. Clark, M.-T. Luong, C. D. Manning, and Q. Le. *Semi-Supervised Sequence Modeling with Cross-View Training*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1914–1925, 2018.

[66] E. Strubell, P. Verga, D. Belanger, and A. McCallum. *Fast and Accurate Entity Recognition with Iterated Dilated Convolutions*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2670–2680, 2017.

[67] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *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, 2018.

[68] Q. Li and H. Ji. *Incremental joint extraction of entity mentions and relations*. In Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics, pages 402–412, 2014.

[69] M. Zhang, Y. Zhang, and G. Fu. *End-to-end neural relation extraction with global optimization*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1730–1740, 2017.

[70] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. *Matching the Blanks: Distributional Similarity for Relation Learning*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 2895–2905, 2019.

[71] Y. Zhang, P. Qi, and C. D. Manning. *Graph Convolution over Pruned Dependency Trees Improves Relation Extraction*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2205–2215, 2018.

[72] W. Hu, B. Ma, Z. Li, Y. Li, and Y. Wang. *A Cross-Media Deep Relationship Classification Method Using Discrimination Information*. Information Processing & Management, 57(6):102344, 2020.

[73] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. *Knowledge Enhanced Contextual Word Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, pages 43–54, 2019.

[74] Z. Guo, Y. Zhang, and W. Lu. *Attention Guided Graph Convolutional Networks for Relation Extraction*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 241–251, 2019.

[75] M. L. McHugh. *Interrater reliability: the kappa statistic*. Biochemia medica: Biochemia medica, 22(3):276–282, 2012.

[76] J. Landis and G. Koch. *The measurement of observer agreement for categorical data.* Biometrics, 33(1):159–174, 1977.

[77] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. *The graph neural network model*. IEEE Transactions on Neural Networks, 20(1):61–80, 2008.

[78] K. Dixit and Y. Al-Onaizan. *Span-level model for relation extraction*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics, pages 5308–5314, 2019.

[79] X. Ma and E. Hovy. *End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF*. In Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics, pages 1064–1074, 2016.

[80] Y. Luan, M. Ostendorf, and H. Hajishirzi. *Scientific Information Extraction with Semi-supervised Neural Tagging*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2641–2651, 2017.

[81] A. Katiyar and C. Cardie. *Nested named entity recognition revisited*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 861–871, 2018.

[82] J. Pennington, R. Socher, and C. D. Manning. *Glove: Global vectors for word representation*. In Proceedings of the 2014 conference on empirical methods in natural language processing, pages 1532–1543, 2014.

[83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, 2019.

[84] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube. *Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation*. In Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics, pages 30–35, 2014.

[85] A. Bagga and B. Baldwin. *Algorithms for scoring coreference chains*. In Proceedings of the 1998 International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, pages 563–566, 1998.

[86] X. Luo. *On coreference resolution performance metrics*. In Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 25–32, 2005.

[87] G. Durrett and D. Klein. *Easy victories and uphill battles in coreference resolution*. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1971–1982, 2013.

[88] S. Wiseman, A. M. Rush, S. M. Shieber, and J. Weston. *Learning Anaphoricity and Antecedent Ranking Features for Coreference Resolution*. In Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing, pages 1416–1426, 2015.

[89] C. Kulkarni, W. Xu, A. Ritter, and R. Machiraju. *An Annotated Corpus for Machine Reading of Instructions in Wet Lab Protocols*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 97–106, 2018.

[90] R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. *OntoNotes: A large training corpus for enhanced processing*. Handbook of Natural Language Processing and Machine Translation. Springer, page 59, 2011.

[91] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. *Reconstructing the house from the ad: Structured prediction on real estate classifieds*. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 274–279, 2017.

[92] X. Han and L. Wang. *A Novel Document-Level Relation Extraction Method Based on BERT and Entity Information*. IEEE Access, 2020.

[93] S. Wu and Y. He. *Enriching pre-trained language model with entity information for relation classification*. In Proceedings of the 2019 ACM International Conference on Information and Knowledge Management, pages 2361–2364, 2019.

[94] J. Cohen. *A coefficient of agreement for nominal scales*. Educational and psychological measurement, 20(1):37–46, 1960.

[95] W. A. Scott. *Reliability of content analysis: The case of nominal scale coding*. Public opinion quarterly, pages 321–325, 1955.

3

Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution

In this chapter, we present a more detailed model on how the entity-centric approach can be used for entity linking task. The entity linking consists in mapping the anchor mentions in text to target entities that describe them in a Knowledge Base (KB) (e.g., Wikipedia). In our work, we showcase that this task can be improved by considering performing entity linking on the coreference cluster level instead of on each of the mentions individually. By adopting this approach, our joint model is able to use the information of all the coreferent mentions at once when choosing the candidate entity. As a result, this leads to more consistent predictions among mentions referring to the same concept, especially boosting the performance on corner cases consisting of unpopular mentions.

⋆⋆⋆

K. Zaporojets, J. Deleu, Y. Yiang, T. Demeester and C. Develder

In Proceedings of the ACL 2022

Abstract We consider the task of document-level entity linking (EL), where it is important to make consistent decisions for entity mentions over the full document

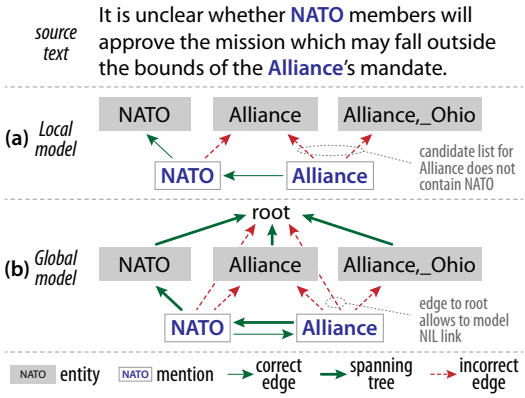


Figure 3.1: Illustration of our 2 explored graph models: (a) Local where edges are only allowed from spans to antecedents or candidate entities, and (b) Global where the prediction involves a spanning tree over all nodes.

jointly. We aim to leverage explicit "connections" among mentions within the document itself: we propose to join EL and coreference resolution (coref) in a *single* structured prediction task over directed trees and use a globally normalized model to solve it. This contrasts with related works where two separate models are trained for each of the tasks and additional logic is required to merge the outputs. Experimental results on two datasets show a boost of up to +5% F1-score on both coref and EL tasks, compared to their standalone counterparts. For a subset of hard cases, with individual mentions lacking the correct EL in their candidate entity list, we obtain a +50% increase in accuracy.[1]

3.1 Introduction

In this paper we explore a principled approach to solve entity linking (EL) jointly with coreference resolution (coref). Concretely, we formulate coref+EL as a *single* structured task over directed trees that conceives EL and coref as two complementary components: a coreferenced cluster can only be linked to a single entity or NIL (i.e., a non-linkable entity), and all mentions linking to the same entity are coreferent. This contrasts with previous attempts to join coref+EL [1–3] where coref and EL models are trained separately and additional logic is required to merge the predictions of both tasks.

Our first approach (Local in Fig. 3.1(a)) is motivated by current state-of-the-art coreference resolution models [4, 5] that predict a single antecedent for each span to resolve. We extend this architecture by also considering entity links as potential antecendents: in the example of Fig. 3.1, the mention "Alliance" can be either connected to its antecedent mention "NATO" or to any of its candidate links (*Alliance* or *Alliance,_Ohio*). While straightforward, this approach cannot solve cases

[1]Our code, models and AIDA$^+$ dataset will be released on https://github.com/klimzaporojets/consistent-EL

where the first coreferenced mention does not include the correct entity in its candidate list (e.g., if the order of "NATO" and "Alliance" mentions in Fig. 3.1 would be reversed). We therefor propose a second approach, Global, which by construction overcomes this inherent limitation by using bidirectional connections between mentions. Because that implies cycles could be formed, we resort to solving a maximum spanning tree problem. Mentions that refer to the same entity form a cluster, represented as a subtree rooted by the single entity they link to. To encode the overall document's clusters in a single spanning tree, we introduce a virtual *root* node (see Fig. 3.1(b)).[2]

This paper contributes: (i) 2 architectures (Local and Global) for joint entity linking (EL) and corefence resolution, (ii) an extended AIDA dataset [6], adding new annotations of linked and NIL coreference clusters, (iii) experimental analysis on 2 datasets where our joint coref+EL models achieve up to +5% F1-score on both tasks compared to standalone models. We also show up to +50% in accuracy for hard cases of EL where entity mentions lack the correct entity in their candidate list.

3.2 Architecture

Our model takes as input (i) the full document text, and (ii) an *alias table* with entity candidates for each of the possible spans. Our end-to-end approach allows to jointly predict the mentions, entity links and coreference relations between them.

3.2.1 Span and entity representations

We use SpanBERT (base) from [7] to obtain *span* representations $\mathbf{g}_i$ for a particular span s_i. Similarly to [8, 9], we apply an additional pruning step to keep only the top-N spans based on the pruning score Φ_{p} from a feed-forward neural net (FFNN):

$$\Phi_{\text{p}}(s_i) = \text{FFNN}_P(\mathbf{g}_i). \tag{3.1}$$

For a candidate entity e_j of span s_i we will obtain representation as $\mathbf{e}_j$ (which is further detailed in Section 3.3).

3.2.2 Joint approaches

We propose two methods for joint coreference and EL. The first, Local, is motivated by end-to-end span-based coreference resolution models [10, 11] that optimize the marginalized probability of the correct antecedents for each given span. We extend this local marginalization to include the span's candidate entity links. Formally, the modeled probability of y (text span or candidate entity) being the antecedent of span s_i is:

$$P_{\text{cl}}(y|s_i) = \frac{\exp\left(\Phi_{\text{cl}}(s_i, y)\right)}{\sum_{y' \in \mathcal{Y}(s_i)} \exp\left(\Phi_{\text{cl}}(s_i, y')\right)}, \tag{3.2}$$

[2]Coreference clusters without a linked entity, i.e., a NIL cluster, have a link of a mention directly to the root.

where $\mathcal{Y}(s_i)$ is the set of antecedent spans unified with the candidate entities for s_i. For antecedent *spans* $\{s_j : j < i\}$ the score Φ_{cl} is defined as:

$$\Phi_{\text{cl}}(s_i, s_j) = \Phi_{\text{p}}(s_i) + \Phi_{\text{p}}(s_j) + \Phi_{\text{c}}(s_i, s_j), \tag{3.3}$$

$$\Phi_{\text{c}}(s_i, s_j) = \text{FFNN}_{\text{C}}([\mathbf{g}_i; \mathbf{g}_j; \mathbf{g}_i \odot \mathbf{g}_j; \boldsymbol{\varphi}_{i,j}]), \tag{3.4}$$

where $\boldsymbol{\varphi}_{i,j}$ is an embedding encoding the distance[3] between spans s_i and s_j. Similarly, for a particular candidate *entity* e_j, the score Φ_{cl} is:

$$\Phi_{\text{cl}}(s_i, e_j) = \Phi_{\text{p}}(s_i) + \Phi_{\ell}(s_i, e_j), \tag{3.5}$$

$$\Phi_{\ell}(s_i, e_j) = \text{FFNN}_{L}([\mathbf{g}_i; \mathbf{e}_j]). \tag{3.6}$$

An example graph of mentions and entities with edges for which aforementioned scores Φ_{cl} would be calculated is sketched in Fig. 3.1(a). While simple, this approach fails to correctly solve EL when the correct entity is only present in the candidate lists of mention spans occurring later in the text (since earlier mentions have no access to it).

To solve EL in the general case, even when the first mention does not have the correct entity, we propose bidirectional connections between mentions, thus leading to a maximum spanning tree problem in our Global approach. Here we define a score for a (sub)tree t, noted as $\Phi_{\text{tr}}(t)$:

$$\Phi_{\text{tr}}(t) = \sum_{(i,j)\in t} \Phi_{\text{cl}}(u_i, u_j), \tag{3.7}$$

where u_i and u_j are two connected nodes (i.e., *root*, candidate entities or spans) in t. For a ground truth cluster $c \in C$ (with C being the set of all such clusters), with its set[4] of correct subtree representations $\mathcal{T}_c$, we model the cluster's likelihood with its subtree scores. We minimize the negative log-likelihood $\mathcal{L}$ of all clusters:

$$\mathcal{L} = -\log \frac{\prod_{c\in C} \sum_{t\in\mathcal{T}_c} \exp\left(\Phi_{\text{tr}}(t)\right)}{\sum_{t\in\mathcal{T}_{all}} \exp\left(\Phi_{\text{tr}}(t)\right)}. \tag{3.8}$$

Naively enumerating all possible spanning trees ($\mathcal{T}_{all}$ or $\mathcal{T}_c$) implied by this equation is infeasible, since their number is exponentially large. We use the adapted Kirchhoff's Matrix Tree Theorem (MTT; [12, 13]) to solve this: the sum of the weights of the spanning trees in a directed graph rooted in r is equal to the determinant of the Laplacian matrix of the graph with the row and column corresponding to r removed (i.e., the *minor* of the Laplacian with respect to r). This way, Eq. (3.8) can be rewritten as

$$\mathcal{L} = -\log \frac{\prod_{c\in C} \det\left(\hat{\mathbf{L}}_c\left(\boldsymbol{\Phi}_{\text{cl}}\right)\right)}{\det\left(\mathbf{L}_r\left(\boldsymbol{\Phi}_{\text{cl}}\right)\right)}, \tag{3.9}$$

where $\boldsymbol{\Phi}_{\text{cl}}$ is the weighted adjacency matrix of the graph, and $\mathbf{L}_r$ is the minor of

[3] Measured in number of spans, after pruning.

[4] For a single cluster annotation, indeed it is possible that multiple correct trees can be drawn.

Dataset	# Linked clusters	# NIL clusters	Linked mentions	# NIL mentions
DWIE	11,967	9,935	28,482	14,891
AIDA	16,673	-	27,817	7,112
AIDA$^+$	16,775	4,284	28,813	6,116

Table 3.1: Datasets statistics.

the Laplacian with respect to the root node r. An entry in the Laplacian matrix is calculated as

$$L_{i,j} = \begin{cases} \sum_k \exp(\Phi_{\text{cl}}(u_k, u_j)) & \text{if } i = j \\ -\exp(\Phi_{\text{cl}}(u_i, u_j)) & \text{otherwise} \end{cases}, \tag{3.10}$$

Similarly, $\hat{\mathbf{L}}_c$ is a *modified Laplacian* matrix where the first row is replaced with the root r selection scores $\Phi_{\text{cl}}(r, u_j)$. For clarity, Appendix 3.6 presents a toy example with detailed steps to calculate the loss in Eq. (3.9).

To calculate the scores of each of the entries $\Phi_{\text{cl}}(u_i, u_j)$ to $\mathbf{\Phi}_{\text{cl}}$ matrix in eqs. (3.7) and (3.9) for Global, we use the same approach as in Local for edges between two mention spans, or between a mention and entity. For the directed edges between the root r and a candidate entity e_j we choose $\Phi_{\text{cl}}(r, e_j) = 0$. Since we represent NIL clusters by edges from the mention spans directly to the root, we also need scores for them: we use Eq. (3.3) with $\Phi_{\text{p}}(r) = 0$. We use Edmonds' algorithm [14] for decoding the maximum spanning tree.

3.3 Experimental setup

We considered two datasets to evaluate our proposed models: DWIE [15] and AIDA [6]. Since AIDA essentially does not contain coreference information, we had to extend it by (i) adding missing mention links in order to make annotations consistent on the coreference cluster level, and (ii) annotating NIL coreference clusters. We note this extended dataset as AIDA$^+$. See Table 3.1 for the details.

As input to our models, for DWIE we generate spans of up to 5 tokens. For each mention span s_i, we find candidates from a dictionary of entity surface forms used for hyperlinks in Wikipedia. We then keep the top-16 candidates based on the prior for that surface form, as per [16, §3]. Each of those candidates e_j is represented using a Wikipedia2Vec embedding $\mathbf{e}_j$ [16].[5] For AIDA$^+$, we use the spans, entity candidates, and entity representations from [17].[6]

To assess the performance of our joint coref+EL models Local and Global, we also provide Standalone implementations for coref and EL tasks. The Standalone coref model is trained using only the coreference component of our joint architecture (Eq. (3.2)–(3.4)), while the EL model is based only on the linking component (Eq. (3.6)).

[5] We use Wikipedia version 20200701.

[6] https://github.com/dalab/end2end_neural_el

Setup	DWIE			$AIDA_a^+$			$AIDA_b^+$		
	EL_m	EL_h	Coref	EL_m	EL_h	Coref	EL_m	EL_h	Coref
Standalone	88.7±0.1	78.4±0.2	94.5±0.1	86.2±0.4	80.7±0.5	93.8±0.1	79.1±0.3	74.0±0.3	91.5±0.3
Local	90.5±0.4	83.4±0.4	94.4±0.2	87.5±0.2	83.1±0.2	94.7±0.1	**79.9±0.4**	75.8±0.3	**92.3±0.1**
Global	**90.7±0.3**	**83.9**±0.5	**94.7**±0.2	**87.6±0.2**	**83.7**±0.3	**95.1**±0.1	79.6±0.4	**76.0**±0.4	92.2±0.2

Table 3.2: Experimental results (F1 scores defined in Section 3.3) using the Standalone coreference and EL models compared to our joint architectures (Local and Global), on DWIE and $AIDA^+$ datasets.

As performance metrics, for coreference resolution we calculate the average-F1 score of commonly used MUC [18], B^3 [19] and $CEAF_e$ [20] metrics as implemented by [21]. For EL, we use (i) *mention*-level F1 score (EL_m), and (ii) *cluster*-level *hard* F1 score (EL_h) that counts a true positive only if both the coreference cluster (in terms of all its mention spans) and the entity link are correctly predicted. These EL metrics are executed in a *strong matching* setting that requires predicted spans to exactly match the boundaries of gold mentions. Furthermore, for EL we only report the performance on non-NIL mentions, leaving the study of NIL links for future work.

Our experiments will answer the following research questions: **(Q1)** How does performance of our joint coref+EL models compare to Standalone models? **(Q2)** Does jointly solving coreference resolution and EL enable more coherent EL predictions? **(Q3)** How do our joint models perform on hard cases where some individual entity mentions do not have the correct candidate?

3.4 Results

Table 3.2 shows the results of our compared models for EL and coreference resolution tasks. Answering **(Q1)**, we observe a general improvement in performance of our coref+EL joint models (Local and Global) compared to Standalone on the EL task. Furthermore, this difference is bigger when using our cluster-level *hard* metrics. This also answers **(Q2)** by indicating that the joint models tend to produce more coherent cluster-based predictions. To make this more explicit, Table 3.3 compares the accuracy for singleton clusters (i.e., clusters composed by a single entity mention), denoted as *S*, to that of clusters composed by multiple mentions, denoted as *M*. We observe that the difference in performance between our joint models and Standalone is bigger on *M* clusters (with a consistent superiority of Global), indicating that our approach indeed produces more coherent predictions for mentions that refer to the same concept. Further analysis reveals that this difference in performance is even higher for a more complex scenario where the clusters contain mentions with different surface forms (not shown in the table).

In order to tackle research question **(Q3)**, we study the accuracy of our models on the important corner case that involves mentions without correct entity in their candidate lists. This is illustrated in Table 3.4, which focuses on such mentions in clusters where at least one mention contains the correct entity in its candidate list. As expected, the Standalone model cannot link such mentions, as it is limited to the local candidate list. In contrast, both our joint approaches can solve some of

	DWIE		$AIDA_a^+$		$AIDA_b^+$	
Setup	*S*	*M*	*S*	*M*	*S*	*M*
Standalone	80.4	69.5	82.9	70.7	77.0	57.0
Local	**82.6**	78.6	84.9	74.8	**79.8**	61.4
Global	**82.6**	**80.0**	**85.1**	**76.8**	79.3	**63.0**

Table 3.3: Cluster-based accuracy of link prediction on singletons (*S*) and clusters of multiple mentions (*M*).

Setup	DWIE	$AIDA_a^+$	$AIDA_b^+$
Standalone	0.0	0.0	0.0
Local	41.7	27.4	26.9
Global	**57.6**	**50.2**	**29.7**

Table 3.4: EL accuracy for corner case mentions where the correct entity is not in the mention's candidate list.

these cases by using the correct candidates from other mentions in the cluster, with a superior performance of our Global model compared to the Local one.

3.5 Related work

Entity Linking: Related work in entity linking (EL) tackles the document-level linking coherence by exploring relations between entities [17, 22, 23], or entities and mentions [24]. More recently, contextual BERT-driven [25] language models have been used for the EL task [26–29] by jointly embedding mentions and entities. In contrast, we explore a cluster-based EL approach where the coherence is achieved on *coreferent* entity mentions level.

Coreference Resolution: Span-based antecedent-ranking coreference resolution [10, 11] has seen a recent boost by using SpanBERT representations [5, 7, 9]. We extend this approach in our Local joint coref+EL architecture. Furthermore, we rely on Kirchhoff's Matrix Tree Theorem [12, 13] to efficiently train a more expressive spanning tree-based Global method.

Joint EL+Coref: [30] introduce a more expensive rule-based Integer Linear Programming component to jointly predict coref and EL. [31] jointly train coreference and entity linking without enforcing single-entity per cluster consistency. More recently, [3, 32] use additional logic to achieve consistent cluster-level entity linking. In contrast, our proposed approach constrains the space of the predicted spanning trees on a structural level (see Fig. 3.1).

3.6 Conclusion

We propose two end-to-end models to solve entity linking and coreference resolution tasks in a joint setting. Our joint architectures achieve superior performance

compared to the standalone counterparts. Further analysis reveals that this boost in performance is driven by more coherent predictions on the level of mention clusters (linking to the same entity) and extended candidate entity coverage.

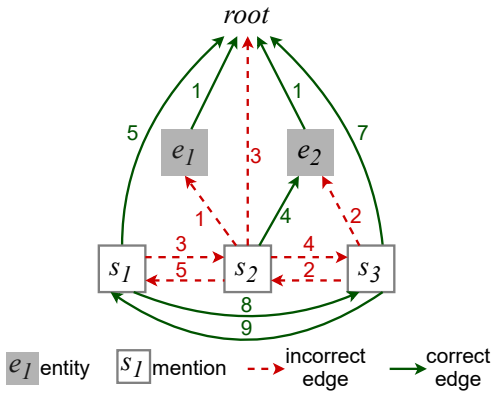


Figure 3.2: Illustrative graph example of Global model. The weights of the edges correspond to $\exp(\mathbf{\Phi}_{\text{cl}})$ (see Eq. (3.11)).

Appendix

In this appendix we will provide a clarifying artificial example in order to walk the reader step by step through MTT (Eq. (3.9)–(3.10)) applied in our Global approach. The graph of the example is illustrated in Fig. 3.2 and is composed by nodes representing *root* (r), entities e_1 and e_2, and spans s_1, s_2 and s_3. The span s_2 is associated with candidate entity set $\{e_1, e_2\}$ (i.e., represented by edges from s_2 to e_1 and e_2), and s_3 with $\{e_2\}$ (i.e., represented by the edge from s_3 to e_2). The candidate entity set of s_1 is empty. The nodes are grouped in two ground truth clusters: NIL cluster $c_1 = \{s_1, s_2\}$, and linked cluster $c_2 = \{e_2, s_2\}$.

The exponential of weighted adjacency matrix[7] $\mathbf{\Phi}_{\text{cl}}$ of the presented example is:

$$\exp(\mathbf{\Phi}_{\text{cl}}) = \begin{array}{c} \\ r \\ e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccccc} r & e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 5 & 3 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 5 & 9 \\ 0 & 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 8 & 4 & 0 \end{bmatrix} \end{array}, \tag{3.11}$$

where the weights of incorrect edges are represented in red (i.e., red dashed edges in Fig. 3.2), the weights of the correct edges in green (i.e., green edges in Fig. 3.2), and the weights between disconnected nodes are set to 0.

In order to compute the *denominator* of the loss function in Eq. (3.9), the Laplacian of the matrix in Eq. (3.11) is calculated as described in Eq. (3.10), and the row and column corresponding to root r removed (i.e., the *minor* $\mathbf{L}_r$ with respect to the

[7] For simplicity, the weights are small integers.

root):

$$\mathbf{L}_r = \begin{array}{c} \\ e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -4 & -2 \\ 0 & 0 & 16 & -5 & -9 \\ 0 & 0 & -3 & 17 & -2 \\ 0 & 0 & -8 & -4 & 20 \end{bmatrix} \end{array}. \tag{3.12}$$

Following Kirchhoff's Matrix Tree Theorem [12, 13], the determinant of $\mathbf{L}_r$ equals to the sum of the weights of all possible spanning trees of the graph represented in Fig. 3.2:

$$\det(\mathbf{L}_r) = 3600 = \sum_{t \in \mathcal{T}_{all}} \exp\left(\Phi_{\text{tr}}(t)\right). \tag{3.13}$$

In order to compute the *numerator* of the loss function in Eq. (3.9) (i.e., the sum of the weights of the spanning trees of ground truth clusters), we first mask out (set to zero) all the weights assigned to incorrect edges:

$$\exp(\mathbf{\Phi}_{\text{cl}})' = \begin{array}{c} \\ r \\ e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccccc} r & e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 5 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 \end{bmatrix} \end{array} \tag{3.14}$$

Next, the *modified Laplacian* (i.e., Laplacian with the first row replaced by root r selection weights) $\hat{\mathbf{L}}$ is calculated for both clusters c_1 and c_2:

$$\hat{\mathbf{L}}_{c_1} = \begin{array}{c} \\ r \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_3 \end{array} \\ \begin{bmatrix} 5 & 7 \\ -8 & 9 \end{bmatrix} \end{array} \tag{3.15}$$

$$\hat{\mathbf{L}}_{c_2} = \begin{array}{c} \\ r \\ s_2 \end{array} \begin{array}{c} \begin{array}{cc} e_2 & s_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \end{array} \tag{3.16}$$

The determinants of $\hat{\mathbf{L}}_{c_1}$ and $\hat{\mathbf{L}}_{c_2}$ equal to the sum of the weights of all spanning trees connecting the nodes in clusters c_1 and c_2 respectively:

$$\det(\hat{\mathbf{L}}_{c_1}) = 101 = \sum_{t \in \mathcal{T}_{c_1}} \exp\left(\Phi_{\text{tr}}(t)\right) \tag{3.17}$$

$$\det(\hat{\mathbf{L}}_{c_2}) = 4 = \sum_{t \in \mathcal{T}_{c_2}} \exp\left(\Phi_{\text{tr}}(t)\right) \tag{3.18}$$

Finally, in order to calculate the final loss, we replace the obtained results in eqs. (3.13), (3.17), and (3.18) in the loss function of Eq. (3.9):

$$\mathcal{L} = -\log \frac{101 * 4}{3600}. \tag{3.19}$$

Note: strictly speaking, there are *three* clusters rooted in *root* in the graph of Fig. 3.2, the third one being $c_3 = \{e_1\}$, whose exponential weight is 1 by definition of

$\Phi_{\text{cl}}(r, e_j) = 0$ (see Section 3.2.2), and has no impact in calculation of the loss function in Eq. (3.19).

References

[1] H. Hajishirzi, L. Zilles, D. S. Weld, and L. Zettlemoyer. *Joint coreference resolution and named-entity linking with multi-pass sieves*. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), pages 289–299, 2013. Available from: https://aclanthology.org/D13-1029/ .

[2] S. Dutta and G. Weikum. *C3EL: A joint model for cross-document co-reference resolution and entity linking*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pages 846–856, 2015. Available from: https://doi.org/10.18653/v1/d15-1101.

[3] R. Angell, N. Monath, S. Mohan, N. Yadav, and A. McCallum. *Clustering-based Inference for Biomedical Entity Linking*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 2598–2608, 2021. Available from: https://doi.org/10.18653/v1/2021.naacl-main.205.

[4] M. Joshi, O. Levy, L. Zettlemoyer, and D. S. Weld. *BERT for Coreference Resolution: Baselines and Analysis*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 5807–5812, 2019. Available from: https://www.aclweb.org/anthology/D19-1588.

[5] W. Wu, F. Wang, A. Yuan, F. Wu, and J. Li. *CorefQA: Coreference Resolution as Query-based Span Prediction*. In Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics (ACL 2020), pages 6953–6963, 2020. Available from: https://www.aclweb.org/anthology/2020.acl-main.622.

[6] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. *Robust disambiguation of named entities in text*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pages 782–792, 2011. Available from: https://www.aclweb.org/anthology/D11-1072/.

[7] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. *SpanBERT: Improving pre-training by representing and predicting spans*. Transactions of the Association for Computational Linguistics (TACL 2020), 8:64–77, 2020. Available from: https://www.aclweb.org/anthology/2020.tacl-1.5.

[8] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi. *A general framework for information extraction using dynamic span graphs*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 3036–3046, 2019. Available from: https://www.aclweb.org/anthology/N19-1308/.

[9] L. Xu and J. D. Choi. *Revealing the Myth of Higher-Order Inference in Coreference Resolution*. In Proceedings of the 2020 Conference on Empirical Methods in

Natural Language Processing (EMNLP 2020), pages 8527–8533, 2020. Available from: https://doi.org/10.18653/v1/2020.emnlp-main.686.

[10] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. *End-to-end Neural Coreference Resolution*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 188–197, 2017. Available from: https://www.aclweb.org/anthology/D17-1018.

[11] K. Lee, L. He, and L. Zettlemoyer. *Higher-Order Coreference Resolution with Coarse-to-Fine Inference*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), pages 687–692, 2018. Available from: https://www.aclweb.org/anthology/N18-2108.

[12] T. Koo, A. Globerson, X. Carreras, and M. Collins. *Structured prediction models via the matrix-tree theorem*. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), pages 141–150, 2007. Available from: https://www.aclweb.org/anthology/D07-1015/.

[13] W. Tutte. *Graph Theory*. Encyclopedia of Mathematics and its Applications, 21, 1984. Available from: https://doi.org/10.1002/net.3230160110.

[14] J. Edmonds. *Optimum branchings*. Journal of Research of the National Bureau of Standards B, 71(4):233–240, 1967. Available from: https://nvlpubs.nist.gov/nistpubs/jres/71b/jresv71bn4p233_a1b.pdf.

[15] K. Zaporojets, J. Deleu, C. Develder, and T. Demeester. *DWIE: An entity-centric dataset for multi-task document-level information extraction*. Information Processing & Management, 58(4):102563, 2021. Available from: https://arxiv.org/abs/2009.12626.

[16] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. *Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation*. In Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016), pages 250–259, 2016. Available from: https://doi.org/10.18653/v1/k16-1025.

[17] N. Kolitsas, O.-E. Ganea, and T. Hofmann. *End-to-End Neural Entity Linking*. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 519–529, 2018. Available from: https://www.aclweb.org/anthology/K18-1050/.

[18] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. *A model-theoretic coreference scoring scheme*. In Proceedings of the 1995 Conference on Message understanding (MUC6, 1995), pages 45–52, 1995. Available from: https://doi.org/10.3115/1072399.1072405.

[19] A. Bagga and B. Baldwin. *Algorithms for scoring coreference chains*. In Proceedings of the 1998 International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference (LREC 1998), pages 563–566, 1998. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.5848&rep=rep1&type=pdf.

[20] X. Luo. *On coreference resolution performance metrics*. In Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP

2005), pages 25–32, 2005. Available from: https://www.aclweb.org/anthology/H05-1004/.

[21] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube. *Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation*. In Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics (ACL 2014), pages 30–35, 2014. Available from: https://doi.org/10.3115/v1/p14-2006.

[22] X. Yang, X. Gu, S. Lin, S. Tang, Y. Zhuang, F. Wu, Z. Chen, G. Hu, and X. Ren. *Learning Dynamic Context Augmentation for Global Entity Linking*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 271–281, 2019. Available from: https://doi.org/10.18653/v1/D19-1026.

[23] P. Le and I. Titov. *Boosting Entity Linking Performance by Leveraging Unlabeled Documents*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019), pages 1935–1945, 2019. Available from: https://www.aclweb.org/anthology/P19-1187.

[24] P. Le and I. Titov. *Improving Entity Linking by Modeling Latent Relations between Mentions*. In Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics (ACL 2018), pages 1595–1604, 2018. Available from: https://www.aclweb.org/anthology/P18-1148.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 4171–4186, 2019. Available from: https://www.aclweb.org/anthology/N19-1423.

[26] S. Broscheit. *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 2019), pages 677–685, 2019. Available from: https://www.aclweb.org/anthology/K19-1063.

[27] N. De Cao, G. Izacard, S. Riedel, and F. Petroni. *Autoregressive Entity Retrieval*. In Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021), 2020. Available from: https://arxiv.org/abs/2010.00904.

[28] N. De Cao, W. Aziz, and I. Titov. *Highly Parallel Autoregressive Entity Linking with Discriminative Correction*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7662–7669, 2021. Available from: https://aclanthology.org/2021.emnlp-main.604, doi:10.18653/v1/2021.emnlp-main.604.

[29] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto. *Global Entity Disambiguation with Pretrained Contextualized Embeddings of Words and Entities*. arXiv preprint arXiv:1909.00426, 2020. Available from: https://arxiv.org/abs/1909.00426.

[30] A. Fahrni and M. Strube. *Jointly disambiguating and clustering concepts and entities with Markov logic*. In Proceedings of the 2012 International Conference on

Computational Linguistics (COLING 2012), pages 815–832, 2012. Available from: https://aclanthology.org/C12-1050/.

[31] G. Durrett and D. Klein. *A joint model for entity analysis: Coreference, typing, and linking*. Transactions of the Association for Computational Linguistics (TACL 2014), 2:477–490, 2014. Available from: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/412.

[32] D. Agarwal, R. Angell, N. Monath, and A. McCallum. *Entity Linking and Discovery via Arborescence-based Supervised Clustering*. arXiv preprint arXiv:2109.01242, 2021. Available from: https://arxiv.org/abs/2109.01242.

4

Injecting Knowledge Base Information into End-to-End Joint Entity and Relation Extraction and Coreference Resolution

In this chapter we adopt a slightly different approach involving entities: instead of using purely textual information to solve information extraction tasks such as relation extraction, we also study how the information of entities from Knowledge Base can be integrated. We achieve significant improvement on all the evaluated tasks by injecting information both from Wikipedia, as well as from Wikidata KBs. Furthermore, while the tasks we are tackling are annotated and defined on named entity level, the information we inject in our text comes from all the existing entities defined in the experimented KBs. We find that this unsupervised technique is still able to detect the entities that are more relevant for a particular text.

⋆⋆⋆

S. Verlinden*, K. Zaporojets*, J. Deleu, T. Demeester and C. Develder

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

*Equal contribution

Abstract We consider a joint information extraction (IE) model, solving named entity recognition, coreference resolution and relation extraction jointly over the whole document. In particular, we study how to inject information from a knowledge base (KB) in such IE model, based on unsupervised entity linking. The used KB entity representations are learned from either (i) hyperlinked text documents (Wikipedia), or (ii) a knowledge graph (Wikidata), and appear complementary in raising IE performance. Representations of corresponding entity linking (EL) candidates are added to text span representations of the input document, and we experiment with (i) taking a weighted average of the EL candidate representations based on their prior (in Wikipedia), and (ii) using an attention scheme over the EL candidate list. Results demonstrate an increase of up to 5% F1-score for the evaluated IE tasks on two datasets. Despite a strong performance of the prior-based model, our quantitative and qualitative analysis reveals the advantage of using the attention-based approach.

4.1 Introduction

Information extraction (IE) comprises several subtasks, e.g., named entity recognition (NER), coreference resolution (coref), relation extraction (RE). State-of-the-art results mainly report performance on single tasks, usually solving them on a sentence level (especially NER, RE). However, in practice, IE system decisions should be consistent on the document level, e.g., when processing news articles to automatically link entities (aside from potentially learning, e.g., new relations). Yet, the challenge of solving the tasks jointly on a document level has not received as much attention and remains hard [1–3].

On the other hand, it is well established that IE models benefit from incorporating background information of knowledge bases (KBs). Still, so far this has been shown from the perspective of solving individual tasks such as relation classification or entity typing (e.g., [4, 5]). Integrating KBs in joint models, realizing and analyzing the more complex end-to-end setting, has been left unexplored.

In terms of the nature of KBs adopted in IE, current approaches use either (i) structured knowledge *graphs* comprising `(subj,rel,obj)` triples, e.g., Wikidata [6–8], or (ii) *textual* descriptions, usually in hyperlinked documents, e.g., Wikipedia [9, 10]. It has not been established to what extent KB-text and KB-graph entity representations complement each other in boosting IE performance.

We address both research gaps of (a) integrating KB information into a joint end-to-end IE model for solving named entity recognition, coreference resolution and relation extraction, and (b) analyzing what KB representation is more beneficial for IE, either *KB-graph* trained on Wikidata, or *KB-text* trained directly on Wikipedia. We particularly contribute: (i) a first span-based end-to-end architecture incorporating KB knowledge in a joint entity-centric setting, exploiting unsupervised entity linking (EL) to select KB entity candidates, (ii) exploration of prior- and attention-based mechanisms to combine the EL candidate representations into the model, (iii) assessment of the complementarity of KB-graph and KB-text representations, and (iv) consistent gains of up to 5% F1-score when incorporating KB knowledge in 3 document-level IE tasks evaluated on 2 different datasets.

4.2 Model

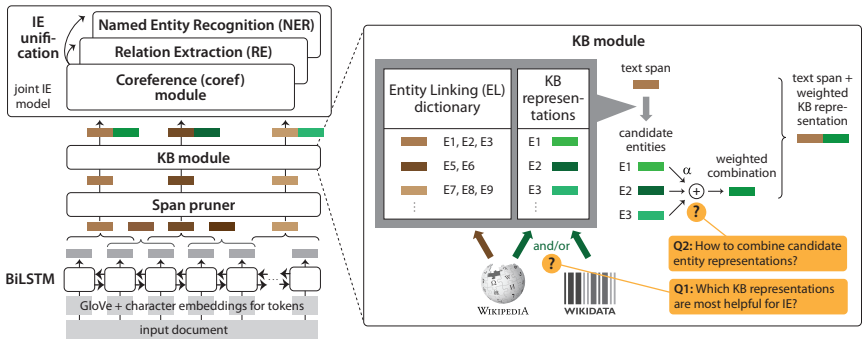


Figure 4.1: Joint information extraction (IE) model with addition of a knowledge base (KB) module.

Figure 4.1 illustrates our model architecture. Input document tokens are represented using concatenated GloVe [11] and character embeddings [12] and pushed through a BiLSTM to obtain contextualized token representations, which are combined into spans. Similar to [3, 13], a span pruner limits the number of spans for downstream modules. The *KB module* (Section 4.2.2) combines span representations with KB entity representations (Section 4.2.1), trained either on Wikidata (*KB-graph*) or Wikipedia (*KB-text*). The KB-enriched span representations then serve as input for joint predictions on downstream IE tasks (Section 4.2.3).

4.2.1 Entity representations

We experiment with 3 possible entity representations: *KB-text*, *KB-graph*, and concatenating *both*.

KB-text: We follow [14] to obtain the entity representations using a skip-gram architecture [15, 16], training to jointly predict (i) the linked entities (through Wikipedia hyperlinks) given the target entity, and (ii) the neighboring words for a given entity hyperlink.

KB-graph: We adopt [17] to train the entity embeddings directly on Wikidata triples `(subj,rel,obj)` by optimizing a linear classifier to predict the `obj` entity from the `subj` entity and the relation type `rel`.

4.2.2 KB module

For a span s_i from token l to r, we obtain the representation $\mathbf{g}_i$ as input to the KB module by concatenating the respective hidden LSTM states $\mathbf{h}_l$ and $\mathbf{h}_r$, and an embedding $\boldsymbol{\psi}_{r-l}$ for the corresponding span width $r - l$:

$$\mathbf{g}_i = [\mathbf{h}_l; \mathbf{h}_r; \boldsymbol{\psi}_{r-l}]. \tag{4.1}$$

We look up a given span s_i in a dictionary built from Wikipedia, to determine its candidate entities set[1] C_i, as well as the prior probability p_{ij} for each $c_{ij} \in C_i$, as per [14, §3].

To combine the KB candidates c_{ij}, we either use (i) a uniform average (*Uniform*), (ii) the prior weights p_{ij} (*Prior*), (iii) an attention scheme (*Attention*), or (iv) attention with prior information (*AttPrior*). The unnormalized attention scores for *Attention* and *AttPrior* are:

$$\Phi_{Attention}(s_i, c_{ij}, \kappa) = \mathcal{F}_A\left([\mathbf{g}_i; \boldsymbol{\xi}_\kappa(c_{ij})]\right) \tag{4.2}$$

$$\Phi_{AttPrior}(s_i, c_{ij}, \kappa) = \mathcal{F}_{AP}([\mathbf{g}_i; \boldsymbol{\xi}_\kappa(c_{ij}); p_{ij}]) \tag{4.3}$$

where $\kappa \in \{KB\text{-}text, KB\text{-}graph, both\}$ refers to the entity representations from Section 4.2.1, $\boldsymbol{\xi}_\kappa$ returns such representation for c_{ij}, and $\mathcal{F}_*$ is a feed-forward neural network (FFNN). The KB representation for span s_i is a weighted average of its candidates C_i:

$$\mathbf{e}_i^\kappa = \sum_{c_{ij} \in C_i} \oslash_{ij} \cdot \boldsymbol{\xi}_\kappa(c_{ij}) \tag{4.4}$$

where weights $\oslash_{ij}$ either are uniform ($1/|C_i|$), the prior p_{ij}, or softmax-normalized attention scores (softmax over Φ from Eq. (4.2) or Eq. (4.3)). The concatenation $[\mathbf{g}_i; \mathbf{e}_i^\kappa]$ forms the KB-enriched representation for span s_i, as input for IE modules (Section 4.2.3).

4.2.3 Joint IE model

The joint IE model comprises 3 modules (Fig. 4.1) using the same KB-enriched representations $[\mathbf{g}_i; \mathbf{e}_i^\kappa]$, and using a weighted combination of the 3 module losses to minimize during training. Note that NER and RE are framed as multi-label classification.

NER module: We use a FFNN on each span s_i to produce scores $\boldsymbol{\Phi}_{\text{NER}}(s_i) \in \mathbb{R}^{|L_E|}$, with L_E the set of possible entity types. At inference, we accept type $l \in L_E$ for span s_i if $\boldsymbol{\Phi}_{\text{NER}}(s_i)_l > 0$.

Coref module: We use the coreference scheme proposed by [18], using a FFNN to produce scores $\Phi_{\text{coref}}(s_i, s_j)$: at inference time, the highest scoring antecedent of span s_j is then chosen (potentially s_j itself). Indeed, to allow for singletons we accept self-references (s_j, s_j) if NER predicts the span s_j to be an entity.

RE module: Similar to [13, 19], we use a FFNN to produce scores $\boldsymbol{\Phi}_{\text{RE}}(s_i, s_j) \in \mathbb{R}^{|L_R|}$ for each pair of spans (s_i, s_j), with L_R the set of relation types. We accept relation $l \in L_R$ for pair (s_i, s_j) if $\boldsymbol{\Phi}_{\text{RE}}(s_i, s_j)_l > 0$.

IE unification: Above modules make span level predictions. We obtain entity-centric predictions using the coref clusters, by assigning the union of predicted entity/relation types within a coref cluster to all its members, as do [3].

Dataset	# Entity clusters	# Entity types	# Relations	# Relation types
DWIE	23,130	311	21,749	65
DocRED	98,610	6	50,503	96

Table 4.1: Dataset statistics.

4.3 Experimental setup

We evaluate our proposed models[2] on entity-centric multi-task datasets, summarized in Table 4.1: DWIE [3] and DocRED [2]. We report on coreference resolution (coref), NER and relation extraction (RE). For coref, we report the average of 3 common F1 scores, as implemented by [20]: MUC [21], B^3 [22] and $CEAF_e$ [23]. Since we focus on entity-centric, document-level IE, for NER and RE we use *hard* metrics [3] on the level of entity clusters (i.e., aforementioned coref clusters): predictions are counted as correct only if (i) all mentions (with exact boundary match) are present in the entity cluster, and (ii) the predicted entity type (for NER) or relation type between two clusters (for RE) is correct.

Our experiments address 2 main questions (see Fig. 4.1): **(Q1)** Which type of KB representation is most helpful for IE (*KB-text*, *KB-graph*, or *both*; see Section 4.2.1)? **(Q2)** Which weighting scheme to use for α (*Uniform*, *Prior*, *Attention*, *AttPrior*; see Section 4.2.2)?

4.4 Results

We summarize the comparison of various model choices for both DWIE and DocRED datasets in Table 4.2. First, looking into **(Q1)**, we note that including background information from *KB-graph* and *KB-text* significantly boosts performance compared to the *Baseline* without any KB. Additionally, our model outperforms the results from [3] (not listed in the table) by about 2 percentage points F1, using the same input (GloVe) representations. Furthermore, we observe a general improvement in results when combining *both* representations, suggesting that a (hyper)text corpus (Wikipedia) and a knowledge graph (Wikidata) embed complementary information for raising IE performance.

Deeper analysis reveals that adding KB representations mainly benefits performance for "rare" entity types: e.g., limiting the test set to entity types that occur $\leq$50 times in the training set for DWIE, compared to *Baseline*, F1 for NER goes up by $+13.9$ for *KB-both* with *AttPrior*, while the benefit gradually decreases for more frequently occurring entity types. For RE, we note that overall we also see a clear performance gain from adding KB information (e.g., +5.1% F1 for *both* KB sources with *AttProp* compared to *Baseline* for DWIE), yet the boost is not as clear for relations with fewer training instances. (The latter makes sense, since we inject KB representations of entities rather than explicitly also for relations; we leave studying adding relation embedding information for future work.)

[1] We limit this to the 16 most frequent ones.

[2] Code and models available at https://github.com/klimzaporojets/e2e-kb-ie.

KB Source	Setup	DWIE Coref	DWIE NER	DWIE RE	DocRED Coref	DocRED NER	DocRED RE
–	Baseline	90.0±0.2	71.7±0.5	47.0±1.4	81.9±0.3	68.5±0.3	23.5±0.6
KB-text	Uniform	90.7±0.2	73.5±0.5	48.5±1.1	82.9±0.1	70.7±0.2	24.5±0.3
	Attention	**90.7±0.3**	73.4±0.8	49.0±0.4	**83.4±0.1**	71.2±0.1	24.5±0.3
	AttPrior	90.7±0.3	73.7±0.6	**49.6±0.8**	83.2±0.2	**71.3±0.2**	24.8±0.4
	Prior	90.7±0.2	**73.8±0.5**	49.4±0.4	82.9±0.2	70.9±0.3	**25.3±0.4**
KB-graph	Uniform	91.0±0.3	73.6±0.4	48.0±1.2	83.3±0.2	71.1±0.2	24.9±0.2
	Attention	91.2±0.3	73.9±0.5	50.1±1.1	**83.7±0.1**	**71.6±0.1**	25.0±0.4
	AttPrior	**91.3±0.2**	**74.6±0.3**	**50.5±1.0**	83.5±0.3	71.5±0.2	25.1±0.2
	Prior	90.8±0.3	73.6±0.6	49.6±1.1	83.4±0.1	71.1±0.1	**25.2±0.2**
both (KB-graph + KB-text)	Uniform	91.1±0.1	74.1±0.5	49.3±0.5	83.5±0.1	71.3±0.2	24.8±0.1
	Attention	91.2±0.3	74.3±0.6	51.3±1.3	83.5±0.2	71.5±0.1	24.8±0.3
	AttPrior	**91.5±0.2**	**75.0±0.4**	**52.1±1.2**	**83.6±0.2**	**71.8±0.3**	**25.7±0.7**
	Prior	90.8±0.1	73.8±0.2	49.8±1.2	83.2±0.1	71.2±0.1	25.1±0.3

Table 4.2: Main results of the experiments in F1 scores grouped by the background KB source. We report Avg. F1 scores of MUC, B^3 and $CEAF_e$ for Coref, and hard F1 metrics for NER and RE. **Bold** font indicates the best results for each of the different *KB source* types. Additionally, the best overall results are underlined.

Second, for **(Q2)**, we note that the *AttPrior* scheme is the overall winner among the different EL candidate weigthing schemes. We observed that in terms of ranking EL candidates, *Prior* performs quite well on DWIE — for 86.5% of entity mentions it assigns the highest score to the correct EL candidate, while *Attention* and *AttPrior* achieve it for 46.2%, resp. 77.2% of the mentions — which basically confirms that DWIE has a similar entity distribution as Wikipedia.[3] Yet, it seems necessary to include alternative candidates, and the attention-based schemes thus can correct EL mistakes of *Prior*, as illustrated in Fig. 4.2. This correction leads to a resulting boost for the IE tasks as reported in Table 4.2. E.g., we found that for DWIE, looking at clusters with entity mentions for which *Prior* makes wrong EL predictions, the *AttPrior* weighting scheme retrieves +3.7% more of the gold standard annotated named entities (as opposed to just +0.6% in the clusters with correct *Prior* EL candidates). Perfecting the EL prediction would potentially boost IE performance even more.

4.5 Related work

As stated earlier, we studied how to integrate (i) knowledge base information into IE, and particularly (ii) end-to-end IE combining multiple tasks (NER, relation extraction, coreference resolution), while (iii) taking an entity-centric perspective, i.e., focus on making consistent decisions on the document level. For (i), integrating KB into IE has been applied for individual tasks: relation classification [6, 8, 24], entity

[3] DWIE is a news article corpus.

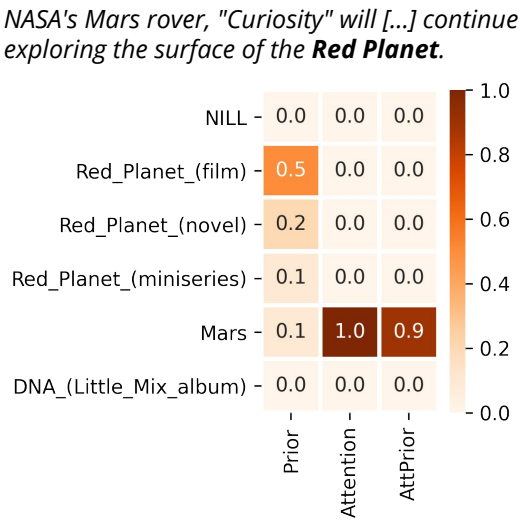


Figure 4.2: Illustration of EL candidate weighting: the α weights for top candidates for "Red Planet" from the example sentence at the top. Attention-based weighting (*Attention*, *AttPrior*) correctly identify the "Mars" entity, while the Wikipedia-based *Prior* fails, as most of Wikipedia's "Red Planet" links refer to the film.

typing [4] and NER [10]. For (ii), recently span-based architectures [13, 18, 25, 26] have been proposed. Our work unifies the KB integration concept into such span-based IE system, in particular an entity-centric one (as per (iii)), building on [3, 27]. For the KB integration approach, we exploit entity representations trained on a hypertext corpus, as in [10, 14, 28] or learnt from a knowledge graph [6–8]. Our results show that both offer complementary value for IE. Similarly to our work, [29] also explore using an attention-weighted combination of entity representations, but they use it to build a full document representation (with mentions having the entities as candidates) for a text classification task. In contrast, our span-based attention model is able to "inject" knowledge in each of the mentions separately, for more fine-grained downstream IE tasks that are mention-dependent, e.g., coreference resolution, relation extraction and NER.

4.6 Conclusion

We propose an end-to-end model for joint IE (NER + relation extraction + coreference resolution) incorporating entity representations from a background knowledge base (KB), using a span-based system. We find that representations built from a knowledge graph and a hypertext corpus are complementary in boosting IE performance. To combine candidate entity representations for text spans, we explore various weighting schemes: an attention-based combination is successful in com-

bining prior frequency information from a hypertext corpus with contextual information to identify the relevant entity, and achieves highest IE performance.

Acknowledgments

Part of the research leading to these results has received funding from (i) the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761488 for the CPN project,[4] and (ii) the Flemish Government under the programme "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen".

[4] https://www.projectcpn.eu/

References

[1] G. Durrett and D. Klein. *A joint model for entity analysis: Coreference, typing, and linking*. Transactions of the Association for Computational Linguistics (TACL 2014), 2:477–490, 2014. Available from: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/412.

[2] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. *DocRED: A Large-Scale Document-Level Relation Extraction Dataset*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019), pages 764–777, 2019. Available from: https://www.aclweb.org/anthology/P19-1074/.

[3] K. Zaporojets, J. Deleu, C. Develder, and T. Demeester. *DWIE: An entity-centric dataset for multi-task document-level information extraction*. Information Processing & Management, 58(4):102563, 2021. Available from: https://arxiv.org/abs/2009.12626.

[4] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. *Knowledge Enhanced Contextual Word Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 43–54, 2019. Available from: https://doi.org/10.18653/v1/D19-1005.

[5] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang. *K-BERT: Enabling Language Representation with Knowledge Graph*. In Proceedings of the 2020 Conference on Artificial Intelligence (AAAI 2020), pages 2901–2908, 2020. Available from: https://aaai.org/ojs/index.php/AAAI/article/view/5681.

[6] B. Yang and T. Mitchell. *Leveraging Knowledge Bases in LSTMs for Improving Machine Reading*. In Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (ACL 2017), pages 1436–1446, 2017. Available from: https://doi.org/10.18653/v1/P17-1132.

[7] X. Han, Z. Liu, and M. Sun. *Neural knowledge acquisition via mutual attention between knowledge graph and text*. In Proceedings of the 2018 Conference on Artificial Intelligence (AAAI 2018), volume 32, 2018. Available from: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16691.

[8] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, and H. Chen. *Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 3016–3025, 2019. Available from: https://doi.org/10.18653/v1/n19-1306.

[9] P. H. Martins, Z. Marinho, and A. F. Martins. *Joint Learning of Named Entity Recognition and Entity Linking*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (ACL 2019, SRW), page 190, 2019. Available from: https://www.aclweb.org/anthology/P19-2026.

[10] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention*. In Proceedings

of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6442–6454, 2020. Available from: https://doi.org/10.18653/v1/2020.emnlp-main.523.

[11] J. Pennington, R. Socher, and C. Manning. *GloVe: Global vectors for word representation*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pages 1532–1543, 2014. Available from: https://www.aclweb.org/anthology/D14-1162/.

[12] X. Ma and E. Hovy. *End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF*. In Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics (ACL 2016), pages 1064–1074, 2016. Available from: https://www.aclweb.org/anthology/P16-1101/.

[13] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi. *A general framework for information extraction using dynamic span graphs*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 3036–3046, 2019. Available from: https://www.aclweb.org/anthology/N19-1308/.

[14] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. *Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation*. In Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016), pages 250–259, 2016. Available from: https://doi.org/10.18653/v1/k16-1025.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient estimation of word representations in vector space*. In Proceedings of the 2013 International Conference on Learning Representations (ICLR 2013), pages 1–12, 2013. Available from: http://arxiv.org/abs/1301.3781.

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. *Distributed representations of words and phrases and their compositionality*. In Proceedings of the 2013 International Conference on Neural Information Processing Systems (NIPS 2013), pages 3111–3119, 2013. Available from: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

[17] A. Joulin, E. Grave, P. Bojanowski, M. Nickel, and T. Mikolov. *Fast linear model for knowledge graph embeddings*. arXiv:1710.10881, 2017. Available from: http://arxiv.org/abs/1710.10881.

[18] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. *End-to-end Neural Coreference Resolution*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 188–197, 2017. Available from: https://doi.org/10.18653/v1/d17-1018.

[19] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi. *Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), pages 3219–3232, 2018. Available from: https://doi.org/10.18653/v1/d18-1360.

[20] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube. *Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation*. In Proceedings of the 2014 Annual Meeting of the Association for Computational

Linguistics (ACL 2014), pages 30–35, 2014. Available from: https://doi.org/10.3115/v1/p14-2006.

[21] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. *A model-theoretic coreference scoring scheme*. In Proceedings of the 1995 conference on Message understanding (MUC6, 1995), pages 45–52, 1995. Available from: https://doi.org/10.3115/1072399.1072405.

[22] A. Bagga and B. Baldwin. *Algorithms for scoring coreference chains*. In Proceedings of the 1998 International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference (LREC 1998), pages 563–566, 1998. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.5848&rep=rep1&type=pdf.

[23] X. Luo. *On coreference resolution performance metrics*. In Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005), pages 25–32, 2005. Available from: https://www.aclweb.org/anthology/H05-1004/.

[24] N. Poerner, U. Waltinger, and H. Schütze. *E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT*. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 803–818, 2020. Available from: https://www.aclweb.org/anthology/2020.findings-emnlp.71.

[25] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi. *Entity, Relation, and Event Extraction with Contextualized Span Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 5788–5793, 2019. Available from: https://doi.org/10.18653/v1/D19-1585.

[26] H. Fei, Y. Ren, and D. Ji. *Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction*. Information Processing & Management, 57(6):102311, 2020. Available from: https://doi.org/10.1016/j.ipm.2020.102311.

[27] R. Jia, C. Wong, and H. Poon. *Document-Level N-ary Relation Extraction with Multiscale Representation Learning*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 3693–3704, 2019. Available from: https://doi.org/10.18653/v1/n19-1370.

[28] O.-E. Ganea and T. Hofmann. *Deep Joint Entity Disambiguation with Local Neural Attention*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 2619–2629, 2017. Available from: https://doi.org/10.18653/v1/d17-1277.

[29] I. Yamada and H. Shindo. *Neural Attentive Bag-of-Entities Model for Text Classification*. In Proceedings of the 2019 Conference on Computational Natural Language Learning (CoNLL 2019), pages 563–573, 2019. Available from: https://www.aclweb.org/anthology/K19-1052.

5

TempEL: Linking Dynamically Evolving and Newly Emerging Entities

In this chapter, we go one step further and analyze the evolution of the entities from temporal perspective. In order to achieve this, we create a new dataset which consists of 10 yearly snapshots of Wikipedia entities from 2013 until 2022. We further study how entity linking task is affected by (i) changes of existing entities in time, and (ii) creation of new emerging entities.. Furthermore, we do not restrict our analysis to the realm of named entities, but incorporate all existing entities and concepts defined in Wikipedia. Our analysis showcases a continual decrease of performance in time, indicating that the entities from later versions of Wikipedia are harder to disambiguate than entities from earlier versions. Additionally, we demonstrate that the decrease in performance is specially sharp for entities requiring additional new knowledge (e.g., new entities related to COVID-19 pandemic) for which the model was not pre-trained.

⋆⋆⋆

K. Zaporojets, L.A. Kaffee, J. Deleu, T. Demeester, C. Develder and I. Augenstein

Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track: NeurIPS, 2022.

Abstract In our continuously evolving world, entities change over time and new, previously non-existing or unknown, entities appear. We study how this evolu-

tionary scenario impacts the performance on a well established *entity linking* (EL) task. For that study, we introduce TempEL, an entity linking dataset that consists of time-stratified English Wikipedia snapshots from 2013 to 2022, from which we collect both *anchor mentions* of entities, and these *target entities*' descriptions. By capturing such temporal aspects, our newly introduced TempEL resource contrasts with currently existing entity linking datasets, which are composed of fixed mentions linked to a single static version of a target Knowledge Base (e.g., Wikipedia 2010 for CoNLL-AIDA). Indeed, for each of our collected temporal snapshots, TempEL contains links to entities that are *continual*, i.e., occur in all of the years, as well as completely *new* entities that appear for the first time at some point. Thus, we enable to quantify the performance of current state-of-the-art EL models for: (i) entities that are subject to changes over time in their Knowledge Base descriptions as well as their mentions' contexts, and (ii) newly created entities that were previously non-existing (e.g., at the time the EL model was trained). Our experimental results show that in terms of temporal performance degradation, (i) *continual* entities suffer a decrease of up to 3.1% EL accuracy, while (ii) for *new* entities this accuracy drop is up to 17.9%. This highlights the challenge of the introduced TempEL dataset and opens new research prospects in the area of time-evolving entity disambiguation.[1]

5.1 Introduction

Entity linking (EL) is a well-established task that is concerned with mapping anchor *mentions* in text to target *entities* that describe them in a Knowledge Base (KB) (e.g., Wikipedia).[2] Existing benchmark datasets for EL [3, 11–13] are composed of a fixed set of annotated mentions linked to a single version of a target KB. This static setup is oblivious to the inherently non-stationary nature of the entity linking task where both target entities as well as anchor mentions change over time. The example in Fig. 5.1 illustrates this time-evolving essence of entity linking with a simple evolutionary comparison between Wikipedia 2013 and 2022. It showcases two scenarios studied in the current paper: (i) temporal evolution of existing (*continual*) entities across temporal snapshots, and (ii) appearance of *new*, previously non-existent entities. For instance, the description of the *continual* entity *The Assembly* differs between Wikipedia 2013 and 2022. Furthermore, the context of a mention "Mejlis" referring to *The Assembly* also changes over time. Conversely, the *new* entity *Janssen COVID-19 vaccine* is newly introduced in 2021 with the corresponding mentions (e.g., "Johnson & Johnson" in Fig. 5.1) that are linked to it.

In this paper we introduce TempEL, a novel dataset to study this time-evolving aspect of the entity linking task. We therefore extract 10 equally spread yearly snapshots from English Wikipedia entities starting from January 1, 2013 until January 1, 2022. We use each of these temporal snapshots of Wikipedia to also extract anchor mentions with the surrounding text. Thus, TempEL captures the temporal

[1] TempEL dataset, code and models are made public at https://github.com/klimzaporojets/TempEL.

[2] Some of the related work [1–5] distinguishes between *entity disambiguation* and *entity linking* tasks. This latter including *mention detection* and *disambiguation* in an end-to-end setting. In the current work, we follow a more conservative naming convention [6–10], and use the term *entity linking* and *entity disambiguation* interchangeably.

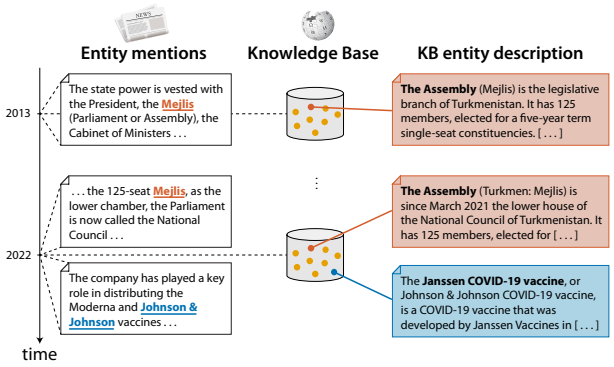


Figure 5.1: Illustration of KB entities changing over time: the "Mejlis" entity changes over time (both in its KB description and the contexts in which it is referenced to), while the Johnson & Johnson vaccine is an entirely new one that did not exist before.

evolution not only in the target entities as they are defined in the Wikipedia KB, but also in the contexts of anchor mentions linked to these entities. Each of the 10 temporal snapshots of our dataset is composed of training, test and validation sets with equal numbers of mentions and entities across the snapshots. Furthermore, TempEL is designed to comprise mentions pointing to *continual* entities across all the temporal snapshots, and to *new* entities inside a given temporal snapshot.

Finally, as a baseline, we finetune and evaluate the bi-encoder component of the BLINK model [7] on the various temporal snapshots of our newly introduced TempEL dataset. The bi-encoder is widely used in state-of-the-art entity linking models [4, 7] to retrieve the top K (in this work we experiment with $K = 64$) candidate target entities for a given anchor mention context. Furthermore, its straightforward finetuning and fast retrieval performance on millions of candidate entities [14], make it an ideal choice to test on TempEL. Our experiments demonstrate a consistent temporal model deterioration for mentions linked to both *continual* (3.1% accuracy@64 points) as well as *new* (17.9% accuracy@64 points) entities. A more detailed analysis reveals that the maximum drop in performance is observed for *new* entities that require fundamentally different world knowledge that was not present in the corpus originally used to pre-train BERT. This is e.g. the case for *new* entities related to COVID-19 for which the bi-encoder model suffers additional deterioration of 14% accuracy@64 points compared to the rest of the new entities.

5.2 Related work

Our work is related to multiple different, yet interconnected research areas described below. First, we explain how TempEL compares to the currently widely used *entity linking datasets*. Next, we relate our work to already existing *temporal datasets* covering different aspects of the temporal evolution of the data. Finally, we

describe the existing *entity-centric* research efforts, comparing the TempEL entity linking dataset to other datasets that heavily depend on the use of entities.

Entity linking datasets Most current state-of-the-art EL models [4, 15–18] report on datasets from predominantly the news domain such as AIDA [19], KORE50 [19], AQUAINT [20], ACE 2004, MSNBC [21], N^3 [22], DWIE [23], VoxEL [24], and TAC-KBP 2010-2015 [25, 26]. Other frequently used datasets include the web-based IITB [27] and OKE 15/16 [28], as well as the tweet-based Derczynski [29]. Additionally, larger yet automatically annotated datasets such as WNED-WIKI and WNED-CWEB [30] have been also widely adopted. Finally, a number of resources such as the domain-specific biomedical MedMentions [31], the zero-shot ZeShEL [8], and the multi tasking DWIE [23] and $AIDA^+$ [5] datasets have been recently introduced. Many of the mentioned datasets are further covered by entity linking evaluation frameworks such as GERBIL [11, 12] and KILT [13] that provide a common interface to evaluate the models. Yet, the mentioned resources are limited to static mention annotations linked to entities from a single version of a Knowledge Base. This contrasts with our newly introduced TempEL dataset, where the anchor mentions as well as the target entity descriptions are taken from different time periods. The datasets most closely related to our work are the recently introduced WikilinksNED [9, 32] and ShadowLink [33]. WikilinksNED contains only unseen mention-entity pairs in its test subset, thus encouraging the design of models invariant to overfitting and memorization biases. Furthermore, ShadowLink contains *overshadowed entities*: entities referred to by ambiguous mentions whose most likely target entity is different, e.g., the anchor mention "Michael Jordan" linked to the scientist instead of to the more widely referred to target entity describing the former basketball player. We incorporate the challenges presented in both of these datasets in TempEL (see Section 5.3.1 for further details).

Temporal datasets Research on temporal drift in data has gained a lot of interest in recent years. The focus has mostly been on creating datasets to train language models on different temporal snapshots of corpora derived from scientific [34], newswire [34, 35], Wikipedia [36], and Twitter [37] domains. More recently, temporal datasets have appeared to address tasks such as sentiment analysis [38–40], text classification [41, 42], named entity recognition [43, 44], question answering [34], and entity typing [45], among others. However, the creation of datasets tackling the temporal aspect of entity linking has largely been left unexplored. To the best of our knowledge, the dataset most closely related to TempEL is diaNED, introduced by [46]. There, the authors annotate mentions that require additional temporal information from the context to be correctly disambiguated. Conversely, in TempEL both mentions and entities are extracted from evolving temporal snapshots.

Entity-driven datasets Recent research has demonstrated the benefits of incorporating entity knowledge in various downstream tasks [47–53]. This progress has been accompanied by the creation of entity-driven datasets for tasks such as language modeling [54–56], question answering [57–61], fact checking [62–64] and information extraction [23, 65], to name a few. Yet, recent findings [66–71] suggest that entity *representation* and *identification* (i.e., identifying the correct entity that match a given text) are among the main challenges that should be solved

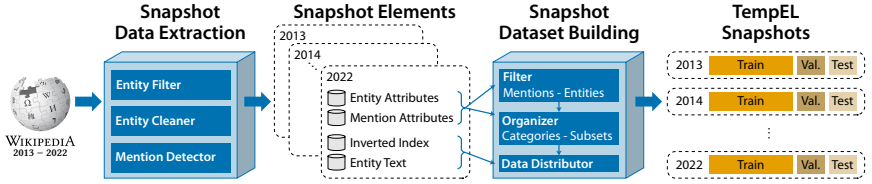


Figure 5.2: The pipeline to create our TempEL dataset. All the components are explained in Section 5.3.1.

to further increase performance on such datasets. We believe that TempEL can contribute to addressing these challenges by: (i) encouraging research on devising more robust methods to creating *entity representations* that are invariant to temporal changes; and (ii) improving entity identification for non-trivial scenarios involving ambiguous and uncommon mentions (e.g., linked to *overshadowed entities* as defined above).

5.3 The TempEL dataset

In this section we will provide details on how TempEL was constructed (Section 5.3.1), describing the main components of the creation pipeline as sketched in Fig. 5.2. Furthermore, we discuss the aspects taken into account to guarantee the overall quality of our dataset (Section 5.3.2). Finally, we present statistics of TempEL (Section 5.3.3), illustrating its dynamically evolving nature.

5.3.1 Dataset construction

Snapshot data extraction As Fig. 5.2 indicates, we start from the history log dumps from February 1, 2022 of Wikipedia itself. We first filter these (see *Entity Filter* in Fig. 5.2) to: (i) exclude pages that are irrelevant for TempEL (i.e., categories, disambiguation pages, redirects and lists); and (ii) select the most temporally stable version of a Wikipedia page from the last month of the year in order to avoid introducing more volatile and potentially corrupted content edits (see Section 5.3.2 for further details). Next, the Wikipedia pages are cleaned (see *Entity Cleaner* in Fig. 5.2) by stripping from the Wikitext markup content.[3] We use both regular expressions as well as the MediaWiki API for more difficult cases, such as the parsing of some of the Wikitext templates. Finally, we detect the mentions (see *Mention Detector* in Fig. 5.2) in each of the Wikipedia entity pages, filtering out the ones that point to anchors (i.e., subsections in Wikipedia pages), pages in languages other than English, files, red links (i.e., links pointing to not yet existing Wikipedia pages) and redirects.

The output of the *Snapshot Data Extraction* step first of all includes a set of *Entity* and *Mention Attributes* (e.g., the last modification date of the target entity), which are detailed in the supplementary material (see Section 5.A.6). These attributes form part of the final dataset, making it possible to perform additional analyses

[3] https://en.wikipedia.org/wiki/Help:Wikitext

of the results. Furthermore, the *Inverted Index* is generated to quickly access the Wikipedia pages that include a mention for a given target entity. Finally, the *Entity Text* files are extracted containing the (potentially yearly varying) textual content from the Wikipedia entity definition, as well as anchor mentions therein. These mentions of Wikipedia anchors that link to an entity will be extracted in the *Snapshot Dataset Building* step described further.

Snapshot dataset building Starting from the *Snapshot Elements* produced by the *Snapshot Data Extraction* process described above, the actual TempEL dataset is now generated. The first step is to apply an additional *Filter* to both entities and mentions with the goal of creating a more challenging dataset. This is done by excluding mentions for which the correct entity it refers to has the highest prior [72]. More formally, the *mention prior* is calculated as follows,

$$P(e|m) = |A_{e,m}|/|A_{*,m}|, \tag{5.1}$$

where $A_{*,m}$ is the set of all anchors that have the same mention m, and $A_{e,m}$ is the subset thereof that links to entity e. Additionally, we exclude the mentions whose normalized edit distance from the target entity title is below an established threshold.[4] By ignoring the mentions with the highest prior and exact match with the title, we ensure that TempEL contains non-trivial disambiguation cases where the naive approaches (e.g., defaulting to the most frequently linked entity for a given mention) would fail [7, 8, 30, 33].

Furthermore, the entities are organized (see *Organizer* in Fig. 5.2) into two *categories*: (i) *new*, emerging and previously non-existent entities that are introduced in a particular snapshot; and (ii) *continual* entities across all the temporal snapshots. Next, the mentions are divided in separate subsets (i.e., train, validation and test), with the constraint of normalized edit distance between the mentions in different subsets referring to the same target entity be higher than 0.2. This way, we expect to discourage potential models from memorizing the mapping between mentions and entities [9].

Finally, the data is distributed equally (see *Data Distributor* in Fig. 5.2) across all of the temporal snapshots. This way, the difference in performance can only be attributed to temporal evolution and not to inconsistencies related to dataset variability (e.g., different number of training instances in each of the temporal snapshots). Concretely, we enforce that the number of *continual* and *new* entities as well as the number of mentions stays the same across the temporal snapshots (see Table 5.1). We achieve this by performing a random mention subsampling in snapshots with higher number of mentions, weighted by the difference in the number of mentions-per-entity. This produces a very similar mention-entity distribution across the temporal snapshots. Finally, the filtered anchor mentions are located in the cleaned Wikipedia pages (i.e., the *Entity Text* in Fig. 5.2) using the *Inverted Index* created in the previous *Snapshot Data Extraction* step. The context of each of the mentions is further paired with the respective content of target pages, outputting this way the final TempEL dataset.

[4] During the generation of TempEL, we use a threshold of 0.2.

Statistic	Train	Validation	Test
Temporal Snapshots	10	10	10
Continual Entities	10,000	10,000	10,000
# Anchor Mentions	136,227	42,096	46,765
New Entities	373	373	373
# Anchor Mentions	1,764	1,231	1,450

Table 5.1: Summary statistics of TempEL. The number of entities and mentions is the same across all of the temporal snapshots.

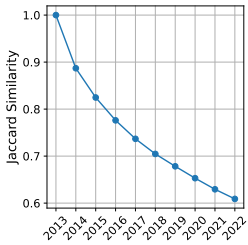
5.3.2 Quality control

Corrupted content Wikipedia is an open resource that relies on efforts of millions of Wikipedians to update and extend its contents.[5] As such, that content is not always reliable, with errors due to human mistakes or intentional vandalism. Despite efforts to prevent the introduction of such erroneous edits [73–75], we have detected numerous cases of corrupted entity descriptions during our preliminary tests. As a result, we adopted a simple, yet very effective heuristic: for each of the entities of a particular yearly snapshot, we select the most *stable* (i.e., the version of the entity that lasted the longest before being changed) content of the last month of the year (December). Due to the fact that most of the corrupted content is rolled back very quickly, and even automatically by specialized bots [76, 77], this heuristic is very robust. We double checked the correctness of the extracted content by manually inspecting the evolution of hundred entities with lowest Jaccard vocabulary similarity between temporal snapshots and observed no obviously erroneous entries.

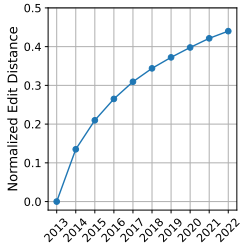
Entity relevance We filter out entities that have less than 10 in-links (i.e., number of mentions linking to the entity) or contain less than 10 tokens in its Wikipedia page in order to avoid including noisy content [32]. Additionally, in order to avoid evaluation bias towards mentions pointing to more popular entities [16, 78], we limit the number of mentions per entity to 10 for our test and validation sets. This way, we expect the accuracy scores to not be dominated by links to popular target entities (i.e., entities with a big number of incoming links).

Content filtering We only consider mentions linked to the main Wikipedia articles describing entities. The mentions pointing to anchors (subsections in a Wikipedia document), images, files, and wiki pages in other languages are filtered out in *Snapshot Data Extraction* step (see Fig. 5.2). In this step we also ignore pages that are not Wikipedia articles (e.g., files, information on Wikipedia users, etc.) as well as redirect pages. This way, the target entities as well as anchor mentions in our dataset are obtained from a cleaned list of candidate pages referring to entities that contain a meaningful textual description in Wikipedia.

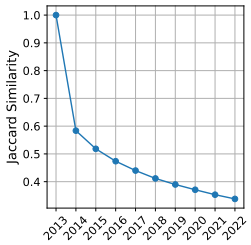
[5] https://en.wikipedia.org/wiki/Wikipedia:Wikipedians



(a) Evolution of entities in terms of Jaccard vocabulary similarity.



(b) Evolution of entities in terms of edit distance of the content.



(c) Evolution of context around the mentions (Jaccard similarity).

Figure 5.3: Change of textual content of entities and context around mentions across temporal yearly snapshots (x-axis).

Dataset distribution During the construction of TempEL, we constrain the subsets to be of equal size and contain similar mention-per-entity distributions across all the temporal snapshots. This is implemented in *Data Distributor* sub-component of the dataset creation pipeline (see Section 5.3.1). For example, the number of mentions linked to continual entities in our training subset is 136,227 across all of the snapshots (see Table 5.1 for further details). We argue that this setting will produce uniform, structurally unbiased snapshots. This will allow to study exclusively the temporal effect on the performance of the models for each of the different time periods. Our reasoning is supported by previous work demonstrating that the size alone of the training set [37] as well as a different distribution of the number of mentions per entity [16] can significantly affect the performance of the final model. Furthermore, we do not constrain the total number of entities from the Wikipedia KB to be equal across the temporal snapshots (see Fig. 5.4c), since we consider it a part of the evolutionary nature of the entity linking task (i.e., the temporal evolution of the target KB) we intend to study.

Flexibility and extensibility Finally, we provide a framework that can be used to re-generate the dataset with different parameters as well as to extend it with newer temporal snapshots. This includes the option to generate a new dataset with a customized number of temporal snapshots (e.g., quarterly instead of yearly spaced), different mention attributes (e.g., filtering by mention prior values), entity popularity (e.g., filtering out entities that have more than a certain number of in-links), among others (see Section 5.A.4 of the supplementary material for a complete list).

5.3.3 Dataset statistics

Table 5.1 summarizes the dataset statistics. We divide each of the temporal snapshots into train, validation and test subsets containing an equal number of *continual* and *new* entities. The number of mentions differs between the subsets since we limit the number of mentions per entity to 10 in both validation and test sets (see *entity relevance* in Section 5.3.2 for further details).

Additionally, we collect statistics related to temporal drift in content for both the target entities (Figs. 5.3a and 5.3b) as well as the context around the anchor mentions (Fig. 5.3c). Concretely, Fig. 5.3a visualizes Jaccard vocabulary similarity between the textual description of *continual* entities in 2013 and that of posterior yearly snapshots in TempEL. We observe a continual decrease, indicating that on average, the content of the entity description in Wikipedia is constantly evolving in terms of the used vocabulary. This is also supported by the graph in Fig. 5.3b, which showcases a continuous temporal increase of the average value of normalized edit distance across *continual* entities. Finally, Fig. 5.3c illustrates the temporal drift in the vocabulary (i.e., Jaccard vocabulary similarity) of the context around the mentions pointing to the same entity. We find it experiences a more significant change compared to the Jaccard similarity of entity content illustrated in Fig. 5.3a. This suggests that the context around the anchor mentions is subject to a higher degree of temporal transformation compared to that of target entities, making it an interesting item of future work.

5.4 Experiments

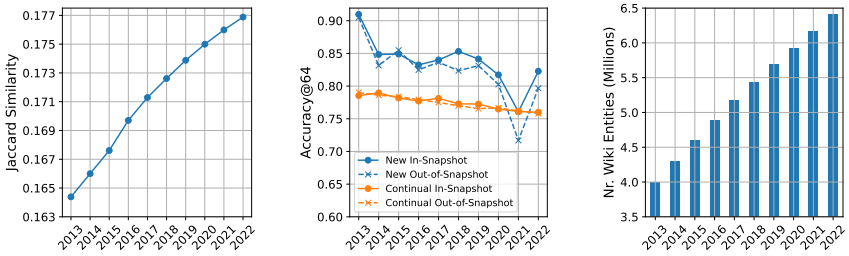
Our final TempEL comprises 10 different yearly snapshots and we evaluate entity linking (EL) performance on each of them individually. This evaluation setup allows us to study the effect of temporal corpus changes and assess the impact of increasing time lapses between the data used for model training and that on which the EL model is deployed [40, 42, 45]. We train a bi-encoder baseline EL model (detailed in Section 5.4.1) on the temporal snapshots from 2014 to 2022 separately and then evaluate EL performance using the test sets of both past and future snapshots.

More specifically, our experiments aim to answer the following research questions: **(Q1)** Does a fixed entity linking (EL) model's performance degrade when applied to newer content? **(Q2)** How does finetuning an EL model on more recent training data affect its performance on both old and newer content? **(Q3)** How does EL performance differ for resolving *new* versus *continual* entities?

5.4.1 Baseline

We experiment with the bi-encoder [79, 80] baseline introduced in the BLINK model [7]. This method independently encodes the mention contexts from the entity descriptions, and then performs the retrieval in a dense space [81] by matching the context of each mention with the closest candidate entities. For the entity description, we concatenate the title to the content of the page describing a particular entity. Both mention context as well as entity descriptions are truncated to 128 BERT tokens as per BLINK model [7]. Similarly to [37, 40], we start from a pre-trained BERT model,[6] which we finetune using our TempEL snapshots' training data — rather than fully re-training the BERT language model on the respective year's full Wikipedia corpus. We leave the latter full-fledged BERT (re-)training approach for future work.

[6] We use BERT-large, which is trained on a Wikipedia snapshot from 2018 [82].



(a) Similarity between candidates returned by the bi-encoder baseline.

(b) Difference in performance between *new* and *continual* entities.

(c) Evolution of the number of entities in the Wikipedia KB.

Figure 5.4: Statistics related to the analysis of the results (Section 5.4.2) across the temporal snapshots (x-axis).

5.4.2 Results and analysis

The results for *continual* and *new* entities are shown in Table 5.2. The rows thereof represent the snapshots whose train set we used to finetune the bi-encoder model, while the columns indicate the snapshots test data each of the finetuned models was tested on. The used metric is accuracy@64, which amounts to the fraction of anchor mentions in the test set for which the top-64 candidate entity list from the EL model includes the correct target. We observe a consistent temporal decrease in performance for *continual* entities **(Q1)**. This is also reflected in Fig. 5.4b, which illustrates the average temporal degradation across all the finetuned models. We hypothesize that this degradation over time is because, as time evolves, the relative "semantic distance" between the ever growing number of entities shrinks: entities become harder to distinguish from one another. In order to demonstrate this, we calculate the *Jaccard Similarity* between consecutive descriptions of the top 64 candidate entities returned by the bi-encoder. We observe a consistent increase in this similarity metric illustrated in Fig. 5.4a. This growth in more similar entities is accompanied with a general increase in the number of entities in the Wikipedia KB (see Fig. 5.4c). Consequently, the model is given an ever-increasing number of candidate target entities, which can potentially impact its performance.

Furthermore, we analyze the impact finetuning on different snapshots has on the performance of the model **(Q2)**. To this end, we distinguish between *in-snapshot* and *out-of-snapshot* finetuning setups. In *in-snapshot* setup, the bi-encoder model is finetuned and evaluated on the same snapshot. Conversely, in *out-of-snapshot* setting, the model is evaluated on a different snapshot than the one used for its finetuning. Figure 5.5a illustrates the difference in performance between the in-snapshot and out-of-snapshot predictions for new and continual entities. We observe a general increase in performance for in-snapshot finetuning with a marginal gain for *continual* entities compared to the *new* ones.[7] This general lower impact of in-snapshot finetuning on *continual* entities, leads us to hypothesize that the actual knowledge needed to disambiguate most of these entities in TempEL changes

[7]We analyze more in detail the difference in performance between *new* and *continual* entities in next paragraphs when addressing **(Q3)**.

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.785	0.782	0.778	0.772	0.769	0.762	0.758	0.758	0.754	0.750
2014	0.792	0.790	0.785	0.781	0.777	0.771	0.767	0.767	0.763	0.760
2015	0.786	0.784	0.782	0.777	0.773	0.769	0.765	0.764	0.760	0.757
2016	0.789	0.784	0.781	0.777	0.773	0.768	0.763	0.763	0.758	0.755
2017	0.794	0.791	0.788	0.785	0.781	0.775	0.771	0.772	0.768	0.763
2018	0.791	0.788	0.786	0.782	0.778	0.773	0.769	0.769	0.764	0.760
2019	0.795	0.792	0.789	0.784	0.781	0.776	0.772	0.773	0.767	0.765
2020	0.787	0.783	0.782	0.777	0.774	0.768	0.765	0.765	0.761	0.756
2021	0.788	0.785	0.782	0.777	0.773	0.769	0.764	0.764	0.761	0.757
2022	0.790	0.787	0.783	0.779	0.776	0.771	0.768	0.768	0.764	0.760

New Entities

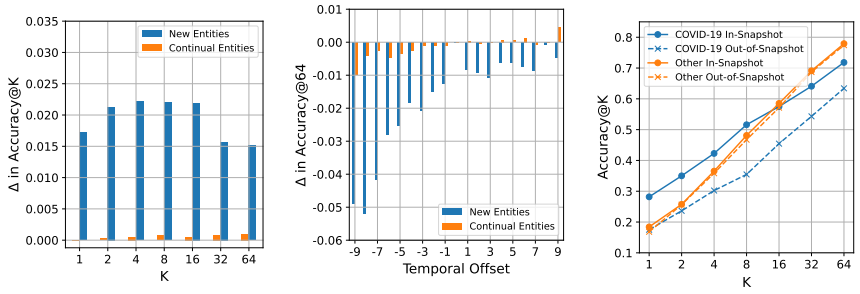
Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.910	0.819	0.853	0.826	0.841	0.812	0.819	0.791	0.688	0.774
2014	0.908	0.848	0.862	0.827	0.843	0.832	0.842	0.814	0.704	0.791
2015	0.898	0.823	0.849	0.822	0.808	0.813	0.832	0.788	0.706	0.781
2016	0.897	0.832	0.862	0.832	0.839	0.823	0.823	0.802	0.718	0.791
2017	0.906	0.832	0.857	0.817	0.840	0.824	0.835	0.791	0.714	0.808
2018	0.908	0.835	0.858	0.830	0.846	0.853	0.835	0.806	0.728	0.803
2019	0.910	0.842	0.853	0.821	0.842	0.843	0.841	0.810	0.734	0.799
2020	0.903	0.828	0.844	0.835	0.843	0.819	0.833	0.817	0.728	0.811
2021	0.910	0.825	0.852	0.825	0.837	0.817	0.830	0.814	0.761	0.812
2022	0.905	0.846	0.852	0.820	0.830	0.830	0.832	0.808	0.732	0.823

Table 5.2: Accuracy@64 for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

very little with time. In order to verify this hypothesis, we randomly selected 100 continual entity-mention pairs, and compared the difference in both mention contexts and entity descriptions between the years 2013 and 2022. We found that in most cases (>95%), while the textual description of the continual entity is changed (supported by Figs. 5.3a–5.3b), its meaning remains the same.

Moreover, we address the second part of **Q2** targeting the effect of timespan between the snapshot used for finetuning and the one used for evaluation. To accomplish this, in Fig. 5.5b we showcase the impact of in-snapshot finetuning relative to the *temporal offset* between the snapshot the model was tested and the snapshot the model was finetuned on. For negative temporal offset,[8] we observe a decrease in the performance difference between in-snapshot and out-of-snapshot setups as the offset approaches to zero. This indicates that the model can benefit more from

[8]Evaluation snapshot comes from later time period than the snapshot the model was finetuned on.



(a) Effect of in-snapshot finetuning (y-axis) across different accuracy thresholds K.

(b) In-snapshot finetuning (offset 0) compared to finetuning on past and future snapshots ($-$ and $+$ offsets).

(c) In-snapshot finetuning effect on COVID-19 related and other *new entities* from 2021 snapshot.

Figure 5.5: Impact of finetuning and evaluating on the same snapshot (*in-snapshot*) compared to finetuning and evaluating on different snapshots (*out-of-snapshot*). We observe: (a) a superior impact of in-snapshot finetuning on *new* entities compared to *continual* ones, (b) a decrease in performance when finetuning on increasingly older spanshots, and (c) dominant effect of in-snapshot finetuning on entities that require fundamentally new knowledge (e.g., COVID-19 related entities).

recent snapshots than from snapshots further in the past. Curiously, we observe a slight increase in performance for out-of-snapshot *continual* entities trained on future snapshots (positive temporal offsets in Fig. 5.5b). This suggests that the changes in continual entities are *accumulative* in Wikipedia, with later versions of entity descriptions also including the information from the past. For instance, we have observed that for entities describing people, the newly added information on the occupation (e.g., soccer coach) is appended to the occupation description a person had in the past (e.g., soccer player).

Next, we analyze the EL performance on *new* entities and whether they are differently affected than the *continual* ones **(Q3)**. We plot the in-snapshot and out-of-snapshot average temporal change in accuracy@64 scores across all finetuned models for both types of entities in Fig. 5.4b. We observe that, in general, the performance on *new* entities is superior to that on *continual* ones. Furthermore, as observed above, the performance gain from in-snapshot finetuning on new entities is superior compared to that on continual ones (supported by Fig. 5.4b and Figs. 5.5a–5.5b). This difference suggests that new entities require a higher degree of additional snapshot-specific knowledge to be correctly disambiguated. Additionally, the graph in Fig. 5.4b reveals that this delta in performance is larger for more recent years (starting from 2018). We hypothesize that this behaviour is due to the fact that the used original BERT model [83] has not been exposed to more recent new entities during pre-training. It also suggests a complementary effect between task-specific finetuning on TempEL dataset and language model pre-training on larger corpora.

Furthermore, to better understand the superior performance on new entities,

we manually analyze 100 randomly selected *new* entities from our dataset. We found that a large majority (∼90%) of entities were either events that are recurrent in nature (e.g., "2018 BNP Paribas Open") (∼68%) or extracts of already existing pages (∼22%). We conjecture[9] that these entities require little additional knowledge to be disambiguated, since either they already exist (as part of the content of other entities) or are very similar to already existing entities in Wikipedia. This contrasts sharply with the performance drop observed for *new* entities in the temporal snapshot 2021, as exhibited in both Fig. 5.4b and Table 5.2. This decrease is mostly driven by COVID-19 related entities, which constitute 24% of the new entities, which are linked to by 30% of the mentions in this spanshot. The disambiguation of these cases requires completely new and fundamentally different, previously non-existent knowledge. Since this knowledge is not present in the original corpus used to pre-train the BERT encoder nor in any of the previous snapshots, our EL model based on it struggles.

Finally, we analyze the impact of new entities finetuning **(Q2)** on the temporal snapshot 2021, for which our model exhibits the lowest temporal performance driven by COVID-19 disambiguation instances (see above). Figure 5.5c showcases the impact of in- and out-of-snapshot finetuning on the performance on COVID-19 related entities compared to *other* new entities for different thresholds K of the accuracy@K metric. We observe a large difference in performance (up to 14% accuracy@64 points) between COVID-19 related and the rest of the instances for out-of-snapshot finetuning. This difference is significantly decreased when finetuning on the 2021 snapshot (in-snapshot finetuning), achieving superior accuracy on COVID-19 related entities for lower values of K compared to *other* entities. In contrast, the difference between out- and in-snapshot performance on these non-COVID-19 related entities (*other* entities in Fig. 5.5c) is marginal. This suggests that in-snapshot finetuning has dominant impact on new entities that require fundamentally new, previously non-existent knowledge in Wikipedia.

5.5 Limitations and future work

A number of dataset and model-related aspects were left unexplored in the current work. Our clarifications thereof below may help the community to understand the limitations and potential future research directions to extend our efforts.

Effect of pre-training on new corpora Recent work has demonstrated the benefits of pre-training language models on more recent corpora (e.g., the latest Wikipedia versions) when applied on downstream tasks [37, 40]. We hypothesize that this pre-training may also improve EL performance for our TempEL, especially for *new* entities that require new world knowledge.

Changes in mention context Our work focused mostly on changes in target entities, leaving the effect of changes in mention context on EL performance unexplored. For example, Fig. 5.3c shows a notable temporal drop in Jaccard vocabulary similarity of the context surrounding mentions. This suggests that mentions, as

[9] See Section 5.A.11 of the supplementary material for further details on the performance on these different new entity types.

well as the text surrounding them, are quite volatile and evolve over time, making them an interesting subject for future research.

Cross-lingual time evolution Our dataset is limited to English Wikipedia. Yet, since recent work [84, 85] has shown the benefits of training EL models in a cross-lingual setting, studying cross-lingual temporal evolution of entity linking task may also be an interesting future research direction. Furthermore, it will complement the recent growing interest in creating entity linking datasets for a number of low-resourced languages [86–89].

5.6 Conclusion

This paper introduced TempEL, a new large-scale temporal entity linking dataset composed of 10 yearly snapshots of Wikipedia target entities linked to by anchor mentions. In our dataset creation pipeline, we put special focus on the quality assurance and future extensibility of TempEL. Furthermore, we established baseline entity linking results across different years, which revealed a noticeable performance deterioration on test data more recent than the training data. We further examined the most challenging cases, suggesting the need for updating the pretrained language model of our EL model, at least to perform well on newly appearing entities that require new world knowledge (e.g., in case of COVID-19). Finally, we described limitations of our work and discussed potential future research directions.

Acknowledgements

Part of the research leading to these results has received funding from (i) the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761488 for the CPN project,[10] (ii) the Flemish Government under the programme "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen", (iii) the Research Foundation – Flanders grant no. V412922N for Long Stay Abroad at Copenhagen University, and (iv) DFF Sapere Aude grant No 0171-00034B 'Learning to Explain Attitudes on Social Media (EXPANSE)'.

References

[1] O.-E. Ganea and T. Hofmann. *Deep Joint Entity Disambiguation with Local Neural Attention*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 2619–2629, 2017. Available from: https://www.aclweb.org/anthology/D17-1277.

[2] N. Kolitsas, O.-E. Ganea, and T. Hofmann. *End-to-End Neural Entity Linking*. In Proceedings of the 2018 Conference on Computational Natural Language Learning (CoNLL 2018), pages 519–529, 2018. Available from: https://www.aclweb.org/anthology/K18-1050/.

[3] Ö. Sevgili, A. Shelmanov, M. Y. Arkhipov, A. Panchenko, and C. Biemann. *Neural entity linking: A survey of models based on deep learning*. Semantic Web, 13(3):527–570, 2022. Available from: https://doi.org/10.3233/SW-222986.

[4] W. Zhang, W. Hua, and K. Stratos. *EntQA: Entity Linking as Question Answering*. In Proceedings of the 2022 International Conference on Learning Representations (ICLR 2022), 2022. Available from: https://arxiv.org/abs/2110.02369.

[5] K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution*. In Proceedings of the 2022 Annual Meeting of the Association for Computational Linguistics (ACL 2022), pages 778–784, 2022. Available from: https://aclanthology.org/2022.acl-short.88.

[6] D. Rao, P. McNamee, and M. Dredze. *Entity linking: Finding extracted entities in a knowledge base*. In Multi-Source, Multilingual Information Extraction and Summarization, pages 93–115. Springer, 2013. Available from: https://doi.org/10.1007/978-3-642-28569-1_5.

[7] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. *Zero-shot entity linking with dense entity retrieval*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6397–6407, 2020. Available from: https://aclanthology.org/2020.emnlp-main.519.

[8] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee. *Zero-Shot Entity Linking by Reading Entity Descriptions*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019), pages 3449–3460, 2019. Available from: https://aclanthology.org/P19-1335.

[10] https://www.projectcpn.eu/

[9] Y. Onoe and G. Durrett. *Fine-Grained Entity Typing for Domain Independent Entity Linking*. In Proceedings of the 2020 Conference on Artificial Intelligence (AAAI 2020), pages 8576–8583, 2020. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/6380.

[10] J. Raiman. *DeepType 2: Superhuman Entity Linking All You Need Is Type Interactions*. In Proceedings of the 2022 Conference on Artificial Intelligence (AAAI 2022), 2022. Available from: https://www.aaai.org/AAAI22Papers/AAAI-2612.RaimanJ.pdf.

[11] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, et al. *GERBIL: general entity annotator benchmarking framework*. In Proceedings of the 2015 International Conference on World Wide Web (WWW 2015), pages 1133–1143, 2015. Available from: https://doi.org/10.1145/2736277.2741626.

[12] M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo. *GERBIL–benchmarking named entity recognition and linking consistently*. Semantic Web, 9(5):605–625, 2018. Available from: https://doi.org/10.3233/SW-170286.

[13] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, et al. *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 (NAACL-HLT 2021), 2021. Available from: https://aclanthology.org/2021.naacl-main.200.

[14] J. Johnson, M. Douze, and H. Jégou. *Billion-scale similarity search with GPUs*. IEEE Transactions on Big Data, 7(3):535–547, 2021. Available from: https://doi.org/10.1109/TBDATA.2019.2921572.

[15] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto. *Global Entity Disambiguation with Pretrained Contextualized Embeddings of Words and Entities*. CoRR, abs/1909.00426, 2020. Available from: https://arxiv.org/abs/1909.00426.

[16] L. Orr, M. Leszczynski, S. Arora, S. Wu, N. Guha, X. Ling, and C. Re. *Bootleg: Chasing the tail with self-supervised named entity disambiguation*. In Proceedings of the 2021 Conference on Innovative Data Systems Research (CIDR 2021), 2021. Available from: http://cidrdb.org/cidr2021/papers/cidr2021_paper13.pdf.

[17] N. De Cao, G. Izacard, S. Riedel, and F. Petroni. *Autoregressive Entity Retrieval*. In Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021), 2021. Available from: https://openreview.net/forum?id=5k8F6UU39V.

[18] N. De Cao, W. Aziz, and I. Titov. *Highly Parallel Autoregressive Entity Linking with Discriminative Correction*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), pages 7662–7669, 2021. Available from: https://aclanthology.org/2021.emnlp-main.604, doi:10.18653/v1/2021.emnlp-main.604.

[19] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. *Robust disambiguation of named entities in text*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pages 782–792, 2011. Available from: https://www.aclweb.org/anthology/D11-1072/.

[20] D. Milne and I. H. Witten. *Learning to link with wikipedia*. In Proceedings of the 2008 ACM conference on Information and knowledge management (CIKM 2008), pages 509–518, 2008. Available from: https://doi.org/10.1145/1458082.1458150.

[21] L. Ratinov, D. Roth, D. Downey, and M. Anderson. *Local and global algorithms for disambiguation to Wikipedia*. In Proceedings of the 2011 Annual Meeting of the Association for Computational Linguistics (ACL 2011), pages 1375–1384, 2011. Available from: https://aclanthology.org/P11-1138/.

[22] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both. *N^3-A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format*. In Proceedings of the 2014 International Conference on Language Resources and Evaluation (LREC 2014), pages 3529–3533, 2014. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/856_Paper.pdf.

[23] K. Zaporojets, J. Deleu, C. Develder, and T. Demeester. *DWIE: An entity-centric dataset for multi-task document-level information extraction*. Information Processing & Management, 58(4):102563, 2021. Available from: https://doi.org/10.1016/j.ipm.2021.102563.

[24] H. Rosales-Méndez, A. Hogan, and B. Poblete. *VoxEL: a benchmark dataset for multilingual entity linking*. In Proceedings of the 2018 International Semantic Web Conference (ISWC 2018), pages 170–186, 2018. Available from: https://doi.org/10.1007/978-3-030-00668-6_11.

[25] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. *Overview of the TAC 2010 knowledge base population track*. In Proceedings of the 2010 Text Analysis Conference (TAC 2010), pages 1–25, 2010. Available from: https://blender.cs.illinois.edu/paper/kbp2010overview.pdf.

[26] H. Ji, J. Nothman, B. Hachey, and R. Florian. *Overview of TAC-KBP 2015 Tri-lingual Entity Discovery and Linking*. In Proceedings of the 2015 Text Analysis Conference (TAC 2015), 2015. Available from: https://tac.nist.gov/publications/2015/additional.papers/TAC2015.KBP_Trilingual_Entity_Discovery_and_Linking_overview.proceedings.pdf.

[27] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. *Collective annotation of Wikipedia entities in web text*. In Proceedings of the 2009 ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2009), pages 457–466, 2009. Available from: https://doi.org/10.1145/1557019.1557073.

[28] A. G. Nuzzolese, A. L. Gentile, V. Presutti, A. Gangemi, D. Garigliotti, and R. Navigli. *Open knowledge extraction challenge*. In Proceedings of the 2015 Semantic Web Evaluation Challenges (SemWebEval@ESWC 2015), pages 3–15, 2015. Available from: https://doi.org/10.1007/978-3-319-25518-7_1.

[29] L. Derczynski, D. Maynard, G. Rizzo, M. Van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. *Analysis of named entity recognition and linking for tweets*. Information Processing & Management, 51(2):32–49, 2015. Available from: https://doi.org/10.1016/j.ipm.2014.10.006.

[30] Z. Guo and D. Barbosa. *Robust Named Entity Disambiguation with Random Walks*. Semantic Web, 9(4):459–479, 2018. Available from: https://doi.org/10.3233/SW-170273.

[31] S. Mohan and D. Li. *MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts*. In Proceedings of the 2018 Automated Knowledge Base Construction (AKBC 2018), 2018. Available from: https://doi.org/10.24432/C5G59C.

[32] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy. *Named Entity Disambiguation for Noisy Text*. In Proceedings of the 2017 Conference on Computational Natural Language Learning (CoNLL 2017), pages 58–68, 2017. Available from: https://aclanthology.org/K17-1008.

[33] V. Provatorova, S. Bhargav, S. Vakulenko, and E. Kanoulas. *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 (EMNLP 2021), pages 10501–10510, 2021. Available from: https://aclanthology.org/2021.emnlp-main.820.

[34] A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liska, T. Terzi, M. Gimenez, C. de Masson d'Autume, T. Kocisky, S. Ruder, et al. *Mind the Gap: Assessing Temporal Generalization in Neural Language Models*. In Proceedings of the 2021 Advances in Neural Information Processing Systems (NeurIPS 2021), pages 29348–29363, 2021. Available from: https://proceedings.neurips.cc/paper/2021/hash/f5bf0ba0a17ef18f9607774722f5698c-Abstract.html.

[35] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen. *Time-Aware Language Models as Temporal Knowledge Bases*. Transactions of the Association for Computational Linguistics, 10:257–273, 2022. Available from: https://arxiv.org/abs/2106.15110.

[36] J. Jang, S. Ye, C. Lee, S. Yang, J. Shin, J. Han, G. Kim, and M. Seo. *TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models*. CoRR, abs/2204.14211, 2022. Available from: https://doi.org/10.48550/arXiv.2204.14211.

[37] D. Loureiro, F. Barbieri, L. Neves, L. E. Anke, and J. Camacho-Collados. *TimeLMs: Diachronic Language Models from Twitter*. In Proceedings of the 2022 Annual Meeting of the Association for Computational Linguistics (ACL 2022), pages 251–260, 2022. Available from: https://aclanthology.org/2022.acl-demo.25.

[38] J. Lukes and A. Søgaard. *Sentiment analysis under temporal shift*. In Proceedings of the 2018 Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA@EMNLP 2018), pages 65–71, 2018. Available from: https://doi.org/10.18653/v1/w18-6210.

[39] J. Ni, J. Li, and J. McAuley. *Justifying recommendations using distantly-labeled reviews and fine-grained aspects*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 188–197, 2019. Available from: https://aclanthology.org/D19-1018.

[40] O. Agarwal and A. Nenkova. *Temporal effects on pre-trained models for language processing tasks*. Transactions of the Association for Computational Linguistics (TACL 2022), 10:904–921, 2022.

[41] X. Huang and M. J. Paul. *Examining temporality in document classification*. In Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics (ACL 2018), pages 694–699, 2018. Available from: https://aclanthology.org/P18-2110.

[42] Y. He, J. Li, Y. Song, M. He, H. Peng, et al. *Time-evolving Text Classification with Deep Neural Networks*. In Proceedings of the 2018 International Joint Conference on Artificial Intelligence (IJCAI 2018), pages 2241–2247, 2018. Available from: https://doi.org/10.24963/ijcai.2018/310.

[43] L. Derczynski, K. Bontcheva, and I. Roberts. *Broad twitter corpus: A diverse named entity recognition resource*. In Proceedings of the 2016 International Conference on Computational Linguistics (COLING 2016), pages 1169–1179, 2016. Available from: https://aclanthology.org/C16-1111.

[44] S. Rijhwani and D. Preoţiuc-Pietro. *Temporally-informed analysis of named entity recognition*. In Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics (ACL 2020), pages 7605–7617, 2020. Available from: https://aclanthology.org/2020.acl-main.680.

[45] K. Luu, D. Khashabi, S. Gururangan, K. Mandyam, and N. A. Smith. *Time Waits for No One! Analysis and Challenges of Temporal Misalignment*. CoRR, 2021. Available from: https://arxiv.org/abs/2111.07408.

[46] P. Agarwal, J. Strötgen, L. Del Corro, J. Hoffart, and G. Weikum. *diaNED: Time-aware named entity disambiguation for diachronic corpora*. In Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics (ACL 2018), pages 686–693, 2018. Available from: https://aclanthology.org/P18-2109.

[47] B. Yang and T. Mitchell. *Leveraging Knowledge Bases in LSTMs for Improving Machine Reading*. In Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (ACL 2017), pages 1436–1446, 2017. Available from: https://doi.org/10.18653/v1/P17-1132.

[48] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. *Knowledge Enhanced Contextual Word Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 43–54, 2019. Available from: https://doi.org/10.18653/v1/D19-1005.

[49] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6442–6454, 2020. Available from: https://aclanthology.org/2020.emnlp-main.523.

[50] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. *REALM: Retrieval-augmented language model pre-training*. CoRR, abs/2002.08909, 2020. Available from: https://arxiv.org/abs/2002.08909.

[51] P. Verga, H. Sun, L. B. Soares, and W. Cohen. *Adaptable and interpretable neural memoryover symbolic knowledge*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 3678–3691, 2021. Available from: https://aclanthology.org/2021.naacl-main.288.

[52] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. *QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 535–546, 2021. Available from: https://aclanthology.org/2021.naacl-main.45.

[53] R. Liu, G. Zheng, S. Gupta, R. Gaonkar, C. Gao, S. Vosoughi, M. Shokouhi, and A. H. Awadallah. *Knowledge infused decoding*. In Proceedings of the 2022 International Conference on Learning Representations (ICLR 2022), 2022. Available from: https://openreview.net/forum?id=upnDJ7itech.

[54] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. *Language Models as Knowledge Bases?* In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 2463–2473, 2019. Available from: https://aclanthology.org/D19-1250.

[55] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou. *Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 3554–3565, 2021. Available from: https://aclanthology.org/2021.naacl-main.278.

[56] N. Kassner, P. Dufter, and H. Schütze. *Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models*. In Proceedings of the 2021 Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021), pages 3250–3258, 2021. Available from: https://aclanthology.org/2021.eacl-main.284.

[57] S. W.-t. Yih, M.-W. Chang, X. He, and J. Gao. *Semantic parsing via staged query graph generation: Question answering with knowledge base*. In Proceedings of the 2015 Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), 2015. Available from: https://doi.org/10.3115/v1/p15-1128.

[58] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension*. In Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (ACL 2017), pages 1601–1611, 2017. Available from: https://aclanthology.org/P17-1147.

[59] K. Jiang, D. Wu, and H. Jiang. *FreebaseQA: A New Factoid QA Data Set Matching Trivia-Style Question-Answer Pairs with Freebase.* In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 318–323, 2019. Available from: https://doi.org/10.18653/v1/n19-1028.

[60] P. Lewis, Y. Wu, L. Liu, P. Minervini, H. Küttler, A. Piktus, P. Stenetorp, and S. Riedel. *Paq: 65 million probably-asked questions and what you can do with them*. Transactions of the Association for Computational Linguistics, 9:1098–1115, 2021. Available from: https://arxiv.org/abs/2102.07033.

[61] A. Saxena, S. Chakrabarti, and P. Talukdar. *Question Answering Over Temporal Knowledge Graphs*. In Proceedings of the 2021 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), pages 6663–6676, 2021. Available from: https://aclanthology.org/2021.acl-long.520.

[62] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. *FEVER: a Large-scale Dataset for Fact Extraction and VERification*. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), pages 809–819, 2018. Available from: https://aclanthology.org/N18-1074.

[63] Y. Onoe, M. J. Zhang, E. Choi, and G. Durrett. *CREAK: A Dataset for Commonsense Reasoning over Entity Knowledge*. In Proceedings of the 2021 Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS 2021), 2021. Available from: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Abstract-round2.html.

[64] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal. *FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information*. In Proceedings of the 2021 Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS 2021), 2021. Available from: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/68d30a9594728bc39aa24be94b319d21-Abstract-round1.html.

[65] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. *DocRED: A Large-Scale Document-Level Relation Extraction Dataset*. In Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019), pages 764–777, 2019. Available from: https://aclanthology.org/P19-1074.

[66] A. Runge and E. Hovy. *Exploring Neural Entity Representations for Semantic Information*. In Proceedings of the 2020 BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP@EMNLP 2020), pages 204–216, 2020. Available from: https://aclanthology.org/2020.blackboxnlp-1.20.

[67] T. Févry, L. B. Soares, N. FitzGerald, E. Choi, and T. Kwiatkowski. *Entities as Experts: Sparse Memory Access with Entity Supervision*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 4937–4951, 2020. Available from: https://aclanthology.org/2020.emnlp-main.400.

[68] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. *Retrieval-augmented generation for knowledge-intensive nlp tasks*. In Proceedings of the 2020 Advances in Neural Information Processing Systems (NeurIPS 2020), pages 9459–9474, 2020. Available from: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[69] S. Verlinden, K. Zaporojets, J. Deleu, T. Demeester, and C. Develder. *Injecting Knowledge Base Information into End-to-End Joint Entity and Relation Extraction*

and Coreference Resolution. In Findings of the 2021 Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), pages 1952–1957, 2021. Available from: https://doi.org/10.18653/v1/2021.findings-acl.171.

[70] B. Heinzerling and K. Inui. *Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries*. In Proceedings of the 2021 Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021), pages 1772–1791, 2021. Available from: https://aclanthology.org/2021.eacl-main.153.

[71] R. Ri, I. Yamada, and Y. Tsuruoka. *mLUKE: The Power of Entity Representations in Multilingual Pretrained Language Models*. In Proceedings of the 2022 Annual Meeting of the Association for Computational Linguistics (ACL 2022), pages 7316–7330, 2022. Available from: https://aclanthology.org/2022.acl-long.505.

[72] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. *Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation*. In Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016), pages 250–259, 2016. Available from: https://doi.org/10.18653/v1/k16-1025.

[73] A. G. West, S. Kannan, and I. Lee. *Detecting Wikipedia vandalism via spatio-temporal analysis of revision metadata?* In Proceedings of the Third European Workshop on System Security (EUROSEC 2010), pages 22–28, 2010. Available from: https://doi.org/10.1145/1752046.1752050.

[74] Q. V. Dang and C.-L. Ignat. *Quality assessment of wikipedia articles without feature engineering*. In Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, pages 27–30, 2016. Available from: https://doi.org/10.1145/2910896.2910917.

[75] P. Wang and X. Li. *Assessing the quality of information on Wikipedia: A deep-learning approach*. Journal of the Association for Information Science and Technology, 71(1):16–28, 2020. Available from: https://doi.org/10.1002/asi.24210.

[76] L. Zheng, C. M. Albano, N. M. Vora, F. Mai, and J. V. Nickerson. *The roles bots play in Wikipedia*. In Proceedings of the 2019 ACM on Human-Computer Interaction (ACM SIGCHI 2019), pages 1–20, 2019. Available from: https://doi.org/10.1145/3359317.

[77] J. Jiang and M. A. Vetter. *The good, the bot, and the ugly: Problematic information and critical media literacy in the postdigital era*. Postdigital Science and Education, 2(1):78–94, 2020. Available from: https://doi.org/10.1007/s42438-019-00069-4.

[78] A. Chen, P. Gudipati, S. Longpre, X. Ling, and S. Singh. *Evaluating Entity Disambiguation and the Role of Popularity in Retrieval-Based NLP*. In Proceedings of the 2021 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), pages 4472–4485, 2021. Available from: https://aclanthology.org/2021.acl-long.345.

[79] P.-E. Mazare, S. Humeau, M. Raison, and A. Bordes. *Training Millions of Personalized Dialogue Agents*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), pages 2775–2779, 2018. Available from: https://aclanthology.org/D18-1298.

[80] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. *Wizard of Wikipedia: Knowledge-powered conversational agents*. In Proceedings of the 2018 International Conference on Learning Representations (ICLR 2018), 2018. Available from: https://openreview.net/forum?id=r1l73iRqKm.

[81] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. *Dense Passage Retrieval for Open-Domain Question Answering*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6769–6781, 2020. Available from: https://aclanthology.org/2020.emnlp-main.550.

[82] M. Joshi, O. Levy, L. Zettlemoyer, and D. S. Weld. *BERT for Coreference Resolution: Baselines and Analysis*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 5807–5812, 2019. Available from: https://www.aclweb.org/anthology/D19-1588.

[83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 4171–4186, 2019. Available from: https://www.aclweb.org/anthology/N19-1423.

[84] J. A. Botha, Z. Shan, and D. Gillick. *Entity Linking in 100 Languages*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 7833–7845, 2020. Available from: https://www.aclweb.org/anthology/2020.emnlp-main.630.

[85] N. De Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, and F. Petroni. *Multilingual Autoregressive Entity Linking*. Transactions of the Association for Computational Linguistics, 10:274–290, 2022. Available from: https://arxiv.org/abs/2103.12528.

[86] L. Hennig, P. T. Truong, and A. Gabryszak. *MobIE: A German Dataset for Named Entity Recognition, Entity Linking and Relation Extraction in the Mobility Domain*. In Proceedings of the 2021 Conference on Natural Language Processing (KONVENS 2021), pages 223–227, 2021. Available from: https://aclanthology.org/2021.konvens-1.22.pdf.

[87] M. Ogrodniczuk and W. Gruszczyński. *Wikipedia-Based Entity Linking for the Digital Library of Polish and Poland-Related News Pamphlets*. In Proceedings of the 2020 International Conference on Asian Digital Libraries (ICADL 2020), pages 81–88, 2020. Available from: https://doi.org/10.1007/978-3-030-64452-9_7.

[88] G. Caillaut, C. Gracianne, N. Abadie, G. Touya, and S. Auclair. *Automated construction of a French Entity Linking dataset to geolocate social network posts in the context of natural disasters*. In Proceedings of the 2022 International Conference on Information Systems for Crisis Response and Management (ISCRAM 2022), 2022. Available from: https://hal.archives-ouvertes.fr/hal-03631387/document.

[89] H. Rosales Méndez. *Towards a fine-grained entity linking approach*. PhD thesis, Universidad de Chile, 2021. Available from: https://repositorio.uchile.cl/

bitstream/handle/2250/181834/Towards-a-fine-grained-entity-linking-approach.pdf?sequence=1.

[90] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford. *Datasheets for datasets*. Communications of the ACM, 64(12):86–92, 2021. Available from: https://doi.org/10.1145/3458723.

[91] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. *Transformers: State-of-the-art natural language processing*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020), pages 38–45, 2020. Available from: https://aclanthology.org/2020.emnlp-demos.6.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default TODO to Yes, No, or N/A. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? Yes. See the supplementary materials.

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   (b) Did you describe the limitations of your work? Yes
   (c) Did you discuss any potential negative societal impacts of your work? N/A
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? N/A
   (b) Did you include complete proofs of all theoretical results? N/A
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes. The link to the dataset will be shared as part of the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes. See the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? No. No additional computational resources for this, yet the results across multiple temporal snapshots used to finetune are consistent.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes. See the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes

(b) Did you mention the license of the assets? Yes. See supplementary material.

(c) Did you include any new assets either in the supplemental material or as a URL? No

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? N/A

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? N/A

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? N/A

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? N/A

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A

5.A Supplementary material

5.A.1 Dataset and code distribution

Link to the dataset The reviewers can access the dataset using the following link: https://cloud.ilabt.imec.be/index.php/s/RinXy8NgqdW58RW. The dataset and the baseline code will be made publicly available in a dedicated GitHub repository upon acceptance.

License TempEL is distributed under Creative Commons Attribution-ShareAlike 4.0 International license (CC BY-SA 4.0).[11]

Maintenance The maintenance and extension to further temporal snapshots of TempEL will be carried out by the authors of the paper. Additionally, we will make the code public to create potential new variations and extensions of TempEL using a number of hyperparameters (see Sections 5.A.4 and 5.A.5 for further details).

5.A.2 Datasheet for TempEL

In this section we provide a more detailed documentation of the dataset with the intended uses. We base ourselves on the datasheet proposed by [90].

5.A.2.1 Motivation

For what purpose was the dataset created? The TempEL dataset was created to evaluate how the temporal change of anchor mentions and that of target Knowledge Base (KB; i.e., modification or creation of new entities) affects the *entity linking* (EL) task. This contrasts with the currently existing datasets [3, 11–13], which are associated with a single version of the target KB such as the Wikipedia 2010 for the widely adopted CoNLL-AIDA [19] dataset. We expect that TempEL will encourage research in devising new models and architectures that are robust to temporal changes both in mentions as well as in the target KBs.

Who created the dataset and on behalf of which entity? The dataset is the result of joint effort involving researchers from the University of Copenhagen and Ghent University.

Who funded the creation of the dataset? The creation of TempEL was funded by the following grants:

1. FWO (Fonds voor Wetenschappelijk Onderzoek) long-stay abroad grant V412922N.
2. The Flemish Government fund under the programme "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen".

[11] https://creativecommons.org/licenses/by-sa/4.0/

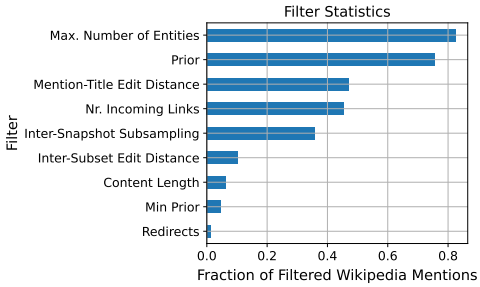


Figure 5.6: Figure showcasing the fraction of filtered Wikipedia mentions by each of the filters executed during TempEL generation.

5.A.2.2 Composition

What do the instances that comprise the dataset represent? Each of the instances consists of a mention in Wikipedia linked to target entity, i.e., a Wikipedia page, with a set of attributes. The dataset is organized in 10 yearly temporal snapshots starting from January 1, 2013 until January 1, 2022. See Section 5.A.6 for further details on the attributes associated with each of the instances of our TempEL dataset.

How many instances are there in total? Table 5.1 of the main manuscript summarizes the number of instances (# Anchor Mentions) of each of the entity categories (*continual* and *new*) in TempEL. See Section 5.A.3 for additional statistics on mention per entity distribution.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? TempEL contains a sample of all the possible anchor mentions linked to target entities from Wikipedia. The following are the filters applied to obtain the instances in the final TempEL dataset whose effect is also summarized in Fig. 5.6:

1. **Prior-based filtering**: we exclude all the mentions for which the correct entity it refers to has the highest *prior* [72] as calculated in Eq. (5.1) of the manuscript. This filtering is done with the goal of creating a more challenging dataset.

   Value to create TempEL: mentions with mention prior rank > 1 among other mentions referring to the same entity.

   Percentage of filtered out instances: between 74.20% and 76.28%, depending on the temporal snapshot.

   Hyperparameter name: `min_men_prior_rank` (see Table 5.3 in Section 5.A.4).

2. **Entity relevance filtering**: we impose the restriction for target entity of having at least 10 incoming links (i.e., at least 10 mentions linking to it) in order to be included in TempEL. Additionally, we filter out target entities whose

description contains less than 10 tokens. This is done in order to avoid introducing potentially noisy and irrelevant entities that have not been sufficiently established by the Wikipedia community.

Value to create TempEL: 10 for minimum number of incoming links and 10 for minimum content length (in number of tokens) of target entity.

Percentage of filtered out instances:

- Minimum number of incoming links: between 42.66% and 48.32%, depending on the temporal snapshot.
- Minimum content length: between 0.06% and 0.95% depending on the temporal snapshot.

Hyperparameter names: `min_nr_inlinks` for minimum number of incoming links and `min_len_target_ent` for minimum number of content length tokens (see Table 5.3 in Section 5.A.4).

3. **Min prior subsampling**: the mentions with very low mention prior are filtered out from TempEL. This way, we avoid introducing too infrequent and potentially erroneous mentions to refer to a particular entity.

   Value to create TempEL: 0.0001

   Percentage of filtered out instances: between 0.37% and 0.61%, depending on the snapshot.

   Hyperparameter name: `min_men_prior` (see Table 5.3 in Section 5.A.4).

4. **Minimum mentions per entity**: has similar effect as previously explained *min prior subsampling* (see above) filter. We do not use it in the creation of TempEL, relying completely on the *min prior subsampling* filter.

   Value to create TempEL: 1

   Percentage of filtered out instances: 0%

   Hyperparameter name: `min_mens_per_ent` (see Table 5.3 in Section 5.A.4).

5. **Edit distance mention title**: filters out the anchor mentions that are very similar to target entity page. This way, we expect to reduce the trivial cases where the entity linking can be simply predicted by mapping the mention to the title of the target entity.

   Value to create TempEL: 0.2 (normalized edit distance).

   Percentage of filtered out instances: between 44.85% and 48.99%, depending on the snapshot.

   Hyperparameter name: `ed_men_title` (see Table 5.3 in Section 5.A.4).

6. **Redirect filtering**: we filter out anchor mentions that point to redirect pages (pages without content redirecting to other pages in Wikipedia).

   Percentage of filtered out instances: between 1.02% and 1.47%, depending on the snapshot.

7. **Inter-subset filtering**: we enforce normalized edit distance between the mentions in different subsets referring to the same target entity to be higher than 0.2. This entails that the entities in TempEL are linked to at least by 3 mentions with different surface form. The main goal of this filter is to avoid mention-entity tuple memorization by the models [9].

Value to create TempEL: 0.2 normalized edit distance between mentions in different subsets.

Percentage of filtered out instances: 10%.

Hyperparameter name: `ed_men_subsets` (see Table 5.3 in Section 5.A.4).

8. **Maximum number of entities**: we restrict the number of target entities to 10,000 for *continual* instances. The reason behind this is to build a dataset of manageable size with a reasonable number of target entities to experiment with.

   Value to create TempEL: 10,000 for *continual* entities.

   Percentage of filtered out instances: 82%.

   Hyperparameter name: `nr_ct_ents_per_cut` (see Table 5.3 in Section 5.A.4)

9. **Maximum number of mentions per entity**: this filtering limits the number of mentions per entity in order for the dataset to not be dominated by most popular entities. Particularly, for test and evaluation subsets we limit the number of mentions per entity to 10. This way, we expect the accuracy scores to not be dominated by links to popular target entities (i.e., entities with a big number of incoming links). The limit for training set is higher (500), since we want it to be representative of the real mention per entity distribution in Wikipedia. The effect of imposing this limits can be observed in Fig. 5.7 for both *continual* as well as *new* entities represented by a significant leap in the mentions-per-entity curve, particularly noticeable for validation and test subsets.

   Value to create TempEL: 10 for validation and test subsets, 500 for the train subset.

   Percentage of filtered out instances: for *continual* instances, 84% for validation and test subsets and 28% for the train subset. For *new* instances, 45% for validation and test subsets and 0.3% for the train subset.

   Hyperparameter name: `max_mens_per_ent` (see Table 5.3 in Section 5.A.4).

10. **Inter-snapshot subsampling**: finally, we enforce that the number of continual and new entities as well as the number of mentions stays the same across the temporal snapshots (see Table 5.1). We achieve this by performing a random mention subsampling in snapshots with higher number of mentions, weighted by the difference in the number of mentions-per-entity. This produces a very similar mention-entity distribution across the temporal snapshots (see Section 5.A.3 for further details).

    Percentage of filtered out instances: between 5% and 35%, it increases for more recent temporal snapshots as they have more instances in Wikipedia.

We do not filter on any attribute that could potentially produce evident biases in TempEL (e.g., gender, geographic location of the entities, etc.).

What data does each instance consist of? Each instance of a snapshot consists of:

1. Cleaned contextual text surrounding the anchor mention from the Wikipedia snapshot. Furthermore, we include the bert-tokenized version of the text used in our baseline.

2. Cleaned textual description of the target entity taken from the Wikipedia snapshot. Furthermore, we include the bert-tokenized version of the text used in our baseline.
3. A set of additional attributes defining the anchor mention and target entity.

For more details about the attributes, see Section 5.A.6. Furthermore, concrete examples of TempEL's instances are showcased in Section 5.A.10.

Is there a label or target associated with each instance? Yes, the target entity is represented by the Wikipedia page id. Furthermore, we also pair it with Wikidata QID of the corresponding Wikidata entity. These targets correspond to the attributes `target_page_id` and `target_qid` described in Table 5.4 (see Section 5.A.6 for further details).

Is any information missing from individual instances? No, all the instances should have a complete information corresponding to the content as well as to the attributes.

Are relationships between individual instances made explicit? Yes, the relations between each of the instances and the target entity are made explicit by means of `target_page_id` and `target_qid` attributes (see Section 5.A.6 for further details), which uniquely identify the id of the Wikipedia page describing a particular entity and the Wikidata entity respectively.

Are there recommended data splits (e.g., training, development/validation, testing)? Yes, the dataset is divided in train, validation and test subsets (see Table 5.1 for the distribution).

Are there any errors, sources of noise, or redundancies in the dataset? We have taken multiple measures to build a high quality dataset, minimizing the number of noise or other errors (see Section 5.3.2 of the main manuscript). Yet, TempEL is not 100% error free, and contains a few errors mostly due to erroneous Wikitext edits by the Wikipedia users.

Is the dataset self-contained, or does it link to or otherwise rely on external resources? Yes, the dataset is self contained and consists of:

1. Instances divided in train, validation and test subsets (see Table 5.1).
2. A description of all the entities of each of the Wikipedia snapshots. These entities form the complete candidate pool used by the models to predict the correct target entity. Figure 5.4c of the main manuscript illustrates the temporal evolution in size of the number of candidate entities.

Does the dataset contain data that might be considered confidential? No, Wikipedia is a public resource.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? No, we haven't detected instances of such characteristics in TempEL.

Does the dataset identify any subpopulations (e.g., by age, gender)? While there are articles on different subpopulations on Wikipedia, there is no emphasis of the dataset on identifying or annotating those.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? Only based on their Wikipedia article, no editor information is retained.

Does the dataset contain data that might be considered sensitive in any way? Wikipedia is overall a resource aiming to be factual, therefore we can exclude this concern for most instances of TempEL.

5.A.2.3 Collection process

How was the data associated with each instance acquired? The textual data of the context of anchor mention and that of the description of the target entity is directly taken from the Wikipedia snapshots. Conversely, the attributes associated with each of the instances are calculated (see Section 5.A.6 for further details).

What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? The dataset was collected using the Wikipedia dumps from February of 2022. We detail further on the aspects related to the preprocessing, cleaning and labeling of TempEL instances in Section 5.A.2.4 of the datasheet.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)? The dataset was automatically generated based on existing Wikipedia articles. Therefore, no human intervention was needed for the dataset generation.

Over what timeframe was the data collected? The TempEL dataset was collected from 10 yearly snapshots of Wikipedia starting from January 1, 2013 until January 1, 2022.

Were any ethical review processes conducted (e.g., by an institutional review board)? N/A

5.A.2.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? The Wikipedia history logs content is available exclusively in Wikitext markup format.[12] In order to obtain cleaned text we proceed as follows:

1. We use MediaWiki API to process the templates which can not be parsed using regular expressions. For example, this is the case of the Wikitext template `Convert`, where the markup like "`{{convert|37|mm|in|abbr=on}}`" is converted to "`1.5 in`".
2. We use regular expressions to extract mentions and links. While this can also be done using online Wikitext parsing tools, we found that these did not account for all the corner cases of mention parsing such as the ones involving the *pipe trick*.[13]
3. Finally, we use `mwparserfromhell`[14] tool for parsing the rest of the Wikitext content.

Furthermore, our dataset files also contain BERT tokenization of the context around the mentions as well as the textual content of entities.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? Yes, the raw data containing the Wikipedia history logs was saved on our cloud server in the following link: https://cloud.ilabt.imec.be/index.php/s/BF9SkmQG2Tdjw8o.

Is the software that was used to preprocess/clean/label the data available? Yes, the software will be made public upon acceptance.

5.A.2.5 Uses

Has the dataset been used for any tasks already? Yes, in our submitted manuscript we describe a retriever bi-encoder baseline [7] (see Section 5.4.2).

Is there a repository that links to any or all papers or systems that use the dataset? N/A

What (other) tasks could the dataset be used for? The covered task is temporally evolving entity linking.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? N/A

[12] https://en.wikipedia.org/wiki/Help:Wikitext
[13] https://en.wikipedia.org/wiki/Help:Pipe_trick
[14] https://github.com/earwig/mwparserfromhell

Are there tasks for which the dataset should not be used? N/A

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? Yes, the dataset is of public access.

How will the dataset be distributed (e.g., tarball on website, API, GitHub)? The TempEL dataset will be made public on a GitHub repository together with the code to generate it. The baseline code and models will also be made public on the same repository. Due to the size, the dataset files will be hosted on the cloud server that belongs to Internet Technology and Data Science Lab (IDLab) at Ghent University (https://cloud.ilabt.imec.be/index.php/s/RinXy8NgqdW58RW).

When will the dataset be distributed? The dataset will be publicly distributed upon the submission of the camera ready version of our manuscript.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? The TempEL dataset will be distributed under Creative Commons Attribution-ShareAlike 4.0 International license (CC BY-SA 4.0).

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? N/A

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? N/A

5.A.2.6 Maintenance

Who will be supporting/hosting/maintaining the dataset? The maintenance and extension of TempEL will be carried out by the authors of the paper. Additionally, we will make the code publicly available to create potential new variations of TempEL using a number of hyperparameters (see Section 5.A.4 and Section 5.A.5 for further details).

The dataset files will be hosted on the cloud server that belongs to Internet Technology and Data Science Lab (IDLab) at Ghent University (https://cloud.ilabt.imec.be/index.php/s/RinXy8NgqdW58RW).

How can the owner/curator/manager of the dataset be contacted (e.g., email address)? The owners of the dataset can be contacted at the following e-mail address: klim.zaporojets@ugent.be.

Is there an erratum? No, there is no erratum yet.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? The TempEL will be regularly updated with newer snapshots (see Section 5.A.5). In circumstances such as labeling errors, we will release the fixed version of the dataset with the respective version number. The introduction of the new version will be communicated using the TempEL GitHub repository.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)? N/A

Will older versions of the dataset continue to be supported/hosted/maintained? Yes, the older version of the dataset will continue to be supported and hosted. All the versions will be numbered and we will provide the link to access each of these versions on our cloud storage server.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Yes, we provide the code and functionality to re-generate and extend the dataset with new temporal snapshots (see Sections 5.A.4 and 5.A.5). Yet, it is the responsibility of the users to provide hosting and maintenance to the newly generated dataset variations.

5.A.3 Mentions per entity distribution

Figure 5.7 illustrates the similarity of mention per entity distribution across the temporal snapshots. This is achieved using weighted random subsampling so all the snapshots have equal number of instances (see *Data Distributor* component description in Section 5.3.1). By enforcing this similarity between temporal snapshots, we ensure that the potential difference in the results is independent of cross-snapshot dataset distributional variations and only influenced by the dynamic temporal evolution of the content in TempEL.

5.A.4 Dataset creation hyperparameters

Table 5.3 summarizes the hyperparameters that can be tuned in order to automatically create the TempEL dataset. This way, it is possible for the user to create different variation of the TempEL. The most relevant hyperparameter is `snapshots` that is used to specify the temporal intervals to create the snapshots. Below we detail two possible options we provide to specify such intervals.

Option 1 - explicit snapshot specification The user is expected to provide a list of timestamps in the format of `YYYY-MM-DDTHH:MM:SSZ`, each one defining a different snapshot.

[5] For train, validation and test sets respectively.

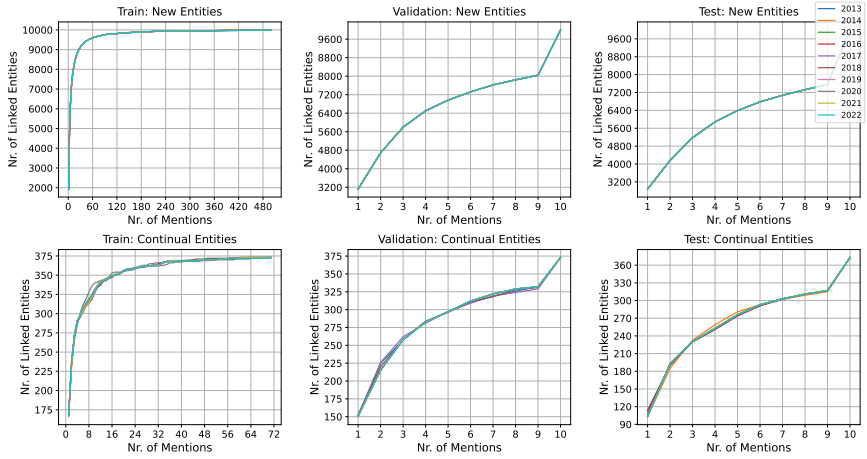


Figure 5.7: Similar distribution of the data across the temporal snapshots (number of mentions per entity). This structurally unbiased setting enable to study exclusively the temporal effect on the performance of the models for each of the different time periods.

Option 2 - time span and interval This option enables the user to define start and end dates of the time span from which the snapshots should be extracted. Furthermore, the interval value (i.e., by using keywords such as "weekly" or specifying the interval in seconds) has to also be specified.

5.A.5 Dataset extension

Additionally, we provide the option to extend the already existing dataset with new snapshots. Similarly as in the creation of new dataset (see Section 5.A.4 above), the `snapshots` hyperparameter is used to specify new snapshots which are then added to already existing TempEL dataset.

5.A.6 Mention and entity attributes

Hyperparamter	Description	TempEL
`snapshots`	Details (e.g., timestamps) of the temporal snapshots to be generated.	10 years
`nr_ct_ents_per_cut`	Number of *continual* entities per snapshot.	10,000
`min_mens_per_ent`	Minimum number of links a particular mention needs to have to target entity in order to be considered to be added in TempEL.	1
`min_men_prior`	Minimum mention prior (see Eq. (5.1) in the main manuscript).	0.0001
`max_men_prior`	Maximum mention prior.	0.5
`min_men_prior_rank`	Minimum rank of mention prior among all the mentions pointing to a specific entity.	2
`min_ent_prior`	Minimum entity prior as defined in [72]: the ratio of links to the entity with respect to all of the links in the Wikipedia snapshot.	0.0
`max_ent_prior`	Maximum entity prior.	1.0
`min_nr_inlinks`	Minimum number of incoming links per entity.	10
`min_len_target_ent`	Minimum length of target entity page (in tokens).	10
`max_mens_per_ent`	Maximum number of mentions per entity.	$500/10/10^{15}$
`ed_men_title`	Minimum normalized edit distance between the mentions and the title of the target page they are linked to.	0.2
`ed_men_subsets`	Minimum normalized edit distance between the mentions in different subsets linked to the same target entity.	0.2
`stable_interval`	In seconds, the interval of time before the end of each snapshot from which the most stable version of Wikipedia has to be taken (see Section 5.3.2 for further details).	2,592,000 (30 days)
`equal_snapshots`	Whether the number of instances and the number of mentions per entity distribution is the same across the snapshots (see Section 5.3.2 for further details). Equal cross-snapshot mention per entity distribution in Fig. 5.7 is the result of setting this hyperparameter in True.	True

Table 5.3: Hyperparameters that can be tuned during TempEL dataset creation

Attribute	Description
subset	The name of current subset (i.e., train, validation or test).
target_page_id	The unique Wikipedia page id of the target entity.
target_qid	The unique Wikidata QID of the target entity.
snapshot	The timestamp of the temporal snapshot from which the anchor mention and target entity attributes were extracted.
target	The textual content of the target entity Wikipedia page.
target_len	The length in tokens of target Wikipedia page.
target_title	The title of target entity Wikipedia page.
category	Category of the target entity (*new* or *continual*).
mention	The text of the mention.
context_left	The textual context to the left of the mention.
context_right	The textual context to the right of the mention.
anchor_len	The length in tokens of the Wikipedia page where the anchor mention is located.
ed_men_title	Normalized edit distance between the anchor mention and the title of the target Wikipedia page.
overlap_type	Overlap type between the anchor mention and the target title as defined by [8].
men_prior	The mention prior (see Eq. (5.1) of the main manuscript).
men_prior_rank	The rank of the current anchor mention compared to other mentions in Wikipedia pointing to target entity.
avg_men_prior	The average value of prior of the mentions linked to the target entity in Wikipedia for snapshot.
ent_prior	Entity prior as defined in [72]: the ratio of links to the entity with respect to all of the links in the Wikipedia snapshot.
nr_inlinks	Total number of incoming links to target entity.
nr_dist_mens	Number of distinct (i.e., with different surface form) mentions linked to target entity.
nr_mens_per_ent	Number of times the current mention appears in Wikipedia linked to target entity.
nr_mens_-extracted	Number of anchor mentions per current target entity in the subset.
anchor_-creation_date	The creation date (timestamp) of Wikipedia page where the anchor mention is located.
anchor_-revision_date	The timestamp of when the anchor Wikipedia page was last revised.
target_-	The timestamp of when the target Wikipedia entity

Table 5.4 describes the anchor mention and target entity related attributes present in TempEL. These attributes can be used to perform more in-depth analysis of the results.

5.A.7 Baseline implementation details

We base our bi-encoder baseline model on the publicly available BLINK code.[16] We train all the models for 10 epochs with the learning rate of 1e-04 and the batch size of 64. We use AdamW optimizer with 10% of warmup steps. Finally, we rely on `transformers` library [91] to get the pre-trained BERT-large representations. All the experiments were run on NVIDIA V100 GPU with the following execution times:

1. *Training*: 36 hours to train for 10 epochs per single snapshot.
2. *All Wikipedia entity encoding*: 7 days per finetuned model (on all the 10 Wikipedia snapshots) running on a single V100 GPU.
3. *Evaluation*: 30 seconds per finetuned model per snapshot using FAISS [14] library on GPU.

5.A.8 Total amount of compute and the type of resources used to create TempEL

In this section we provide the details on the computational resources used in each of the processing steps (see Section 5.3.1 and Fig. 5.2 for further details) to create the TempEL dataset:

1. *Snapshot Data Extraction*: this processing step is responsible for creating the snapshots from the Wikipedia log files from February 1, 2022. This is a multi-processing step that is executed on a cluster with 80 CPUs and 110 GB of RAM and takes 5 days and 8 hours to complete.
2. *Snapshot Dataset Building*: this is a multi-processing step that is executed on a cluster with 30 CPUs and 250 GB of RAM and takes 5 hours to complete.

5.A.9 License of the assets

We base the implementation of our baseline bi-encoder model on the publicly available BLINK [7] code. This asset is made available under MIT License (https://opensource.org/licenses/MIT).

5.A.10 Examples

This section presents two illustrative examples of instances in TempEL. The first example contains the anchor mention linked to *continual* entity, while the second one is the example of a link to *new* entity. Both of the examples were taken from the snapshot of January 1, 2021. Furthermore, we trim the content length (e.g., `target` attribute value) to only a few tokens for space reasons.

[16] https://github.com/facebookresearch/BLINK

5.A.10.1 Example 1: continual target entity

Table 5.5 illustrates an example of the link to *continual* target entity *Sacramental_bread*. It is worth noting that the creation date of this entity in Wikipedia (`target_creation_date` attribute) is of January 3, 2005. Yet, the version saved in the snapshot (`target_revision_date` attribute) is from December 30, 2020.

Attribute	Value
subset	train
target_page_id	1359030
target_qid	Q207104
snapshot	2021-01-01T00:00:00Z
target	"Sacramental bread, sometimes called altar bread, Communion ..."
target_len	7,568
target_title	"Sacramental_bread".
category	continual
mention	"host"
context_left	"... devotional image, portrait or other religious symbol (such as the"
context_right	"). Garland paintings were typically collaborations between a ..."
anchor_len	6,519
ed_men_title	0.9411
overlap_type	LOW_OVERLAP
men_prior	0.0750
men_prior_rank	7
avg_men_prior	0.6864
ent_prior	1.7790e-6
nr_inlinks	225
nr_dist_mens	13
nr_mens_per_ent	79
nr_mens_-extracted	58
anchor_-creation_date	2009-09-25T21:09:07Z
anchor_-revision_date	2020-10-04T16:15:13Z
target_-creation_date	2005-01-03T17:41:14Z
target_-revision_date	2020-12-30T12:38:50Z

Table 5.5: Example of the instance corresponding to mention link to *continual* entity (Sacramental_bread created in 2005-01-03) in TempEL.

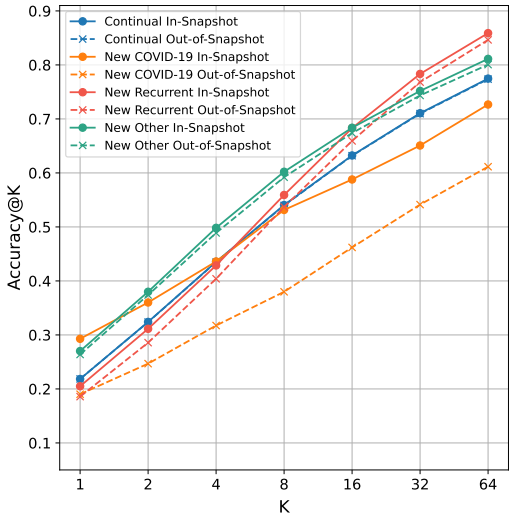


Figure 5.8: Accuracy@K for different values of $K \in \{1, 2, 4, 8, 16, 32, 64\}$. The results are grouped in four main categories: (i) mentions linked to *continual* entities that exist in all of the TempEL snapshots, (ii) mentions linked to *COVID-related* new entities (i.e., with keywords such as "COVID" in target entity title), (iii) mentions linked to *recurrent* new entities (i.e., entities representing events occurring periodically such as "2018 BNP Paribas Open"), and (iv) mentions linked to *other* new entities.

5.A.10.2 Example 2: new target entity

Table 5.6 illustrates an example of the link to *new* target entity *COVID-19_pandemic_in_Portland,_Oregon*. It is worth noting that the creation date of this entity in Wikipedia (`target_creation_date` attribute) is of March 23, 2020, which belongs to the interval of the considered snapshot: from January 1, 2020 until January 1, 2021.

5.A.11 Additional results

Tables 5.7-5.13 present the results for different accuracy@K for $K \in \{1, 2, 4, 8, 16, 32, 64\}$. Furthermore, Fig. 5.8 illustrates the mean in- and out-of-snapshot (see Section 5.4.2 of the main manuscript) accuracy@K performance across temporal snapshots on the following four target entity categories:

1. *Continual*: all the target *continual* entities (i.e., the entities that exist across all the temporal snapshots in TempEL dataset).
2. *COVID-19*: target *new* entities that have COVID-related (e.g., "COVID", "coronavirus", etc.) terms in the target entity title.
3. *Recurrent*: target *new* entities whose titles contain the year and some of the keywords (e.g., "league", "election", "cup", etc.) that indicate that an entity

is a repetitive event (e.g., "2018 BNP Paribas Open" which is part of *yearly* BNB Paribas Open competitions).

4. *Other*: all the other target *new* entities.

The following are the main conclusions that can be drawn from the graph in Fig. 5.8 that support or complement the findings described in Section 5.4.2 of the main manuscript:

1. New entities that require fundamentally new, previously non-existent knowledge to be disambiguated tend to have the lowest out-of-snapshot performance. This is the case of COVID-19 related disambiguation instances. These instances also experience the highest boost in performance when evaluated on in-snapshot setting (i.e., the model is evaluated and finetuned on the same temporal snapshot).
2. The difference between in- and out-of-snapshot performances on *continual* entities is the lowest. This is also supported by Fig. 5.4b and Figs. 5.5a–5.5b in the main manuscript. This suggests that the actual knowledge needed to disambiguate most of the *continual* entities in TempEL changes very little with time.
3. The model has the highest accuracy@64 performance on *recurrent* new entities. Yet, the performance on these entities drops sharply for lower values of K. We hypothesize that predicting the correct recurrent event gets more challenging as K decreases because of the large number of very similar candidates to pick from (e.g., many "BNP Paribas Open" championships that only differ in very few details such as the date).
4. The difference between in- and out-of-snapshot performance for *other* new entities is lower than for *recurrent* and *COVID-19* related ones. This is driven by new entities that are derived from existing entities in Wikipedia (i.e., their content is a copy of already established entities). We hypothesize that the model requires little additional knowledge to disambiguate these entities. Still, it is part of future work to study *other* new entities more in detail in order to find cases that represent intrinsically new knowledge similar to the identified COVID-19 entity cluster.

Attribute	Value
subset	train
target_page_id	63449958
target_qid	Q88484856
snapshot	2021-01-01T00:00:00Z
target	"The COVID-19 pandemic was confirmed to have reached ..."
target_len	26,432
target_title	"COVID-19_pandemic_in_Portland,_Oregon"
category	new
mention	"COVID-19 pandemic"
context_left	"Xico Xico and Xica both offered pickup service during the"
context_right	", as of May 2020. "
anchor_len	2,437
ed_men_title	0.5405
overlap_type	AMBIGUOUS_SUBSTRING
men_prior	0.0009
men_prior_rank	4
avg_men_prior	0.2548
ent_prior	2.9255e-7
nr_inlinks	37
nr_dist_mens	3
nr_mens_per_ent	23
nr_mens_-extracted	18
anchor_-creation_date	2020-12-08T00:23:50Z
anchor_-revision_date	2020-12-09T15:41:18Z
target_-creation_date	2020-03-23T04:22:55Z
target_-revision_date	2020-11-16T03:59:06Z

Table 5.6: Example of the instance corresponding to mention link to *new* entity (COVID-19_pandemic_in_Portland,_Oregon created in 2020-03-23) in TempEL.

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.225	0.219	0.215	0.217	0.212	0.206	0.203	0.203	0.197	0.192
2014	0.229	0.226	0.220	0.221	0.217	0.212	0.211	0.207	0.203	0.197
2015	0.228	0.223	0.219	0.219	0.216	0.211	0.208	0.206	0.204	0.196
2016	0.230	0.227	0.222	0.221	0.218	0.214	0.211	0.208	0.205	0.199
2017	0.240	0.237	0.229	0.229	0.226	0.221	0.219	0.216	0.211	0.207
2018	0.238	0.236	0.228	0.229	0.226	0.222	0.219	0.217	0.211	0.206
2019	0.237	0.235	0.228	0.228	0.226	0.220	0.217	0.216	0.212	0.208
2020	0.232	0.227	0.223	0.221	0.219	0.214	0.210	0.209	0.205	0.199
2021	0.239	0.235	0.231	0.230	0.228	0.222	0.219	0.217	0.213	0.210
2022	0.238	0.235	0.229	0.229	0.226	0.222	0.218	0.218	0.214	0.206

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.280	0.226	0.253	0.203	0.230	0.198	0.226	0.144	0.168	0.212
2014	0.291	0.268	0.258	0.201	0.234	0.217	0.245	0.150	0.159	0.214
2015	0.252	0.206	0.206	0.181	0.194	0.179	0.210	0.139	0.174	0.193
2016	0.277	0.248	0.242	0.214	0.221	0.206	0.226	0.144	0.181	0.206
2017	0.271	0.226	0.223	0.176	0.230	0.201	0.219	0.144	0.173	0.204
2018	0.284	0.255	0.240	0.190	0.228	0.268	0.246	0.157	0.178	0.222
2019	0.278	0.243	0.237	0.177	0.223	0.230	0.230	0.130	0.174	0.203
2020	0.284	0.236	0.225	0.206	0.214	0.201	0.212	0.183	0.177	0.221
2021	0.291	0.236	0.232	0.195	0.219	0.229	0.230	0.183	0.214	0.217
2022	0.294	0.260	0.251	0.188	0.206	0.241	0.240	0.170	0.170	0.219

Table 5.7: **Accuracy@1** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.337	0.330	0.324	0.322	0.317	0.311	0.306	0.302	0.301	0.293
2014	0.339	0.335	0.329	0.328	0.322	0.317	0.314	0.310	0.306	0.299
2015	0.339	0.333	0.327	0.325	0.323	0.317	0.312	0.309	0.305	0.299
2016	0.341	0.334	0.328	0.326	0.322	0.316	0.314	0.310	0.306	0.301
2017	0.351	0.346	0.338	0.338	0.332	0.328	0.324	0.320	0.316	0.309
2018	0.348	0.342	0.336	0.334	0.331	0.327	0.323	0.322	0.315	0.309
2019	0.348	0.345	0.337	0.335	0.332	0.325	0.322	0.320	0.317	0.310
2020	0.341	0.336	0.330	0.327	0.322	0.316	0.312	0.310	0.307	0.300
2021	0.349	0.344	0.338	0.335	0.331	0.325	0.321	0.319	0.315	0.310
2022	0.348	0.343	0.336	0.336	0.331	0.325	0.321	0.320	0.317	0.309

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.401	0.322	0.359	0.310	0.327	0.309	0.340	0.266	0.236	0.291
2014	0.397	0.366	0.357	0.318	0.328	0.347	0.357	0.278	0.234	0.306
2015	0.358	0.305	0.319	0.277	0.276	0.294	0.304	0.265	0.249	0.272
2016	0.379	0.351	0.345	0.344	0.308	0.320	0.315	0.270	0.244	0.311
2017	0.372	0.328	0.340	0.290	0.317	0.313	0.339	0.266	0.250	0.294
2018	0.395	0.369	0.346	0.305	0.326	0.380	0.344	0.270	0.250	0.306
2019	0.397	0.363	0.346	0.296	0.303	0.344	0.341	0.250	0.249	0.294
2020	0.385	0.343	0.337	0.321	0.294	0.323	0.319	0.301	0.250	0.315
2021	0.392	0.338	0.346	0.303	0.308	0.334	0.333	0.301	0.286	0.307
2022	0.408	0.372	0.355	0.301	0.294	0.352	0.336	0.289	0.250	0.322

Table 5.8: **Accuracy@2** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities										
Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.449	0.442	0.439	0.433	0.428	0.422	0.417	0.417	0.411	0.405
2014	0.455	0.448	0.443	0.439	0.433	0.428	0.424	0.423	0.416	0.410
2015	0.455	0.446	0.444	0.438	0.434	0.427	0.422	0.422	0.415	0.409
2016	0.453	0.446	0.442	0.437	0.432	0.426	0.422	0.422	0.415	0.408
2017	0.464	0.458	0.454	0.448	0.443	0.438	0.434	0.433	0.428	0.423
2018	0.461	0.453	0.449	0.445	0.440	0.437	0.430	0.431	0.425	0.417
2019	0.462	0.455	0.452	0.446	0.443	0.437	0.433	0.434	0.427	0.421
2020	0.455	0.446	0.442	0.438	0.433	0.427	0.422	0.423	0.417	0.411
2021	0.461	0.454	0.450	0.445	0.440	0.434	0.429	0.428	0.423	0.416
2022	0.460	0.453	0.450	0.444	0.440	0.433	0.429	0.430	0.424	0.417

New Entities										
Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.512	0.442	0.479	0.429	0.426	0.421	0.455	0.392	0.328	0.397
2014	0.526	0.486	0.475	0.434	0.446	0.463	0.489	0.410	0.317	0.406
2015	0.479	0.414	0.452	0.401	0.372	0.403	0.430	0.389	0.337	0.377
2016	0.500	0.464	0.466	0.463	0.418	0.434	0.430	0.408	0.330	0.414
2017	0.507	0.448	0.452	0.401	0.428	0.445	0.474	0.394	0.328	0.408
2018	0.520	0.487	0.477	0.428	0.435	0.496	0.469	0.388	0.340	0.417
2019	0.517	0.486	0.482	0.419	0.415	0.475	0.472	0.398	0.339	0.403
2020	0.506	0.449	0.457	0.418	0.414	0.443	0.443	0.414	0.331	0.428
2021	0.509	0.453	0.457	0.422	0.421	0.446	0.439	0.417	0.383	0.427
2022	0.527	0.491	0.472	0.439	0.397	0.471	0.474	0.422	0.341	0.434

Table 5.9: **Accuracy@4** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.556	0.551	0.546	0.539	0.532	0.526	0.520	0.520	0.513	0.507
2014	0.563	0.559	0.553	0.546	0.540	0.532	0.527	0.526	0.520	0.514
2015	0.561	0.555	0.552	0.543	0.540	0.533	0.526	0.526	0.520	0.514
2016	0.559	0.554	0.550	0.542	0.537	0.531	0.524	0.524	0.518	0.511
2017	0.569	0.565	0.562	0.555	0.549	0.542	0.537	0.537	0.530	0.525
2018	0.567	0.561	0.558	0.550	0.544	0.537	0.532	0.531	0.523	0.519
2019	0.571	0.565	0.562	0.554	0.550	0.541	0.537	0.537	0.529	0.524
2020	0.561	0.555	0.553	0.545	0.539	0.532	0.527	0.528	0.522	0.515
2021	0.565	0.559	0.557	0.548	0.544	0.535	0.530	0.530	0.524	0.519
2022	0.566	0.560	0.556	0.549	0.545	0.537	0.532	0.533	0.527	0.521

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.632	0.572	0.585	0.563	0.541	0.539	0.574	0.517	0.425	0.504
2014	0.633	0.624	0.586	0.565	0.569	0.561	0.599	0.538	0.421	0.531
2015	0.603	0.541	0.559	0.524	0.495	0.534	0.562	0.510	0.425	0.497
2016	0.626	0.608	0.600	0.586	0.532	0.572	0.567	0.526	0.428	0.526
2017	0.617	0.570	0.567	0.532	0.534	0.567	0.587	0.528	0.435	0.517
2018	0.634	0.606	0.585	0.566	0.559	0.611	0.594	0.527	0.449	0.526
2019	0.651	0.621	0.601	0.536	0.536	0.590	0.605	0.523	0.459	0.526
2020	0.633	0.582	0.574	0.553	0.533	0.548	0.563	0.540	0.434	0.543
2021	0.637	0.584	0.577	0.555	0.531	0.565	0.571	0.549	0.492	0.546
2022	0.646	0.632	0.593	0.554	0.504	0.581	0.591	0.541	0.433	0.556

Table 5.10: **Accuracy@8** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.648	0.643	0.639	0.632	0.626	0.617	0.613	0.613	0.605	0.600
2014	0.657	0.650	0.647	0.639	0.635	0.627	0.622	0.620	0.613	0.608
2015	0.651	0.645	0.642	0.636	0.633	0.624	0.619	0.619	0.612	0.608
2016	0.652	0.646	0.643	0.637	0.631	0.621	0.616	0.615	0.610	0.605
2017	0.660	0.655	0.652	0.646	0.640	0.633	0.628	0.628	0.621	0.618
2018	0.656	0.651	0.647	0.642	0.636	0.627	0.624	0.622	0.614	0.611
2019	0.662	0.658	0.653	0.646	0.642	0.633	0.630	0.630	0.622	0.618
2020	0.652	0.647	0.644	0.636	0.632	0.622	0.619	0.619	0.612	0.608
2021	0.655	0.650	0.648	0.641	0.635	0.627	0.624	0.622	0.615	0.611
2022	0.657	0.651	0.647	0.641	0.637	0.630	0.625	0.625	0.619	0.614

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.748	0.690	0.686	0.690	0.648	0.647	0.676	0.627	0.526	0.612
2014	0.761	0.730	0.691	0.681	0.661	0.670	0.706	0.641	0.522	0.625
2015	0.727	0.661	0.677	0.660	0.606	0.629	0.664	0.610	0.530	0.589
2016	0.746	0.701	0.712	0.701	0.647	0.662	0.670	0.621	0.514	0.629
2017	0.733	0.681	0.686	0.659	0.666	0.662	0.691	0.637	0.539	0.614
2018	0.759	0.701	0.697	0.670	0.665	0.705	0.694	0.643	0.539	0.624
2019	0.761	0.714	0.702	0.673	0.656	0.690	0.696	0.633	0.559	0.634
2020	0.746	0.683	0.678	0.677	0.632	0.654	0.661	0.650	0.538	0.640
2021	0.750	0.689	0.687	0.670	0.636	0.667	0.667	0.650	0.582	0.648
2022	0.760	0.726	0.692	0.676	0.630	0.672	0.690	0.637	0.536	0.649

Table 5.11: **Accuracy@16** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.723	0.719	0.716	0.710	0.705	0.697	0.693	0.692	0.687	0.682
2014	0.731	0.727	0.723	0.717	0.714	0.706	0.702	0.702	0.695	0.690
2015	0.727	0.723	0.721	0.714	0.710	0.703	0.700	0.699	0.693	0.688
2016	0.726	0.721	0.719	0.713	0.709	0.700	0.696	0.696	0.692	0.687
2017	0.734	0.730	0.726	0.722	0.718	0.710	0.706	0.706	0.701	0.696
2018	0.732	0.727	0.724	0.719	0.714	0.707	0.702	0.701	0.697	0.693
2019	0.736	0.731	0.727	0.723	0.718	0.711	0.708	0.707	0.703	0.698
2020	0.727	0.722	0.719	0.714	0.711	0.703	0.699	0.699	0.693	0.689
2021	0.728	0.724	0.721	0.715	0.712	0.705	0.702	0.701	0.696	0.691
2022	0.730	0.726	0.723	0.717	0.714	0.707	0.704	0.703	0.698	0.695

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.839	0.763	0.778	0.763	0.752	0.736	0.763	0.718	0.626	0.686
2014	0.852	0.794	0.791	0.767	0.756	0.765	0.788	0.736	0.635	0.701
2015	0.835	0.756	0.774	0.763	0.711	0.727	0.760	0.706	0.632	0.701
2016	0.848	0.771	0.801	0.779	0.756	0.759	0.765	0.722	0.633	0.709
2017	0.845	0.760	0.788	0.754	0.763	0.747	0.779	0.716	0.638	0.710
2018	0.847	0.776	0.785	0.766	0.760	0.788	0.778	0.735	0.645	0.726
2019	0.856	0.786	0.786	0.764	0.765	0.769	0.785	0.740	0.669	0.713
2020	0.850	0.771	0.775	0.771	0.747	0.751	0.763	0.746	0.642	0.734
2021	0.852	0.771	0.774	0.757	0.734	0.749	0.768	0.743	0.676	0.741
2022	0.852	0.797	0.784	0.759	0.752	0.752	0.780	0.739	0.643	0.733

Table 5.12: **Accuracy@32** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

Continual Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.785	0.782	0.778	0.772	0.769	0.762	0.758	0.758	0.754	0.750
2014	0.792	0.790	0.785	0.781	0.777	0.771	0.767	0.767	0.763	0.760
2015	0.786	0.784	0.782	0.777	0.773	0.769	0.765	0.764	0.760	0.757
2016	0.789	0.784	0.781	0.777	0.773	0.768	0.763	0.763	0.758	0.755
2017	0.794	0.791	0.788	0.785	0.781	0.775	0.771	0.772	0.768	0.763
2018	0.791	0.788	0.786	0.782	0.778	0.773	0.769	0.769	0.764	0.760
2019	0.795	0.792	0.789	0.784	0.781	0.776	0.772	0.773	0.767	0.765
2020	0.787	0.783	0.782	0.777	0.774	0.768	0.765	0.765	0.761	0.756
2021	0.788	0.785	0.782	0.777	0.773	0.769	0.764	0.764	0.761	0.757
2022	0.790	0.787	0.783	0.779	0.776	0.771	0.768	0.768	0.764	0.760

New Entities

Train \ Test	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
2013	0.910	0.819	0.853	0.826	0.841	0.812	0.819	0.791	0.688	0.774
2014	0.908	0.848	0.862	0.827	0.843	0.832	0.842	0.814	0.704	0.791
2015	0.898	0.823	0.849	0.822	0.808	0.813	0.832	0.788	0.706	0.781
2016	0.897	0.832	0.862	0.832	0.839	0.823	0.823	0.802	0.718	0.791
2017	0.906	0.832	0.857	0.817	0.840	0.824	0.835	0.791	0.714	0.808
2018	0.908	0.835	0.858	0.830	0.846	0.853	0.835	0.806	0.728	0.803
2019	0.910	0.842	0.853	0.821	0.842	0.843	0.841	0.810	0.734	0.799
2020	0.903	0.828	0.844	0.835	0.843	0.819	0.833	0.817	0.728	0.811
2021	0.910	0.825	0.852	0.825	0.837	0.817	0.830	0.814	0.761	0.812
2022	0.905	0.846	0.852	0.820	0.830	0.830	0.832	0.808	0.732	0.823

Table 5.13: **Accuracy@64** for *continual* (top) and *new* (bottom) entities. The intensity of colors is set on a row-by-row basis and indicates whether performance is **better** or **worse** compared to the year the model was finetuned on (i.e., the values that form the white diagonal).

6

Conclusions and Future Research

We outline the main conclusions for each of the presented chapters in the current thesis. Additionally, we discuss possible future research directions that can address some of the limitations of the current work.

⋆⋆⋆

6.1 Conclusions

6.1.1 DWIE: an entity-centric dataset for multi-task document-level information extraction

In Chapter 2 we introduce DWIE, a manually annotated multi-task dataset that comprises named entity recognition, coreference resolution, relation extraction and entity linking as the main tasks. We highlight how DWIE differs from the mainstream datasets by focusing on document-level and entity-centric annotations. This also makes the predictions on this dataset more challenging by having not only to consider explicit, but also implicit document-level interactions between entities. Furthermore, we show how Graph Neural Networks can help to tackle this issue by propagating local contextual mention span information on a document level for a single task as well as across the tasks on the DWIE dataset. We experiment with known graph propagation techniques driven by the scores of the coreference resolution (`CorefProp`) and relation extraction (`RelProp`) components, as well as introduce a new latent task-independent attention-based graph propagation method (`AttProp`). We demonstrate that, without relying on the task-specific

scorers, `AttProp` can boost the performance of single-task as well as joint models, performing on par and even outperforming significantly in some scenarios the `RelProp` and `CorefProp` graph propagations. Furthermore, our experimental results show complementarity between some of the evaluated IE tasks, with superior performance when using *joint* model compared to independently trained *single* models.

6.1.2 Towards consistent document-level entity linking: joint models for entity linking and coreference resolution

In Chapter 3, we propose two end-to-end models to solve entity linking and coreference resolution tasks in a joint setting. Both of our joint architectures are characterized by formulating EL+coref as a single, structurally constrained task. This contrasts with previous attempts to join coref+EL tasks [1–3], where both of the models are trained separately and additional logic is required to merge the predictions of coref and EL tasks. It further contrasts with the joint architecture proposed in Chapter 2, where the loss function is composed of a weighted linear combination of multiple losses, each one corresponding to a particular task. This multi-task approach presents additional challenges when defining the weights in order to normalize and avoid interference between each of the task's gradients [4–9]. Conversely, both of our proposed models allow to efficiently compute the *exact* log-likelihood loss of the joint EL+coref target by marginalization over all possible configurations (e.g., all possible spanning trees of our *global* model). Our joint architectures achieve superior performance compared to the standalone counterparts on both coreference and entity linking tasks. Further analysis reveals that this boost in performance is driven by more coherent predictions on the level of mention clusters (linking to the same entity) and extended candidate entity coverage.

6.1.3 Injecting knowledge base information into end-to-end joint entity and relation extraction and coreference resolution

In Chapter 4, we propose an end-to-end model for joint IE (NER + relation extraction + coreference resolution) incorporating entity representations from a background knowledge base (KB) in a span-based model. We find that representations built from a knowledge graph and a hypertext corpus are complementary in boosting IE performance. Concretely, when using both entity embeddings from the textual Wikipedia [10] and entity representations derived from the Wikidata Knowledge Graph [11], we observe an improvement of performance compared to when incorporating each of these representations separately. To combine these candidate entity representations for text spans, we explore various weighting schemes: (i) a uniform average of candidate entities (*Uniform*), (ii) the prior weights of each of the candidate entities (*Prior*), (iii) an attention scheme (*Attention*), or (iv) attention with prior information (*AttPrior*). Our experimental results show a superior performance when applying the *AttPrior* scheme, showcasing the complementary effect of combining prior frequency information from a hypertext corpus with contextual information to identify the relevant entities.

6.1.4 Temporal entity linking

In Chapter 5 we introduced TempEL, a first large-scale temporal entity linking dataset composed of 10 yearly snapshots of Wikipedia target entities linked to by anchor mentions. We divided this dataset into mentions linked to *continual* (existing in all the temporal snapshots) and *new* (new to a particular snapshot) target entities. We described the dataset creation pipeline, putting special focus on the quality assurance and future extensibility of TempEL. Our preliminary analysis of the TempEL showcases a decrease in Jaccard similarity of entity definition as well as the context of mentions linked to a specific entity. This demonstrates the dynamic and evolving nature of TempEL, affected by changes in (i) target entity definitions, and (ii) the anchor mentions linked to those target entities. Furthermore, we experiment with the bi-encoder baseline model and showcase a consistent temporal deterioration in performance of entity linking task. We conclude that such a decrease in performance is particularly affected by the temporally increasing number of (ever more granular) candidate entities in Wikipedia. We further examined the most challenging cases for this model, concluding the critical aspect of new knowledge acquisition during the pre-training phase in order to successfully disambiguate *new* entities.

6.2 Future directions

Even though the presented work has extended the knowledge in multiple areas of information extraction, we have only scratched the surface of potential research avenues to be explored. In this section we aim to provide a brief description of such future research directions that could complement or extend the methodologies introduced in this thesis.

On extending coreference annotations: One future direction consists in extending the coreference annotations to include nominal and anaphoric expressions. This will challenge and open new perspective into studying the complementary relation between entity linking and coreference component described in Chapter 3. Furthermore, we expect that including these diverse mention types (whose initial span embedding representation can be different from coreferenced named entities), will allow to investigate further the potential benefits of using joint entity-centric models such as the ones explored in Chapter 2 and Chapter 3.

Effect of pre-training on new corpora: Recent work has demonstrated the benefits of pre-training language models on more recent corpora (e.g., the latest Wikipedia versions) when applied on the downstream tasks [12, 13]. Yet, in our work described in Chapter 5, we fine-tune existing pre-trained BERT based models on downstream task of entity linking. We hypothesize that pre-training BERT from scratch on newer versions of Wikipedia can boost the performance on TempEL introduced in Chapter 5, specially on *new* entities that often require additional knowledge to be correctly disambiguated.

Temporal changes in mention context: the work in Chapter 5 focused mostly on changes in target entities, leaving unexplored a study on how changes in mention context affect EL task. However, Fig. 5.3c in Chapter 5 showcases a big drop in Jaccard vocabulary similarity of context around mentions, specially compared to the immediately preceding year. This suggests that the mentions, as well as the text

surrounding them, are highly volatile and evolving with time, making the temporal mention evolution an interesting subject for future research. Concretely, our hypothesis is that the temporal performance drift in entity linking task is not only affected by changes in the target entities, but also by changes in the mentions linked to these entities.

Cross-lingual entity information: the work described in this thesis is limited to entity definitions in English Wikipedia. Yet, recent research [14, 15] has shown the benefits of training entity linking models in a cross-lingual setting. We hypothesize that this setting can also be applied to study the evolution of entity linking task introduced in Chapter 5 of this thesis. This can be achieved by adapting the TempEL creation framework introduced in Chapter 5 to process the history of Wikipedias in other languages than English and extract the respective anchor mentions and target entities.

KB-driven entity linking: in the current thesis we have covered entity linking constrained to anchor mentions with specific characteristics. For example, in Chapters 2–4, we focused on linking proper nouns (i.e., named entities). Yet, we envision a complete entity linking approach driven exclusively by the entities in a particular Knowledge Base and not by some characteristics of anchor mentions. We argue this setup would enrich the connection between the text and KB, allowing to study in greater detail the effects of entity-centric reasoning and transfer of knowledge in models such as the ones developed in Chapters 2–4 of this thesis. Recent work in this direction has been limited to domain-specific entity linking datasets such as MedMentions [16] in the biomedical domain. Each of the possible textual mention spans in MedMentions was carefully examined by professionals with experience in biomedical content to find the corresponding match in UMLS [17] biomedical knowledge base. As a result, MedMentions presents densely annotated documents with entity links driven by the content of the target UMLS KB. More recently, the authors of [18] propose a fine-grained entity linking annotation scheme, including mentions in grammatical categories such as adjectives, verbs and adverbs. The authors further extend three entity linking datasets (VoxEL [19], KORE50 [20], and ACE2004 [21]) using the proposed annotation scheme. Experimental results on these datasets show very low recall of state-of-the-art models, indicating their limitation in identifying the set of all mentions to be linked. Yet, despite these interesting findings, the annotated datasets are small (ranging between 1 and 20 documents each), which makes them impractical to train models. This research gap is exacerbated by a lack of KB-driven entity linking annotations on *multi-task* IE datasets such as DWIE.

Cross-document IE: in Chapters 2 – 4 we focused in document-level entity-centric annotations. We think that the next natural step is to extend entity-centric information extraction (IE) approach to cross-document setting. This setting is characterized by IE annotations, such as coreferent mentions for coreference resolution task, created on a set of multiple related documents (i.e., cross-document). This contrasts with DWIE introduced in Chapter 2, where all the structured annotations are made on a single document, which represents a specific news article. Most of the related work on cross-document IE has focused on *coreference resolution* task [22–33]. This task consists in identifying coreferent mentions on a set of documents given as input. More recently, there has been an ever-growing interest in other cross-document tasks such as entity linking [34] and relation extraction [35]. Yet, to the best of our knowledge, there is still a research gap in integrating the dif-

ferent cross-document IE annotations in a single multi-task dataset. Such a dataset would encourage the research in creating joint IE models to efficiently process the information in multiple documents. This setting involving multiple documents as input can be challenging to tackle with current state-of-the-art BERT-driven IE models that can only process a very limited context. Fortunately, recent work in efficient transformers such as Longformer [36] and BigBird [37], among others [38], is promising to be applied to tackle this problem. This is the case of the recently introduced Longformer-based cross-document language model (CDLM) [39] that achieves state-of-the-art results in the cross-document coreference resolution IE task.

Harnessing and editing knowledge in language models, recent work using BERT-based language models (LMs) [40–42] has advanced the state-of-the-art in information extraction (IE). For instance, span-based IE models [43–47] use contextualized BERT embeddings as input to generate span representations. Furthermore, recent dense passage retrieval models [48] have successfully used BERT encoders as retriever components in Information Extraction tasks such as entity linking [49–51] and slot filling [52]. More recently, autoregressive and generative models [41, 52–56] have advanced the state-of-the-art by generating directly the information stored in LMs for IE tasks such as entity linking [15, 53, 57–59], relation extraction [60–62], event prediction [63], argument extraction [64] and slot filling [65, 66]. Finally, recent advances in prompt engineering [67] go one step further by explicitly probing the language models for answers. Such prompt-driven models have been applied in a few-shot setting for a number of IE tasks such as named entity recognition [68, 69], relation extraction [70–74] and event argument extraction [75, 76]. Yet, the performance of these language models is highly dependent on the knowledge stored in their internal structure [77–79]. This is also suggested by the drop in performance in Chapter 5 for our BERT-based bi-encoder on new entities that require additional knowledge related to COVID-19 pandemic. In order to tackle this issue, future work should focus on exploring efficient mechanisms, ideally working with one-shot exposure to new knowledge in KBs in order to inject new facts in the pre-trained language models. Recent work [13, 80–86] demonstrates that knowledge injection methods such as continual pre-training and hypernetworks can indeed improve the performance on various IE tasks. Yet, such mechanisms can be further improved, particularly in the number of training examples the model has to be exposed to in order to be able to effectively incorporate (or learn to incorporate) new knowledge.

Entity-centric semantic frames, we use the concept *semantic frames* to refer to preliminarly defined structure interconnecting multiple entities such as lexically driven FrameNet [87] as well as more general *n-ary relations* [88–94] and *events* [95–97]. For example, we can envision a semantic frame *marriage* interconnecting entities related to both *partners*, *date*, *place*, etc. Unlike relations, the number of linked entities in semantic frames is not restricted to 2 (head and tail). The current structure of most of these frames is mention-driven. We believe the design and annotation of entity-centric semantic frames (i.e., frames where each slot is connected to conceptual entities, instead of entity mentions), would allow to abstract from individual mentions and focus on entities, some of which might even not be explicitly mentioned in a document. Following this direction, one of the possible future works could consist in annotating such semantic frames as an additional task in the intro-

duced multi-task DWIE dataset (see Chapter 2).

References

[1] H. Hajishirzi, L. Zilles, D. S. Weld, and L. Zettlemoyer. *Joint coreference resolution and named-entity linking with multi-pass sieves*. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), pages 289–299, 2013. Available from: https://aclanthology.org/D13-1029/ .

[2] S. Dutta and G. Weikum. *C3EL: A joint model for cross-document co-reference resolution and entity linking*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pages 846–856, 2015. Available from: https://doi.org/10.18653/v1/d15-1101.

[3] R. Angell, N. Monath, S. Mohan, N. Yadav, and A. McCallum. *Clustering-based Inference for Biomedical Entity Linking*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pages 2598–2608, 2021. Available from: https://doi.org/10.18653/v1/2021.naacl-main.205.

[4] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. *Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks*. In International conference on machine learning, pages 794–803. PMLR, 2018.

[5] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. *Multi-task learning for dense prediction tasks: A survey*. IEEE transactions on pattern analysis and machine intelligence, 2021.

[6] Y. Zhang and Q. Yang. *A survey on multi-task learning*. IEEE Transactions on Knowledge and Data Engineering, 2021.

[7] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. *Gradient surgery for multi-task learning*. Advances in Neural Information Processing Systems, 33:5824–5836, 2020.

[8] Z. Chen, J. Ngiam, Y. Huang, T. Luong, H. Kretzschmar, Y. Chai, and D. Anguelov. *Just pick a sign: Optimizing deep multitask models with gradient sign dropout*. Advances in Neural Information Processing Systems, 33:2039–2050, 2020.

[9] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu. *Conflict-averse gradient descent for multi-task learning*. Advances in Neural Information Processing Systems, 34:18878–18890, 2021.

[10] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. *Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation*. In Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016), pages 250–259, 2016. Available from: https://doi.org/10.18653/v1/k16-1025.

[11] A. Joulin, E. Grave, P. Bojanowski, M. Nickel, and T. Mikolov. *Fast linear model for knowledge graph embeddings*. arXiv:1710.10881, 2017. Available from: http://arxiv.org/abs/1710.10881.

[12] O. Agarwal and A. Nenkova. *Temporal Effects on Pre-trained Models for Language Processing Tasks*. CoRR, 2021. Available from: https://arxiv.org/abs/2111.12790.

[13] D. Loureiro, F. Barbieri, L. Neves, L. E. Anke, and J. Camacho-Collados. *TimeLMs: Diachronic Language Models from Twitter*. In Proceedings of the 2022 Annual Meeting of the Association for Computational Linguistics (ACL 2022), pages 251–260, 2022. Available from: https://aclanthology.org/2022.acl-demo.25.

[14] J. A. Botha, Z. Shan, and D. Gillick. *Entity Linking in 100 Languages*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 7833–7845, 2020. Available from: https://www.aclweb.org/anthology/2020.emnlp-main.630.

[15] N. De Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, and F. Petroni. *Multilingual Autoregressive Entity Linking*. Transactions of the Association for Computational Linguistics, 10:274–290, 2022. Available from: https://arxiv.org/abs/2103.12528.

[16] S. Mohan and D. Li. *MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts*. In Proceedings of the 2018 Automated Knowledge Base Construction (AKBC 2018), 2018. Available from: https://doi.org/10.24432/C5G59C.

[17] O. Bodenreider. *The unified medical language system (UMLS): integrating biomedical terminology*. Nucleic acids research, 32(suppl_1):D267–D270, 2004.

[18] H. Rosales-Méndez, A. Hogan, and B. Poblete. *Fine-Grained Entity Linking*. Journal of Web Semantics, 65:100600, 2020.

[19] H. Rosales-Méndez, A. Hogan, and B. Poblete. *VoxEL: a benchmark dataset for multilingual entity linking*. In Proceedings of the 2018 International Semantic Web Conference (ISWC 2018), pages 170–186, 2018. Available from: https://doi.org/10.1007/978-3-030-00668-6_11.

[20] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. *KORE: keyphrase overlap relatedness for entity disambiguation*. In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 545–554, 2012.

[21] L. Ratinov, D. Roth, D. Downey, and M. Anderson. *Local and global algorithms for disambiguation to Wikipedia*. In Proceedings of the 2011 Annual Meeting of the Association for Computational Linguistics (ACL 2011), pages 1375–1384, 2011. Available from: https://aclanthology.org/P11-1138/.

[22] A. Bagga and B. Baldwin. *Entity-Based Cross-Document Core f erencing Using the Vector Space Model*. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, pages 79–85, 1998.

[23] R. L. Logan IV, A. McCallum, S. Singh, and D. Bikel. *Benchmarking scalable methods for streaming cross document entity coreference*. 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 4717–4731, 2021.

[24] B. Hsu and G. Horwood. *Contrastive Representation Learning for Cross-Document Coreference Resolution of Events and Entities*. arXiv preprint arXiv:2205.11438, 2022.

[25] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan. *Realistic Evaluation Principles for Cross-document Coreference Resolution*. In Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, pages 143–151, 2021.

[26] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan. *Cross-document Coreference Resolution over Predicted Mentions*. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5100–5107, 2021.

[27] A. Cattan, S. Johnson, D. Weld, I. Dagan, I. Beltagy, D. Downey, and T. Hope. *Scico: Hierarchical cross-document coreference for scientific concepts*. arXiv preprint arXiv:2104.08809, 2021.

[28] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan. *Streamlining Cross-Document Coreference Resolution: Evaluation and Modeling*. arXiv preprint arXiv:2009.11032, 2020.

[29] C. H. Gooi and J. Allan. *Cross-document coreference on a large scale corpus*. Technical report, MASSACHUSETTS UNIV AMHERST CENTER FOR INTELLIGENT INFORMATION RETRIEVAL, 2004.

[30] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. *Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 793–803, 2011.

[31] S. Barhom, V. Shwartz, A. Eirew, M. Bugert, N. Reimers, and I. Dagan. *Revisiting Joint Modeling of Cross-document Entity and Event Coreference Resolution*. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4179–4189, 2019.

[32] A. Caciularu, A. Cohan, I. Beltagy, M. E. Peters, A. Cattan, and I. Dagan. *Cross-document language modeling*. arXiv preprint arXiv:2101.00406, 2021.

[33] J. Ravenscroft, A. Cattan, A. Clare, I. Dagan, and M. Liakata. *CD2CR: Co-reference Resolution Across Documents and Domains*. arXiv preprint arXiv:2101.12637, 2021.

[34] D. Agarwal, R. Angell, N. Monath, and A. McCallum. *Entity Linking and Discovery via Arborescence-based Supervised Clustering*. arXiv preprint arXiv:2109.01242, 2021. Available from: https://arxiv.org/abs/2109.01242.

[35] Y. Yao, J. Du, Y. Lin, P. Li, Z. Liu, J. Zhou, and M. Sun. *CodRED: A Cross-Document Relation Extraction Dataset for Acquiring Knowledge in the Wild*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4452–4472, 2021.

[36] I. Beltagy, M. E. Peters, and A. Cohan. *Longformer: The long-document transformer*. arXiv preprint arXiv:2004.05150, 2020.

[37] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. *Big Bird: Transformers for Longer Sequences.* In NeurIPS, 2020.

[38] M. Treviso, T. Ji, J.-U. Lee, B. van Aken, Q. Cao, M. R. Ciosici, M. Hassid, K. Heafield, S. Hooker, P. H. Martins, et al. *Efficient Methods for Natural Language Processing: A Survey*. arXiv preprint arXiv:2209.00099, 2022.

[39] A. Caciularu, A. Cohan, I. Beltagy, M. E. Peters, A. Cattan, and I. Dagan. *CDLM: Cross-Document Language Modeling*. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2648–2662, 2021.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pages 4171–4186, 2019. Available from: https://www.aclweb.org/anthology/N19-1423.

[41] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. *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, 2020.

[42] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. *Language models are few-shot learners*. arXiv preprint arXiv:2005.14165, 2020.

[43] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi. *Entity, Relation, and Event Extraction with Contextualized Span Representations*. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pages 5788–5793, 2019.

[44] M. Eberts and A. Ulges. *Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training*. In ECAI 2020, pages 2006–2013. IOS Press, 2020.

[45] Z. Zhong and D. Chen. *A Frustratingly Easy Approach for Entity and Relation Extraction*. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 50–61, 2021.

[46] Y. Lin, H. Ji, F. Huang, and L. Wu. *A Joint Neural Model for Information Extraction with Global Features*. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7999–8009, 2020.

[47] J. Wang and W. Lu. *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, 2020.

[48] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. *Dense Passage Retrieval for Open-Domain Question Answering*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6769–6781, 2020. Available from: https://aclanthology.org/2020.emnlp-main.550.

[49] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. *Zero-shot entity linking with dense entity retrieval*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pages 6397–6407, 2020. Available from: https://aclanthology.org/2020.emnlp-main.519.

[50] W. Zhang, W. Hua, and K. Stratos. *EntQA: Entity Linking as Question Answering*. In Proceedings of the 2022 International Conference on Learning Representations (ICLR 2022), 2022. Available from: https://arxiv.org/abs/2110.02369.

[51] B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W.-t. Yih. *Efficient One-Pass End-to-End Entity Linking for Questions*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6433–6441, 2020.

[52] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, et al. *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 (NAACL-HLT 2021), 2021. Available from: https://aclanthology.org/2021.naacl-main.200.

[53] N. De Cao, G. Izacard, S. Riedel, and F. Petroni. *Autoregressive Entity Retrieval*. In Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021), 2021. Available from: https://openreview.net/forum?id=5k8F6UU39V.

[54] F. Petroni, P. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, and S. Riedel. *How Context Affects Language Models' Factual Predictions*. In Automated Knowledge Base Construction, 2020.

[55] S. Rongali, L. Soldaini, E. Monti, and W. Hamza. *Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing*. In Proceedings of The Web Conference 2020, pages 2962–2968, 2020.

[56] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. *Document Ranking with a Pre-trained Sequence-to-Sequence Model*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 708–718, 2020.

[57] N. De Cao, W. Aziz, and I. Titov. *Highly Parallel Autoregressive Entity Linking with Discriminative Correction*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), pages 7662–7669, 2021. Available from: https://aclanthology.org/2021.emnlp-main.604, doi:10.18653/v1/2021.emnlp-main.604.

[58] K. Mrini, S. Nie, J. Gu, S. Wang, M. Sanjabi, and H. Firooz. *Detection, Disambiguation, Re-ranking: Autoregressive Entity Linking as a Multi-Task Problem*. arXiv preprint arXiv:2204.05990, 2022.

[59] H. Yuan, Z. Yuan, and S. Yu. *Generative Biomedical Entity Linking via Knowledge Base-Guided Pre-training and Synonyms-Aware Fine-tuning*. arXiv preprint arXiv:2204.05164, 2022.

[60] P.-L. H. Cabot and R. Navigli. *REBEL: Relation extraction by end-to-end language generation*. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2370–2381, 2021.

[61] M. Josifoski, N. De Cao, M. Peyrard, and R. West. *GenIE: generative information extraction*. arXiv preprint arXiv:2112.08340, 2021.

[62] A. Saxena, A. Kochsiek, and R. Gemulla. *Sequence-to-Sequence Knowledge Graph Completion and Question Answering*. In Proceedings of the 60th Annual Meeting

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2814–2828, 2022.

[63] W. Jin, M. Qu, X. Jin, and X. Ren. *Recurrent event network: Autoregressive structure inference over temporal knowledge graphs*. arXiv preprint arXiv:1904.05530, 2019.

[64] K.-H. Huang, I. Hsu, P. Natarajan, K.-W. Chang, N. Peng, et al. *Multilingual Generative Language Models for Zero-Shot Cross-Lingual Event Argument Extraction*. arXiv preprint arXiv:2203.08308, 2022.

[65] M. Glass, G. Rossiello, M. F. M. Chowdhury, A. R. Naik, P. Cai, and A. Gliozzo. *Re2G: Retrieve, Rerank, Generate*. arXiv preprint arXiv:2207.06300, 2022.

[66] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu. *Unified Structure Generation for Universal Information Extraction*. arXiv preprint arXiv:2203.12277, 2022.

[67] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*. arXiv preprint arXiv:2107.13586, 2021.

[68] R. Ma, X. Zhou, T. Gui, Y. Tan, Q. Zhang, and X. Huang. *Template-free prompt tuning for few-shot NER*. arXiv preprint arXiv:2109.13532, 2021.

[69] A. T. Liu, W. Xiao, H. Zhu, D. Zhang, S.-W. Li, and A. Arnold. *QaNER: Prompting question answering models for few-shot named entity recognition*. arXiv preprint arXiv:2203.01543, 2022.

[70] H. Zhang, B. Liang, M. Yang, H. Wang, and R. Xu. *Prompt-Based Prototypical Framework for Continual Relation Extraction*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022.

[71] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen. *Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction*. In Proceedings of the ACM Web Conference 2022, pages 2778–2788, 2022.

[72] H.-S. Yeh, T. Lavergne, and P. Zweigenbaum. *Decorate the Examples: A Simple Method of Prompt Design for Biomedical Relation Extraction*. arXiv preprint arXiv:2204.10360, 2022.

[73] J. Gong and H. Eldardiry. *Prompt-based Zero-shot Relation Classification with Semantic Knowledge Augmentation*. arXiv preprint arXiv:2112.04539, 2021.

[74] Y. K. Chia, L. Bing, S. Poria, and L. Si. *RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction*. In Findings of the Association for Computational Linguistics: ACL 2022, pages 45–57, 2022.

[75] X. Liu, H.-Y. Huang, G. Shi, and B. Wang. *Dynamic Prefix-Tuning for Generative Template-based Event Extraction*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5216–5228, 2022.

[76] Y. Ma, Z. Wang, Y. Cao, M. Li, M. Chen, K. Wang, and J. Shao. *Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6759–6774, 2022.

[77] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. *Language Models as Knowledge Bases?* In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 2463–2473, 2019. Available from: https://aclanthology.org/D19-1250.

[78] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, and M. Ghazvininejad. *A review on language models as knowledge bases*. arXiv preprint arXiv:2204.06031, 2022.

[79] D. Yin, L. Dong, H. Cheng, X. Liu, K.-W. Chang, F. Wei, and J. Gao. *A survey of knowledge-intensive nlp with pre-trained language models*. arXiv preprint arXiv:2202.08772, 2022.

[80] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, G. Cao, D. Jiang, M. Zhou, et al. *K-adapter: Infusing knowledge into pre-trained models with adapters*. arXiv preprint arXiv:2002.01808, 2020.

[81] N. De Cao, W. Aziz, and I. Titov. *Editing Factual Knowledge in Language Models*. arXiv preprint arXiv:2104.08164, 2021.

[82] A. Sinitsin, V. Plokhotnyuk, D. Pyrkin, S. Popov, and A. Babenko. *Editable Neural Networks*. arXiv preprint arXiv:2004.00345, 2020.

[83] A. Cossu, T. Tuytelaars, A. Carta, L. Passaro, V. Lomonaco, and D. Bacciu. *Continual Pre-Training Mitigates Forgetting in Language and Vision*. arXiv preprint arXiv:2205.09357, 2022.

[84] X. Jin, D. Zhang, H. Zhu, W. Xiao, S.-W. Li, X. Wei, A. Arnold, and X. Ren. *Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora*. In Proceedings of BigScience Episode\# 5–Workshop on Challenges & Perspectives in Creating Large Language Models, pages 1–16, 2022.

[85] C. Zhu, A. S. Rawat, M. Zaheer, S. Bhojanapalli, D. Li, F. Yu, and S. Kumar. *Modifying Memories in Transformer Models*. arXiv preprint arXiv:2012.00363, 2020.

[86] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. *Meta-learning in neural networks: A survey*. IEEE transactions on pattern analysis and machine intelligence, 2021.

[87] C. F. Baker, C. J. Fillmore, and J. B. Lowe. *The berkeley framenet project*. In Proceedings of the 1998 International Conference on Computational Linguistics, pages 86–90, 1998.

[88] R. Jia, C. Wong, and H. Poon. *Document-Level N-ary Relation Extraction with Multiscale Representation Learning*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3693–3704, 2019.

[89] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-t. Yih. *Cross-sentence n-ary relation extraction with graph lstms*. Transactions of the Association for Computational Linguistics, 5:101–115, 2017.

[90] L. Song, Y. Zhang, Z. Wang, and D. Gildea. *N-ary Relation Extraction using Graph-State LSTM*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2226–2235, 2018.

[91] M. Giunti, G. Sergioli, G. Vivanet, and S. Pinna. *Representing n-ary relations in the Semantic Web*. Logic Journal of the IGPL, 29(4):697–717, 2021.

[92] M. Lentschat, P. Buche, J. Dibie-Barthelemy, and M. Roche. *A new method to extract n-Ary relation instances from scientific documents*. Expert Systems with Applications, page 118332, 2022.

[93] P.-T. Lai and Z. Lu. *BERT-GT: cross-sentence n-ary relation extraction with BERT and Graph Transformer*. Bioinformatics, 36(24):5678–5685, 2020.

[94] O. Lehmberg and C. Bizer. *Synthesizing n-ary relations from web tables*. In Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics, pages 1–12, 2019.

[95] W. Xiang and B. Wang. *A survey of event extraction from text*. IEEE Access, 7:173111–173137, 2019.

[96] F. Hogenboom, F. Frasincar, U. Kaymak, F. De Jong, and E. Caron. *A survey of event extraction methods from text for decision support systems*. Decision Support Systems, 85:12–22, 2016.

[97] L. Zhan and X. Jiang. *Survey on event extraction technology in information extraction research area*. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pages 2121–2126. IEEE, 2019.

A

Solving Arithmetic Word Problems by Scoring Equations with Recursive Neural Networks

In this chapter we propose to use recursive neural networks to mimic the structure on equation trees to solve mathematical world problems. We showcase a significant improvement using our approach.

⋆ ⋆ ⋆

K. Zaporojets, G. Bekoulis, J. Deleu, C. Develder and T. Demeester

Expert Systems with Applications, 2021.

Abstract Solving arithmetic word problems is a cornerstone task in assessing language understanding and reasoning capabilities in NLP systems. Recent works use automatic extraction and ranking of candidate solution equations providing the answer to arithmetic word problems. In this work, we explore novel approaches to score such candidate solution equations using tree-structured recursive neural network (Tree-RNN) configurations. The advantage of this Tree-RNN approach over using more established sequential representations, is that it can naturally capture the structure of the equations. Our proposed method consists of transforming the mathematical expression of the equation into an expression tree. Further, we encode this tree into a Tree-RNN by using different Tree-LSTM architectures. Experimental results show that our proposed method (i) improves overall performance

with more than 3% accuracy points compared to previous state-of-the-art, and with over 15% points on a subset of problems that require more complex reasoning, and (ii) outperforms sequential LSTMs by 4% accuracy points on such more complex problems.

A.1 Introduction

Natural language understanding often requires the ability to comprehend and reason with expressions involving numbers. This has produced a recent rise in interest to build applications to automatically solve math word problems [1–5]. These math problems consist of a textual description comprising numbers with a question that will guide the reasoning process to get the numerical solution (see Fig. A.1 for an example). This is a complex task because of (i) the large output space of the possible equations representing a given math problem, and (ii) reasoning required to understand the problem.

The research community has focused in solving mainly two types of mathematical word problems: *arithmetic word problems* [3, 6–9] and *algebraic word problems* [1, 10–12]. Arithmetic word problems can be solved using basic mathematical operations ($+, -, \times, \div$) and involve a single unknown variable. Algebraic word problems, on the other hand, involve more complex operators such as square root, exponential and logarithm with multiple unknown variables. In this work, we focus on solving *arithmetic word problems* such as the one illustrated in Fig. A.1. This figure illustrates (a) *arithmetic word problem* statement, (b) the arithmetical formula of the *solution* to the problem, and (c) the *expression tree* representation of the solution formula where the leaves are connected to quantities and internal nodes represent operations.

The main idea of this paper is to explore the use of tree-based Recursive Neural Networks (Tree-RNNs) to encode and score the expression tree (illustrated in Fig. A.1(c) that represents a candidate arithmetic expression of a specific arithmetic word problem). This contrasts with predominantly sequential neural representations [7, 9, 13] that encode the problem statement from left to right or vice versa. By using Tree-RNN architectures, we can naturally embed the equation inside a tree structure such that the link structure directly reflects the various mathematical operations

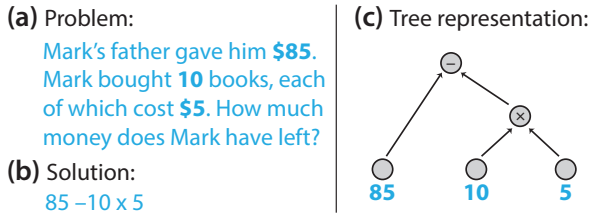


Figure A.1: An example of arithmetic word problem from the SingleEQ dataset. It illustrates the (a) *an arithmetic word problem* statement, (b) the respective *solution* formula, and (c) the *expression tree* representing the solution.

between operands selected from the sequential textual input. We hypothesize that this structured approach can efficiently capture the semantic representations of the candidate equations to solve more complex arithmetic problems involving multiple and/or non-commutative operators. To test our results, we use the recently introduced SingleEQ dataset [2]. It contains a collection of 508 arithmetic word problems with varying degrees of complexity. This allows us to track the performance of the evaluated systems on subsets that require different reasoning capabilities. More concretely, we subdivide the initial dataset into different subsets of varying reasoning complexity (i.e., based on the number of operators, commutative (symmetric) or non-commutative (asymmetric) operations), to investigate whether the performance of the proposed architecture remains consistent across problems of increasing complexity.

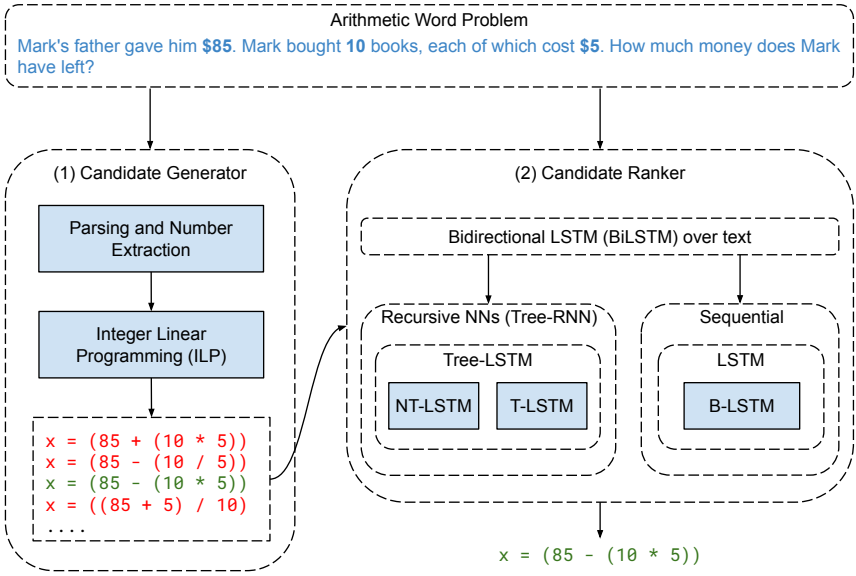


Figure A.2: High-level conceptual view of the arithmetic word problem architecture used throughout the paper. It consists of two main components: (1) *candidate generator* responsible for generating candidate equations to solve a particular *arithmetic word problem*, and (2) *candidate ranker*, for selecting the best candidate from the list provided by *candidate generator*, using the models NT-LSTM, T-LSTM, or B-LSTM.

Figure A.2 provides a high-level conceptual view of the interconnection between the main components of our proposed system. The processing flow consists of two main steps. In the first step, we use the *candidate generator* to generate a list of potential candidate equations for solving a particular *arithmetic word problem*. To achieve this, we employ the Integer Linear Programming (ILP) constraint optimization component proposed by [2] (see Section A.3.1). In the second step, the candidate equations are ranked by the *candidate ranker*, and the equation with

the highest score is chosen as the solution to the processed *arithmetic word problem* (see Section A.3.2). In this paper, we focus on this second step by exploring the impact of structural Tree-RNN-based and sequential Long Short Term Memory-based (LSTM; [14]) candidate equation encoding methods. More specifically, we define two Tree-RNN models inspired by the work of [15] on Tree-LSTM models: (i) T-LSTM (Child-Sum Tree-LSTM), and (ii) NT-LSTM (N-ary Tree-LSTM). In the rest of the manuscript we refer to the general tree-structured architecture of these models as Tree-LSTM. The main difference between the two is that, while in T-LSTM the child node representations are summed up, in NT-LSTM they are concatenated. Unlike the representation used in [15], where the input is given by the word embeddings, our Tree-LSTM models also take as input the operation embeddings (in inner nodes) that represent each of the arithmetic operators ($-$, $+$, $\div$, $\times$). This allows our architecture to distinguish between different operators that are contained in a particular expression tree. We show that NT-LSTM is more suitable to deal with equations that involve non-commutative operators because this architecture is able to capture the order of the operands. We also compare our Tree-LSTM models with a sequential LSTM model which we call B-LSTM. All the models (T-LSTM, NT-LSTM, and B-LSTM) take as input the contextualized representation of the numbers in text produced by a bidirectional LSTM layer (BiLSTM) (see Section A.3.2 for details). After conducting a thorough multi-fold experimentation phase involving multiple random weight re-initializations in order to ensure the validity of our results, we will show that the main added value of our Tree-LSTM-based models compared to state-of-the-art methods lays in an increased performance for more complex arithmetic word problems.

More concretely, our contribution is three-fold: (i) we propose using Tree-LSTMs for solving arithmetic word problems, to embed structural information of the equation, (ii) we compare it against a strong neural baseline model (B-LSTM) that relies on sequential LSTMs, and (iii) we perform an extensive experimental study on the SingleEQ dataset, showing that our Tree-LSTM model achieves an overall accuracy improvement of 3%, including an increase $>$15% for more complex problems (i.e., requiring multiple and non-commutative operations), compared to previous state-of-the-art results.

A.2 Related work

Over the last few years, there has been an increasing interest in building systems to solve *arithmetic word problems*. The adopted approaches can be grouped in three main categories: (i) Rule-based systems, (ii) Statistical systems, and (iii) Neural network systems.

Rule-based systems: The first attempts to solve arithmetic problems date back to the 1960s with the work by [16], who proposed and implemented STUDENT, a rule-based parsing system to extract numbers and operations between them by using pattern matching techniques. [17, 18] extended STUDENT by including basic coreference resolution and capability to work with rate expressions (e.g., "kms per hour"). On the other hand, [19] designed and implemented a system that given a propositional representation of a math problem[1], applies a set of rules to calcu-

[1]With propositions such as *GIVE Y X P9*, where entity Y gives to entity X the object defined

late the final solution. The disadvantage of this system is that it needs a parsed propositional representation of a problem as input and cannot operate directly on raw text. This issue was tackled by [20], who developed a schema-based system that consisted of six main reasoning schemas, each one with slots to fill in. After instantiating the schemas for a particular math problem using lexical verb-based rules, the system could derive the corresponding mathematical equation to solve the problem.

The main disadvantages of such rule-based approaches are that they (i) rely on hard-coded lexico-grammar rules, and (ii) lack an integrated view of the problem to be solved, extracting operations one by one. We address these issues by proposing a model that integrates the mathematical representation of a problem in a single structured expression tree. This way, we are able to capture the operator-operator and number-operator relations involved in a particular mathematical expression in a unified manner. Furthermore, we avoid the use of lexico-grammar hard-coded rules (e.g., the use of pattern-based matching) when connecting numbers with the operators, replacing them by composition-semantic representations that link the arithmetic operations with parameters (numbers or other operations) in a recursive tree. Consequently, our solution is more generalizable by not depending on explicit hand-crafted logic.

Statistical systems: Recently, there has been a shift towards statistical feature-driven systems that automatically produce models by capturing patterns present in arithmetic word problem datasets. For example, [6] presented an inductive model that links specific lexicon-based features (e.g., verb categories) to equation operators. The mathematical solution to the problem is built sequentially using state transitions related to operators that are triggered by different verb categories found in the problem statement. On the other hand, [3] connected carefully designed features to equation templates in order to solve specific problem types. While these techniques produced competitive results, they were limited to addition ($+$) and subtraction ($-$) operations on a very narrow problem set domain. In order to solve more diverse types of problems that also involve multiplication and division operators, the community shifted towards more integrated approaches involving tree structure representations. [2] proposed to rank candidate expression trees by training jointly a *local* model to link spans of text with operator tree nodes, and a *global* model that is used to score the consistency of an entire tree. The list of candidates to these two models is generated by an ILP constraint optimization component that, given a set of extracted numbers from a arithmetic word problem text as input, produces a set of candidate solution equations. Conversely, [21, 22] introduced the concept of *monotonic expression tree* to generate candidates. It defines a set of conditions (e.g., two division and subtraction nodes cannot be connected to each other) that considerably restricts the expression tree search space. The authors propose to score the resulting monotonic expression trees jointly by summing up the scores of different classifiers related to a specific expression tree (e.g., the mathematical operator between two numbers in the tree, whether a particular number is related to a rate such as “kms per hour", etc). Recently, the same authors [23] included additional latent declarative rules (e.g.,

in P9. This proposition in particular can be linked to the first sentence of example in Fig. A.1: “Mark’s father gave him $85", where Y represents “Mark’s father", X represents “him" which is coreferenced to “Mark", and P9 represents “$85" that are being given.

$[\text{Verb1} \in \text{HAVE}] \wedge [\text{Verb2} \in \text{GIVE}] \wedge [\text{Coref}(\text{Subj1}, \text{Subj2})] \implies \text{Subtraction}$) to link textual expression patterns (derived from preliminary dependency parsing) to specific operations. While these statistical approaches rely on tree structures to evaluate the mathematical expressions, on one hand, they require high manual effort to engineer the features and, on the other hand, it is hard to scale the features to capture operations between more than two numbers. This makes it challenging to apply such models to more complex equations that involve multiple operators. We tackle this problem by defining a single Tree-RNN structure that evaluates an entire mathematical expression at once. This is done by recursively combining the information from the child nodes in the expression tree and then using a backpropagation mechanism to correspondingly adjust the weights of our model. Furthermore, our equation ranking architecture does not depend on hand-crafted features and parsing-dependent rules, making it more effective in generalizing across different domains.

Neural network systems: Recently, as in all sub-domains of natural language processing, neural network architectures have been applied to tackle math word problems. The first contribution was made by [7], who introduced a model trained to map problem statements to equation templates. Their model was expanded upon by [24], who introduced an attention-based copy mechanism for tokens representing numbers. They used a reinforcement learning setting, where positive rewards were assigned when the predicted mathematical expression resulted in a correct answer. Recently, [9] used stack structures inside a sequential encoder-decoder setting where the encoder captures the semantics of a math word problem in a vector that is used by decoder to generate the equation to solve the problem. Moreover, [4] proposed the use of Q-Networks in order to generate expression trees, by giving positive reward whenever the operator between two numbers is correct. The aforementioned studies, while showing promising results, were not designed to naturally capture the structural form of mathematical expressions when multiple operators are involved (e.g., $1 + (2/3)$ vs. $(1 + 2)/3$). We propose encoding equations with Tree-LSTMs [15], which are recursive neural sequence models, thus allowing to naturally reflect the execution order of operations in an expression tree by recursively combining the children nodes' semantic representations.

Table A.1 compares our approach (the use of Tree-LSTM-based T-LSTM and NT-LSTM models) with the rest of the methods described in this Section. The main difference of our architecture is that we explore the impact of using tree-based neural encoding (i.e., by means of Tree-LSTM models). We hypothesize that this approach allows to better capture the arithmetic equation structure than the currently predominant neural sequential models [7, 9, 13]. Furthermore, the independence from feature-based and rule-based methods makes our solution more generalizable. This is because our model does not depend on hand-crafted rules or features to capture the patterns of a particular dataset. This aspect will be explored further when comparing the performance of our model to the current feature-based state-of-the-art system [2] in Section A.5.

Tree-RNN models [25] have been shown to perform better for modeling data on tasks that have an inherently hierarchical structure. For example, [25] proposed to use recursive models in order to model the compositional structure of scene images (e.g., a scene image of a house can be split in composing regions such as doors, windows, walls, etc.). The authors show that a Tree-RNN-based architecture outperforms previous methods in prediction of hierarchical structure of scene images

Method	Rules	Features	N-Nets	Tree-Based Representation	Tree-Based Encoding
[16]	✓	–	–	–	–
[17, 18]	✓	–	–	–	–
[19]	✓	–	–	–	–
[20]	✓	–	–	–	–
[6]	–	✓	–	–	–
[2]	✓	✓	–	✓	–
[3]	–	✓	–	–	–
[21, 22]	–	✓	–	✓	–
[7]	–	–	✓	–	–
[23]	✓	✓	–	✓	–
[24]	–	✓	✓	–	–
[4]	–	✓	✓	✓	–
[9]	–	–	✓	–	–
[8]	–	–	✓	–	–
Our Approach (T-LSTM & NT-LSTM)	–	–	✓	✓	✓

Table A.1: Comparison of the various architectures explored in related work. We focus on the following five characteristics: (i) *Rules* indicates whether a rule-based approach is used or not, (ii) *Features* specifies whether the architecture relies on manually engineered features, (iii) *N-Nets* indicates whether artificial neural networks are used or not, (iv) *Tree-Based Representation* groups the models that incorporate information coming from tree structures (e.g., by using trees for feature engineering), and (v) *Tree-Based Encoding* indicates whether the tree structures are used as encoders in a neural network model. The ✓ indicates the presence of a particular characteristic.

and in scene image classification. Later, [26] also showed how recursive structures can be used to encode the inherently hierarchical phrase structural grammar (e.g., the sentence “riding a bike" can be decomposed in the verb “riding" and the noun phrase “a bike", which itself can be decomposed into determiner “a" and the noun “bike"). This way, the authors achieved state-of-the-art performance in grammatical parsing of the sentences. More recently, [15] and [27] showed how encoding the syntactic parsing trees of the sentence with Tree-LSTM models can improve the performance in tasks such as sentiment classification and semantic relatedness (e.g., natural language inference). Similarly, we propose to take advantage of the inherently hierarchical representation of mathematical expression trees by encoding them using Tree-LSTM architectures. Our experiments demonstrate that this representation can be helpful in capturing the semantic relations between operators needed in order to solve more complex arithmetic problems consisting of multiple and/or non-commutative operations.

A.3 Proposed architecture

Shortly stated, our task at hand is to identify the correct arithmetic equation, corresponding to an arithmetic problem expressed in natural language text. We follow a two-step approach similar to the work of [2], which formalizes solving multi-sentence arithmetic word problems as (i) the generation and (ii) ranking of ex-

pression trees. The first step consists of generating candidate equations using the ILP optimization solver proposed in [2] (*candidate generator* component in Fig. A.2). The second step ranks these candidates and selects the top ranked one as the final answer to the arithmetic word problem (*candidate ranker* component in Fig. A.2). We use the rest of this section to provide more insights into the *candidate generator* component in Section A.3.1, and to describe in detail our proposed *candidate ranker* model in Section A.3.2.

A.3.1 Candidate generator

This component is responsible for generating possible candidate equations to solve a given arithmetic word problem. A straightforward solution would be to perform an exhaustive search on all the possible arithmetic expression trees given n extracted numbers from a particular problem. However, the resulting search space would grow exponentially with n, which makes this approach not scalable. In order to deal with this exponential growth in the number of candidates, we re-use the Integer Linear Programming (ILP) solver proposed by [2]. This solver takes as input the extracted numeric quantities with extra attributes derived from syntactic parsing[2], and generates the most promising candidate equations using two types of constraints:

1. *Hard Constraints:* such as the maximum equation length and syntactic validity of equations (e.g., only one unknown allowed, no division by 0, etc.). As a post-processing step, the ILP solver also removes the arithmetic expressions that produce negative or fractional results.
2. *Soft Constraints:* these constraints assign additional weight to candidate equations whose related entity types (extracted from dependency parse tree) are consistent. For example, in the problem of Fig. A.1, the sum (85 + 5) will be prioritized over the sum (5 + 10), because both 85 and 5 refer to the same entity type ("$"), while 10 refers to entity type "books".

To provide a fair comparison between the *candidate ranker* model of ALGES proposed by [2] and our approach (see Section A.3.2), we use both the same constraint configuration, and also consider only the top 100 equations produced by the candidate generator. As in ALGES, we report the coverage as *ILP Coverage* in our results section (see Section A.5). Additionally, we include in our result tables the performance of the *ILP Naive* approach, which consists of selecting the highest scored candidate by the ILP solver. This score allows us to estimate the impact of the *candidate ranker* component.

A.3.2 Candidate ranker

Our proposed candidate ranker model architecture is sketched in Fig. A.3 and comprises: (i) a word embedding layer, (ii) a bidirectional LSTM layer (BiLSTM) over the text, and (iii) an additional layer that encodes the equation, using either BiLSTM (B-LSTM model) or Tree-LSTM (T-LSTM and NT-LSTM models) based approaches, detailed below.

[2]Stanford Dependency Parser in CoreNLP 3.4 is used.

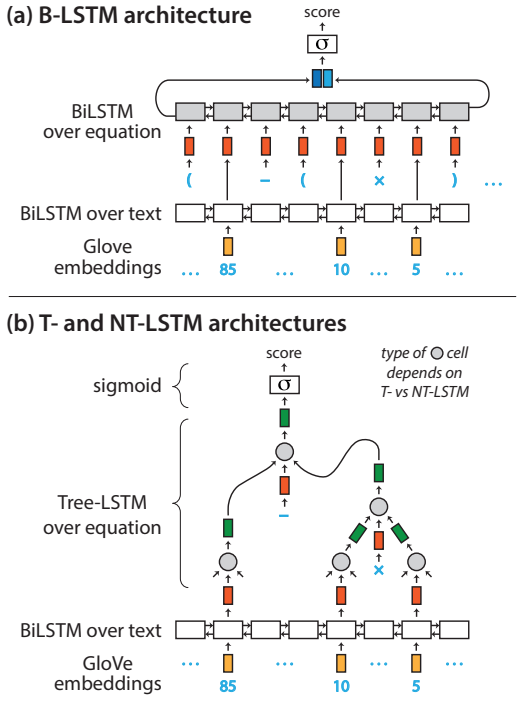


Figure A.3: Models for scoring equations, taking the text and the equation from Fig. A.1 to score (e.g., $85 - (10 \times 5)$) as input: (i) a word embedding layer at the bottom, (ii) a BiLSTM layer over the text, and (iii) a top layer that encodes the equation. For the latter we consider either **(a)** a sequential BiLSTM (B-LSTM architecture), or **(b)** a structured Tree-LSTM (T-LSTM and NT-LSTM architectures).

The input to our model is a **sequence of tokens** of length N, $W = \{w_1, ..., w_N\}$ of the arithmetic word problem, which we pass through an **embedding layer** to obtain embedded representations $X = \{x_1, ..., x_N\}$ where $x_t \in \mathbb{R}^{d_1}$. We adopt a BiLSTM to obtain **contextual representations** of the tokens. The following is the formal representation of the first LSTM [14] layer used to produce the representation referred to as "*BiLSTM over text*" in Fig. A.3:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{A.1}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{A.2}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{A.3}$$

$$u_t = \tanh\left(W_u x_t + U_u h_{t-1} + b_u\right) \tag{A.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{A.5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{A.6}$$

where $t \in \{1, ..., N\}$ represents a particular recursive execution time step and $h_t \in$

$\mathbb{R}^{d_2}$ is the LSTM hidden state. The advantage of using the LSTM-based structure instead of a simpler recursive formulation, such as $h_t = \tanh(Wx_t + Uh_{t-1} + b)$, is that an LSTM model avoids the problems of exploding or vanishing gradients during the training process discussed in [14, 28]. This is achieved by using additional weight matrices and *gates* σ in

(A.7)

A.1–A.3 in order to regulate the amount of information from previous execution steps h_{t-1} and current input x_t that affect the current state h_t.[3] More concretely, W_i, W_f, W_o, $W_c \in \mathbb{R}^{d_2 \times d_1}$ and U_i, U_f, U_o, $U_c \in \mathbb{R}^{d_2 \times d_2}$ are the weight matrices related to different LSTM gates, and b_i, b_f, b_o, $b_c \in \mathbb{R}^{d_2}$ are the respective biases. In our experiments we initialize x_t with GloVe word embeddings [29] and keep them static during training. These *GloVe embeddings* are depicted at the bottom of graphs **(a)** and **(b)** in Fig. A.3. In order to obtain the BiLSTM representation (*"BiLSTM over text"* in Fig. A.3), we run two LSTMs in different directions and concatenate the respective hidden states. This results in N hidden state representations $H = \{h_1^{(b)}, ..., h_N^{(b)}\}$ where $h_i^{(b)} \in \mathbb{R}^{d_3}$ and $d_3 = 2 \cdot d_2$. Using the input in H, we propose two different models to encode the candidate equations referred to as **(a)** and **(b)** in Fig. A.3, and explained below:

(a) Sequential B-LSTM: We perform an in-order traversal of the expression tree to obtain a sequential representation of the equation (e.g., $(85 - (10 \times 5))$) that is encoded using a second BiLSTM (see *"BiLSTM over equation"* in Fig. A.3**(a)**). We use as input the hidden state representations H calculated above for the numbers and (trainable) embeddings $O = \{o_-, o_+, o_\div, o_\times, o_(, o_)\}$ for the operators $(-, +, \div, \times)$ and opening/closing parentheses. More formally, the input to BiLSTM is represented by $X^E = \{x_1^e, ..., x_K^e\}$ where $x_t^e \in \{H \cup O\}$, $x_t^e \in \mathbb{R}^{d_3}$ and K is the number of tokens in the equation, including parentheses and operations. E.g., the equation $(85 - (10 \times 5))$ contains 9 tokens. In terms of the formal notation of LSTM in

(A.8)

A.1–A.6, each x_t^e corresponds to input vector x_t. In order to obtain a score for ranking the equation, we concatenate the last (left and right) hidden states of the BiLSTM producing a vector of dimensionality d_4, and then apply a linear transformation followed by a *sigmoid* function.

(b) Tree-LSTM: We base our implementation on the Tree-LSTM architecture proposed by [15]. This architecture is based on the LSTM formulation described in

(A.9)

A.1–A.6, but instead of being linearly linked, the input to a particular LSTM cell can come from different child step LSTM executions. More formally, we can describe the T-LSTM structure as follows:

$$\tilde{h}_t = \sum_{k \in \{L,R\}} h_{t-1}^k \tag{A.10}$$

$$i_t = \sigma\left(W_i x_t + U_i \tilde{h}_t + b_i\right) \tag{A.11}$$

[3] For a more detailed description of the LSTM architecture please refer to [14].

$$o_t = \sigma\left(W_o x_t + U_o \tilde{h}_t + b_o\right) \tag{A.12}$$

$$f_t^k = \sigma\left(W_f x_t + U_f h_{t-1}^k + b_f\right) \tag{A.13}$$

$$u_t = \tanh\left(W_u x_t + U_u \tilde{h}_t + b_u\right) \tag{A.14}$$

$$c_t = i_t \odot u_t + \sum_{k \in \{L,R\}} f_t^k \odot c_{t-1}^k \tag{A.15}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{A.16}$$

where $\{L, R\}$ is the set that consists of left (L) and right (R) child nodes for the current execution node at step t. More specifically, a particular execution step t corresponds to the respective arithmetic operation in the expression tree (see Fig. A.1(c)). This step takes as input the cell (c) and hidden (h) states of previous execution step ($t-1$) for each of the child nodes ($\{L, R\}$) that correspond to left and right operands in the expression tree. This execution process is recursive: each of the execution steps produces as output a hidden state h_t (Eq. (A.16)) which is used by the parent execution step recursively in Eq. (A.10) either as left (h_{t-1}^L) or right (h_{t-1}^R) child. Additionally, a *cell state* c_t is passed across the execution steps, and contains a summarized historic information of the tree traversal[4] operations performed so far. Similarly as with LSTM, a *forget* gate f_t^k, *input* (i_t) and *update* (u_t) gates are used to determine which historic information is kept (forget gate) and which new information is added (input/update gates) to the cell state. W_i, W_o, W_f, $W_u \in \mathbb{R}^{d_4 \times d_3}$ together with U_i, U_o, U_f, $U_u \in \mathbb{R}^{d_4 \times d_4}$ are the weight matrices that transform the inputs $x_t \in \mathbb{R}^{d_3}$, the current hidden state $\tilde{h}_t \in \mathbb{R}^{d_4}$ and the children's hidden states $h_{t-1}^k \in \mathbb{R}^{d_4}$, by means of the Tree-LSTM gate representations. As depicted in Fig. A.3**(b)**, the inputs x_t to the leaf nodes are the hidden state representations in H (coming from *"BiLSTM over text"* in Fig. A.3**(b)**) on the positions where the numbers occur in the problem statement. The input x_t to the inner nodes, on the other hand, are one of the randomly initialized operation embeddings $O = \{o_-, o_+, o_\div, o_\times\}$ depending on the operation represented by the node. This contrasts with the original setup proposed in [15] where the input x_t always comes from the word representation in the sentence. By using a separate operation embeddings set O as input, we expect our model to be able to capture a semantic representation for each of the different operations $o \in O$. The Tree-LSTM model finally outputs the hidden state for the root of the expression tree (i.e., the last executed operation), which is then passed through a sigmoid to deliver the score for a particular candidate arithmetic expression.

While T-LSTM allows to encode the equation information in a tree structure, it is symmetric in its child nodes. This is because the hidden states of the children are first summed up in Eq. (A.10) before applying the linear transformation and the gate activation functions. This could be problematic for non-commutative operations ($-$ and $\div$) where the result depends on the order of the operands. The reason for this is that Eq. (A.10) is commutative with respect to child nodes. Thus, given two child nodes $k \in \{L, R\}$ we have that $\tilde{h}_t = h_{t-1}^L + h_{t-1}^R = h_{t-1}^R + h_{t-1}^L$. As a

[4]Post-order traversal is used, since it reflects the order of operator execution in an arithmetic equation to obtain the final result.

consequence, the affine transformations U_i, U_o, and U_u in

(A.17)

A.11, A.12 and A.14 cannot capture the order of the states of the input nodes. Furthermore, since there is only one weight matrix U_f for both h^L_{t-1} and h^R_{t-1} in Eq. (A.13), it can not apply a different affine transformation for left and right child nodes. This makes the T-LSTM model indifferent to the order of the arguments of the operations in a particular expression tree. Therefore, we introduce a second model, called NT-LSTM, that uses distinct weight matrices to transform each of the children's hidden states. More formally, the gate definition in NT-LSTM is as follows:

$$i_t = \sigma\left(W_i x_t + \sum_{k\in\{L,R\}} U^k_i h^k_{t-1} + b_i\right) \tag{A.18}$$

$$o_t = \sigma\left(W_o x_t + \sum_{k\in\{L,R\}} U^k_o h^k_{t-1} + b_o\right) \tag{A.19}$$

$$f^k_t = \sigma\left(W_f x_t + \sum_{l\in\{L,R\}} U^{kl}_f h^l_{t-1} + b_f\right) \tag{A.20}$$

$$u_t = \tanh\left(W_u x_t + \sum_{k\in\{L,R\}} U^k_u h^k_{t-1} + b_u\right) \tag{A.21}$$

$$c_t = i_t \odot u_t + \sum_{k\in\{L,R\}} f^k_t \odot c^k_{t-1} \tag{A.22}$$

$$h_t = o_t \odot \tanh(c_t) \tag{A.23}$$

where, similarly as for T-LSTM, $\{L, R\}$ is the set of child nodes. By introducing different weights U for each of the child node states h^k_{t-1}, we make sure that the model can differentiate between the order of the operands. This is because now each of the affine transformations $U^{(l)}_i$, $U^{(l)}_o$ and $U^{(l)}_t$ is different for each input child hidden state h^l_{t-1} in

(A.24)

A.18, A.19 and A.21. Similarly, each of the children's ($k \in \{L, R\}$) forget gates f^k_t contains now two affine transformations U^{kl}_f ($l \in \{L, R\}$), one for each child. This way, the model can prioritize (components of f^k_t close to 1) or inhibit (components of f^k_t close to 0) separately the input of a particular child k based on the state of another child l ($k \neq l$). This can be useful when the state of one of the operands (e.g., influenced by the words that surround a particular number in text) has a strong indication of some operation, while the state of the other has very little evidence. As we will show in Section A.5, the use of NT-LSTM makes a big difference compared to the performance of T-LSTM for equations involving non-commutative operations.

A.4 Experimental setup

We evaluate the proposed models (code publicly available[5]) on the SingleEQ dataset introduced by [2]. SingleEQ consists of 1,117 sentences and 15,292 words, and includes 508 arithmetic problems of varying complexity (i.e., equations with single or multiple operators). Each of the word problems is mapped to a single correct equation with one unknown. These equations include one or more of the following operators: multiplication ($\times$), division ($\div$), subtraction ($-$), and addition ($+$). The data was gathered from the following grade-school websites: http://math-aids.com, http://k5learning.com, and http://ixl.com as well as from a subset of problems from [1]. To obtain results comparable to previous work, we perform 5-fold cross-validation using the original splits defined in [2]. Similar to the work of [2] and [4], we report performance using the overall accuracy metric. The training/testing process is run for 5 different splits, in each one a separate fold is left as test set. This way, our results are reported on the whole SingleEQ dataset by concatenating the predictions of *test* folds across the splits. In total, we train 25 models with different seeds (5 for each split) and report average and standard deviation in Tables A.4–A.5 and A.7 in Section A.5. Furthermore, we tune the neural net hyperparameters independently for each of the splits on the validation set that consists of 20% randomly selected arithmetic problems in each of the train folds. Due to limited resources that prevented us to perform a complete grid search, we conduct the hyperparameter tuning in steps. More specifically, in each step we perform a grid search on two hyperparameters that we identified as most correlated with each other. Table A.2 summarizes our hyperparameter search space for each of the sequential tuning steps. Besides the usual hyperparameters (i.e., learning rate, batch size and dropout) tuning, we also adjust the dimensionalities d_3 (Dim LSTM) of the first BiLSTM layer (indicated as "*BiLSTM over text*" in Fig. A.3), and d_4 (Dim Encoder) of either the sequential BiLSTM ("*BiLSTM over equation*" in Fig. A.3) or the tree-based NT-LSTM models' encoder layers ("*Tree-LSTM*" in Fig. A.3). The best hyperparameters are chosen after training for 75 epochs for each of the cross-validation splits independently.

Step	Hyperparameters				
	Learning Rate	Batch Size	Dim LSTM	Dim Encoder	Dropout
1	{3e − 4, 1e − 4}	{64, 128}	-	-	-
2	-	-	{256, 512}	-	{0.3, 0.4}
3	-	-	-	{256, 512}	{0.3, 0.4}

Table A.2: The range of the hyperparameter search space for each of the hyperparameter tuning steps for each of the cross-validation splits of SingleEQ dataset.

Furthermore, we partition the dataset into several subsets to investigate the effect of varying problem complexity on the models' performances. These different subsets are characterized in Table A.3. We form three main categories: (i) **Full:** the whole dataset is included in this setting, (ii) **Complexity:** two subsets (i.e., Single,

[5] https://github.com/klimzaporojets/arithmetic-word-problems

Subset	Equation types	# Problems
Full	All operators	508
Single	Single operator	390
Multi	Multiple operators	118
$\text{Single}_{\text{sym}}$	Single symmetric operators	208
$\text{Multi}_{\text{sym}}$	Multiple symmetric operators	68
$\text{Single}_{\text{asym}}$	Single asymmetric operators	182
$\text{Multi}_{\text{asym}}$	Multiple asymmetric operators	50

Table A.3: The defined subsets of the SingleEQ dataset with varying degrees of complexity.

Multi) are formed based on the number of operators in the solution's equation, and (iii) **Symmetry:** four main subsets, namely $\text{Single}_{\text{sym}}$, $\text{Single}_{\text{asym}}$, $\text{Multi}_{\text{sym}}$, and $\text{Multi}_{\text{asym}}$ are formed to indicate whether the solution's equation contains single/-multiple symmetric ($\times$ and $+$) or asymmetric ($\div$ and $-$) operations.

We hypothesize that our Tree-LSTM models will exhibit stronger performance on subsets involving multiple and/or non-commutative operations (Multi, $\text{Multi}_{\text{sym}}$, $\text{Multi}_{\text{asym}}$), since they should be able to better capture the semantic relationships between operator nodes encoded in a tree structure. We also expect a significant difference between T-LSTM and NT-LSTM architectures on subsets involving non-commutative operations ($\text{Single}_{\text{asym}}$ and $\text{Multi}_{\text{asym}}$). By using different weight matrices to transform each of the children's states (see Eqs. A.18–A.21 of the NT-LSTM in Section A.3.2 for more details), the NT-LSTM model should be able to capture the order of the operands and link the resulting structural information of a particular non-commutative mathematical expression to the semantic representation of the problem statement.

We obtain the top-100 equation-trees using the ILP solver of [2], which we rank using scores provided by our proposed model (see Section A.3.2). Training of our model is performed using the Adam optimizer [30]. As a bottom token representation layer, we use pre-trained 100-dimensional ($d_1 = 100$) GloVe embeddings [29][6] which we keep static during the training process.

A.5 Results

In this section, we evaluate the performance of our proposed models on the SingleEQ dataset. Besides the performance on the full dataset, we are particularly interested in evaluating how each architecture behaves when evaluated on arithmetic problems of varying complexity. We assume that the problems become more complex (i) as the number of needed mathematical operators grows, and (ii) when the used operators are non-commutative (asymmetric). We hypothesize that our structured Tree-LSTM-based approach is better suited to solve the aforementioned complex problems. In order to demonstrate this, we perform an extensive evaluation (Tables A.4–A.5 and A.7) of our models on subsets of different degree of

[6] https://nlp.stanford.edu/projects/glove/

Model	Features	Trees	Accuracy (%)
[6]	✓	✗	48.00
[4]	✓	✓	52.96
[21]	✓	✓	66.38
[22]	✓	✓	72.25
ALGES	✓	✓	72.39
ILP Coverage	-	-	91.34
ILP Naive	-	-	52.56
B-LSTM	✗	✗	74.88±0.64
T-LSTM	✗	✓	74.88±1.06
NT-LSTM	✗	✓	**75.47±0.62**

Table A.4: Accuracy attained by the proposed and state-of-the-art methods on the *Full* SingleEQ dataset. The ✓ and ✗ symbols indicate whether or not a model adopts hand-crafted features ('Features') or tree-structured encoding of the equations ('Trees'). The best result is typeset in **bold**.

complexity as defined in Table A.3. Furthermore, in all of the result tables we include the potential maximum accuracy that can be achieved when using the candidates from the ILP *candidate generator* (ILP Coverage). This allows us to estimate how much improvement can still be achieved by *candidate ranker*. Conversely, in order to evaluate the impact of *candidate ranker* models, we also report the accuracy achieved when picking the top-weighted candidate by ILP solver (ILP Naive).

Comparison on the Full dataset: Table A.4 shows the results of the evaluated systems on the Full SingleEQ dataset. The proposed models are the (i) B-LSTM, (ii) T-LSTM, and (iii) NT-LSTM as presented in Section A.3.2. Clearly, all newly proposed architectures outperform previous methods. Concretely, our methods are able to outperform strong baselines on the task, reporting an accuracy improvement of more than 3% without relying on hand-crafted features [2, 6, 21, 22]. As detailed later on in this section (see analysis of Table A.5 and Table A.7), most of this improvement with respect to the current state-of-the-art [2] comes from an increased performance on the more complex arithmetic word problems that involve non-commutative and multiple operations. This supports our original hypothesis that tree-based architectures are superior in representing mathematical operations between operands, specially when the mathematical expressions involve multiple operations. The hand-crafted features, used in previous works, are usually related to terms indicating specific operations and thus if they are not detected in the data, the system cannot generalize well on out-of-domain mathematical descriptions. This also applies to recent neural-based methods (see, e.g., [4]) where explicitly defined features are encoded in the neural structure. Furthermore, in order to ensure the validity of the differences between our proposed approaches, we carry out a bootstrap significance analysis [31] by sampling with replacement the results of B-LSTM, T-LSTM, and NT-LSTM models 10,000 times. We compare the performance with respect to the NT-LSTM model in Table A.4. We observe that, while our NT-

Model	Complexity		Symmetric		Asymmetric	
	Single	Multi	$\text{Single}_{\text{sym}}$	$\text{Multi}_{\text{sym}}$	$\text{Single}_{\text{asym}}$	$\text{Multi}_{\text{asym}}$
ILP Coverage	93.33	84.75	94.71	83.82	91.76	86.00
ILP Naive	56.41	39.83	53.85	69.12	59.34	0.00
ALGES	77.69‡	54.70‡	**89.90**	72.06	63.74‡	30.64‡
B-LSTM	79.59±0.72	59.32±2.34	80.87±0.64‡	69.12±2.08‡	78.13±1.36	**46.00±4.38**
T-LSTM	79.59±1.24	59.32±1.61	81.35±0.98‡	**72.35±1.44**	77.58±2.72	41.60±2.33⋆
NT-LSTM	**80.21±0.95**	**59.83±1.75**	81.35±1.44‡	71.17±2.20	**78.90±2.13**	44.40±4.96

Table A.5: Comparison of the proposed methods with the state-of-the-art on the SingleEQ dataset in terms of accuracy. **Bold** font indicates the best results for each subset of SingleEQ (see Table A.3). The markers ⋆, †, ‡ respectively indicate the achieved bootstrap significance levels α <0.1, <0.05 and <0.01 with respect to the best performing model in each of the subsets.

LSTM model seems to outperform T-LSTM and B-LSTM models, this difference in performance is not significant.

Comparison for different problem complexity: Table A.5 compares our models with ALGES [2] (i.e., the best performing state-of-the-art model of Table A.4), for subsets of different complexity levels (defined in Table A.3). We use bootstrap significance testing to estimate the degree of certainty between the lower performing models and the best performing one in each of the subsets. We indicate significant differences with p-values below the 1%, 5%, and 10% level (respectively denoted with ‡, †, and ⋆) in order to identify models performing significantly different from the best performing model in each of the subsets.

We observe that our newly proposed models do not significantly differ among each other for solving problems involving single (Single, $\text{Single}_{\text{sym}}$, and $\text{Single}_{\text{asym}}$ subsets) operations. Conversely, on the problem subset requiring multiple commutative operations in their solution ($\text{Multi}_{\text{sym}}$), our tree-based T-LSTM significantly outperforms the sequential B-LSTM model, suggesting a potential benefit in using tree-based models to solve the problems involving multiple operations. For the subset involving multiple non-commutative operations ($\text{Multi}_{\text{asym}}$) the B-LSTM and NT-LSTM models outperform the T-LSTM model, indicating a potential limitation of the latter in dealing with non-commutative operations, due to its symmetrical structure in its child nodes (a single weight matrix is used on the sum of children's states $\tilde{h}_t$ as described in Section A.3.2). We were surprised by an overall good performance of our sequential B-LSTM model, specially on $\text{Multi}_{\text{asym}}$ subset, where it performs on par with the potentially more expressive NT-LSTM model. This fact also motivated us to explore the robustness of our models against additional asymmetric noise (see further analysis in the next paragraphs corresponding to the results in Table A.7).

The results in Table A.5 further show that the feature-based ALGES model has competitive performance on problems requiring single and/or non-commutative operators in the solution equations. In fact, it significantly outperforms all our models on the $\text{Single}_{\text{sym}}$ dataset and is only marginally outperformed by our tree-based T-LSTM model on $\text{Multi}_{\text{sym}}$. This suggests that the feature-based ALGES is able to explicitly capture symmetric operations by focusing on carefully engi-

Candidates	Metric	Subsets						
		Full	Single	Multi	$\text{Single}_{\text{sym}}$	$\text{Multi}_{\text{sym}}$	$\text{Single}_{\text{asym}}$	$\text{Multi}_{\text{asym}}$
ILP	Correct	2.53	1.44	6.13	1.89	7.72	0.92	3.96
	Incorrect	12	2.9	42.08	2.48	28.43	3.38	60.64
ILP + Asym	Correct	2.41	1.44	5.62	1.89	7.66	0.92	2.84
	Incorrect	15.08	4.06	51.5	3.57	35.43	4.62	73.36
	Δ Correct	−4.74%	0.00%	−8.32%	0.00%	−0.78%	0.00%	−28.28%
	Δ Incorrect	25.67%	40.00%	22.39%	43.95%	24.62%	36.69%	20.98%

Table A.6: This table illustrates the difference in average number of *Correct* and *Incorrect* candidate equations per problem between the original *ILP* candidate generation process and the one obtained by adding noisy equations with asymmetric operators (*ILP + Asym*).

neered features. However, we observe a large drop in performance of ALGES on problems that require non-commutative (asymmetric) operations to be solved. This is showcased by a difference of more than 15% accuracy points on $\text{Single}_{\text{asym}}$ and $\text{Multi}_{\text{asym}}$ subsets in Table A.5. This validates our initial intuition that feature-based models fall short to capture the reasoning necessary to address problems that require more complex (non-commutative and multiple) operators.

Robustness against asymmetric noise: The results analyzed so far are based on scoring the candidates generated by the ILP component introduced in [2]. However, this component already significantly reduces the number of incorrect candidates, particularly those involving asymmetric operators (e.g., by removing candidate equations that produce negative or fractional results as described in Section A.3.1). In order to evaluate the robustness of the proposed models, we train and evaluate them on a noisy asymmetric candidate set where we add all possible permutations to the equations involving non-commutative operators. For example, if a particular candidate equation is $x = 8/2$, we would also add $x = 2/8$ to the candidate set. Table A.6 shows the statistics of the noisy dataset (ILP + Asym) with respective deltas that indicate the percentage points (%) of increase/decrease in the average number of correct/incorrect candidate equations per problem with respect to the original ILP-generated candidate set. We observe a significant increase in the number of incorrect candidates for all subsets, as well as a drop in average number of correct equations for the subsets involving asymmetric operations (Multi and $\text{Multi}_{\text{asym}}$). This is because, similarly as in the original *ILP* setup, we only consider the first 100 generated candidates, which in *ILP + Asym* include more incorrect equations, leaving many correct ones out. This results in a lower correct/incorrect ratio that makes it more challenging for the evaluated models to find the right mathematical expression to solve a particular problem. Table A.7 compares our models with the best performing state-of-the-art model (i.e., ALGES) on candidates generated in the *ILP + Asym* setting. Compared to the results presented in Table A.5, we observe a sharp decrease in performance of the ALGES model on subsets involving multiple operations (Multi, $\text{Multi}_{\text{sym}}$ and $\text{Multi}_{\text{asym}}$). This demonstrates once more the weakness of this feature-based model in capturing the reasoning necessary to distinguish the order of the operands involved in equations containing multiple and non-commutative operators. Furthermore, we observe that the sequential B-LSTM model is now significantly outperformed by

Model	Full	Complexity		Symmetric		Asymmetric	
		Single	Multi	$\text{Single}_{\text{sym}}$	$\text{Multi}_{\text{sym}}$	$\text{Single}_{\text{asym}}$	$\text{Multi}_{\text{asym}}$
ILP Coverage	91.14	93.33	83.90	94.71	83.82	91.76	84.00
ILP Naive	52.56	56.41	39.83	53.85	69.12	59.34	0.00
ALGES	68.44‡	75.90†	43.59‡	**85.58**	61.76‡	64.83‡	18.36‡
B-LSTM	72.99±1.14	**78.21±0.97**	55.76±2.10‡	83.36±1.20†	71.76±3.40‡	72.30±2.37	34.00±2.19⋆
T-LSTM	57.95±1.34‡	61.69±1.49‡	45.59±1.25‡	80.58±2.44‡	72.65±2.20‡	40.11±0.92‡	8.80±0.98‡
NT-LSTM	**73.19±0.93**	76.97±1.02†	**60.67±1.15**	80.76±2.37‡	**76.47±0.93**	**72.63±1.61**	**39.20±2.40**

Table A.7: Comparison of the proposed methods with the state-of-the-art model (i.e., ALGES) on the SingleEQ dataset in terms of accuracy evaluated on candidate equations generated using *ILP + Asym* procedure (see Table A.6). **Bold** font indicates the best results for each subset of SingleEQ (see Table A.3). The markers ⋆, †, ‡ respectively indicate the achieved bootstrap significance levels α <0.1, <0.05 and <0.01 with respect to the best performing model in each of the subsets.

the tree-based NT-LSTM on subsets involving multiple operations to be solved (Multi, $\text{Multi}_{\text{sym}}$ and $\text{Multi}_{\text{asym}}$). This again supports our initial hypothesis that tree-structured approach is better suited to capture more complex reasoning which is necessary to solve arithmetic problems. In the *ILP + Asym* candidate generation setting this is even more important because of the additional noise introduced with the incorrect candidates that involve multiple and asymmetric operations. Conversely, for arithmetic problems involving single operations to be solved (Single, $\text{Single}_{\text{sym}}$, and $\text{Single}_{\text{asym}}$ subsets), the B-LSTM model shows a competitive performance, surpassing the tree-based NT-LSTM model on problems requiring single commutative operations ($\text{Single}_{\text{sym}}$). Additionally, we observe an important drop in performance of T-LSTM model which is mainly influenced by low accuracy scores on asymmetric subsets ($\text{Single}_{\text{asym}}$ and $\text{Multi}_{\text{asym}}$). This is in line with our initial intuition that by using a single weight matrices U_i, U_o, U_f, U_u to transform either the sum of the children's states $\tilde{h}_t$ (see

(A.25)

A.10–A.12 and A.14) or the individual children states h_k (Eq. (A.13)), the T-LSTM model is unable to distinguish the order of the operands involved in asymmetric equations. This difference is less evident in Table A.5 because most of the incorrect candidates involving non-commutative operations are already filtered out by the ILP component. However, in our *ILP + Asym* candidate generation setup, we make sure that for each candidate involving non-commutative operation, we also include noisy candidates with all the possible asymmetric permutations. This makes it necessary not only to detect the right operation, but also to distinguish the order of the operands, where the T-LSTM model fails. Finally, we observe that overall (on Full dataset) our tree-based NT-LSTM model exhibits less variance among the different bootstrap results, compared to the sequential B-LSTM model. This indicates that NT-LSTM model is less susceptible to different seed initialization during the training process, making it more robust than other proposed models (T-LSTM and B-LSTM).

Error Analysis: In order to understand our system's weaknesses, we manually analyzed the errors that it consistently makes across different training seed instances.

Type	Problem Text	NT-LSTM
Complex reasoning (57%)	Seth bought 20 cartons of ice cream and 2 cartons of yogurt. Each carton of ice cream cost $6 and each carton of yogurt cost $1. How much more did Seth spend on ice cream than on yogurt?	$20/2 - 1 \times 6$
Parsing and counting (22%)	Jane's dad brought home 24 marble potatoes. If Jane's mom made potato salad for lunch and served an equal amount of potatoes to Jane, herself and her husband, how many potatoes did each of them have?	n/a
World Knowledge (21%)	Bert runs 2 miles every day. How many miles will Bert run in 3 weeks?	3×2
	The sum of three consecutive odd numbers is 69. What is the smallest of the three numbers?	n/a

Table A.8: Examples of problems where our NT-LSTM model fails.

We grouped them into three main categories represented in Table A.8: *complex reasoning*, *parsing and counting*, and *world knowledge* errors. We observe that more than half (57%) of our system's errors are due to problems requiring *complex reasoning* while the numbers have been correctly extracted from the text. This reflects the results from Tables A.5 and A.7 that show lower performance of our models on problems requiring multiple and/or non-commutative operations. As future work to alleviate this type of problems we can complement the tree-structures using additional information such as the entities inside the sentence. For instance, in the first example illustrated in Table A.8, if the system would know that "ice cream" from the second sentence represents the same concept as in the first one, it would be easier to link numbers 6 and 20. A second consistent type of error is related to *parsing and counting*. It mainly happens when there are several entities involved in a problem statement and the system has to count them correctly. For instance, in the second example presented in Table A.8, our current system is unable to produce the correct candidate mathematical expression since it can only extract the number 24 from text. Further work in improving aspects related to parsing and entity identification in the problem statement should significantly reduce this kind of mistakes. Finally, the *world knowledge* related errors account for 21% of the total mistakes. Most of these errors are due to the fact that the system is unable to capture the units correctly (i.e., there are 7 days in a week, or one dime equals 0.1 dollars). However, as in the second example, some of the problems require a more advanced conceptual world understanding, such as the notion of odd numbers. Future work can be directed towards methods that are able to capture and represent this kind of world knowledge.

Limitations of the current state-of-the-art: We performed an empirical study on the predicted results to understand better where our proposed model outperforms the current state-of-the art model, ALGES [2]. Table A.9 illustrates some examples

Problem Text	ALGES	NT-LSTM
Nancy bought 615 crayons that came in packs of 15. How many packs of crayons did Nancy buy?	$615 - 15$	$615/15$
Carrie's mom gave her \$91 to go shopping. She bought a sweater for \$24, a T-shirt for \$6, and a pair of shoes for \$11. How much money does Carrie have left?	$91 + 24 + 6 + 11$	$91 - (24 + 6 + 11)$
Melanie had 19 dimes in her bank. Her dad gave her 39 dimes and her mother gave her 25 dimes. How many dimes does Melanie have now ?	$19 - 39 + 25$	$19 + 39 + 25$
On Saturday, Sara spent \$10.62 each on 2 tickets to a movie theater. Sara also rented a movie for \$1.59, and bought a movie for \$13.95. How much money in total did Sara spend on movies?	$10.62 + 2 \times 1.59 + 13.95$	$10.62 \times 2 + 13.95 + 1.59$

Table A.9: Examples of problems that NT-LSTM provides a correct solution, but current state-of-the-art ALGES [2] fails.

of the problems where our model gets consistently correct predictions on different training initialization weights (Section A.4). Most of the gains came from improving on problems requiring multiple and/or asymmetric operations, corroborating our previous findings.

Strengths of the current state-of-the-art and limitations of our approach: Tables A.5 and A.7 illustrate that in the case of single symmetric operations ($\text{Single}_{\text{sym}}$), the ALGES method outperforms the proposed architectures (i.e., B-LSTM, T-LSTM, and NT-LSTM). We hypothesize that the main reason for this is the use of carefully hand-engineered features, many of which depend on third-party tools (e.g., dependency parsing). Table A.10 illustrates four examples whose solution requires mathematical expressions with a single operator. In the first two cases our NT-LSTM model is outperformed by the current state-of-the-art ALGES which correctly predicts the commutative operators ($+$ in the first example and $\times$ in the second one). We have found that these correctly predicted commutative cases are highly correlated with the *entity match* feature (i.e., when the noun phrase connected to the number such as "pounds" in the first example is the same in two numbers). This feature has high positive correlation with addition and negative correlation with multiplication operations, which is illustrated in the first and second examples respectively. It also requires an additional dependency parsing which, in case of ALGES, is performed using Stanford Dependency Parser[7]. Other word-based features are also highly correlated with some operations. For example, the presence of the word "and" in the description of the problem is correlated with addition. However, while these features may be a strong indicators of some operators, their

[7]More concretely, the Stanford Dependency Parser in CoreNLP 3.4 is used.

Problem Text	ALGES	NT-LSTM
Diane is a beekeeper. Last year, she harvested 2,479 **pounds** of honey. This year, she bought some new hives **and** increased her honey harvest by 6,085 **pounds**. How many pounds of honey did Diane harvest this year?	$6,085 + 2,479$	$6,085 - 2,479$
Jack has a section filled with short story booklets. If each booklet has 9 **pages** and there are 49 **booklets** in the short story section, how many pages will Jack need to go through if he plans to read them all?	9×49	$9 + 49$
Benny received 67 **dollars** for his birthday. He went to a sporting goods store **and** bought a baseball glove, baseball, **and** bat. He had 33 **dollars** left over. How much did he spent on the baseball gear?	$67 + 33$	$67 - 33$
Jane's mom picked cherry tomatoes from their backyard. If she gathered 56 **cherry tomatoes and** is about to place them in small jars which can contain 8 **cherry tomatoes** at a time, how many jars will she need?	$56 + 8$	$56/8$

Table A.10: Examples of problems that require a single operation to be solved. The first two involve commutative operations ($+$ and $\times$ respectively) where our NT-LSTM model fails compared to the feature-based model (ALGES; [2]). The rest of the examples illustrate cases where ALGES fails and NT-LSTM returns the correct answer. The words that represent features used in ALGES that are highly correlated with the predicted operation (*entity match* and the word *"and"*) are highlighted.

application is limited to problems where the underlying patterns appear. This is illustrated in the last two examples that contain two features highly correlated with the addition (i.e., *entity match* and "and" word), but that require a different (non-commutative) operation in their solutions. In both cases, biased by the most likely feature-based operation, the answer given by ALGES is incorrect. This contrasts with our feature-independent NT-LSTM model which manages to predict the correct equation. This is reflected in Tables A.5 and A.7, where the features-based approach falls short in capturing the more intricate nature of solutions involving non-commutative operations ($\text{Single}_{\text{asym}}$ and $\text{Multi}_{\text{asym}}$). In these cases, our tree-based NT-LSTM model exhibits superior performance.

A.6 Conclusion

In this work we addressed the reasoning component involved in solving arithmetic word problems. We proposed a recursive tree architecture to encode the underlying equations for solving arithmetic word problems. More concretely, we proposed to use two different Tree-LSTM architectures for the task of scoring candidate equations. We performed an extensive experimental study on the SingleEQ dataset and demonstrated consistent effectiveness (i.e., more than 3% increase in accuracy on the Full dataset and more than 15% for a subset of complex reasoning tasks) of our models compared to current state-of-the-art.

We observed that, while very strong on simple instances involving single operations, the current feature-based state-of-the-art model exhibits a significant gap in performance for mathematical problems whose solution comprises non-commutative and/or multiple operations. This reveals the weakness of this method to capture the intricate nature of reasoning necessary to solve more complex arithmetic problems. Furthermore, our experiments show that, while a traditional sequential approach based on recurrent encoding implemented using BiLSTMs over the equation proves to be a robust baseline, it is outperformed by our recursive Tree-LSTM architecture to encode the candidate solution equation on more complicated problems that require multiple operations to be solved. This difference in performance becomes more significant as we introduce additional noise in our set of candidates by adding incorrect equations that contain non-commutative operations.

Acknowledgments

Part of the research leading to these results has received funding from (i) the European Union's Horizon 2020 research and innovation programme for the CPN project under grant agreement no. 761488, and (ii) the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

[1] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay. *Learning to automatically solve algebra word problems*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 271–281, 2014.

[2] R. Koncel-Kedziorski, H. Hajishirzi, A. Sabharwal, O. Etzioni, and S. D. Ang. *Parsing algebraic word problems into equations*. Transactions of the Association for Computational Linguistics, 3:585–597, 2015.

[3] A. Mitra and C. Baral. *Learning to use formulas to solve simple arithmetic problems*. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 2144–2153, 2016.

[4] L. Wang, D. Zhang, L. Gao, J. Song, L. Guo, and H. T. Shen. *Mathdqn: Solving arithmetic word problems via deep reinforcement learning*. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[5] D. Zhang, L. Wang, L. Zhang, B. T. Dai, and H. T. Shen. *The gap of semantic parsing: A survey on automatic math word problem solvers*. IEEE, 2019.

[6] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman. *Learning to solve arithmetic word problems with verb categorization*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 523–533, 2014.

[7] Y. Wang, X. Liu, and S. Shi. *Deep neural solver for math word problems*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 845–854, 2017.

[8] J. Li, L. Wang, J. Zhang, Y. Wang, B. T. Dai, and D. Zhang. *Modeling Intra-Relation in Math Word Problems with Different Functional Multi-Head Attentions*. In Proceedings of the 57th Conference of the Association for Computational Linguistics, pages 6162–6167, 2019.

[9] T.-R. Chiang and Y.-N. Chen. *Semantically-Aligned Equation Generation for Solving and Reasoning Math Word Problems*. 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 2656–2668, 2019.

[10] S. Shi, Y. Wang, C.-Y. Lin, X. Liu, and Y. Rui. *Automatically solving number word problems by semantic parsing and reasoning*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1132–1142, 2015.

[11] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom. *Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 158–167, 2017.

[12] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. *MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms*. 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 2357–2367, 2019.

[13] L. Wang, Y. Wang, D. Cai, D. Zhang, and X. Liu. *Translating a Math Word Problem to a Expression Tree*. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1064–1069, 2018.

[14] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural computation, 9(8):1735–1780, 1997.

[15] K. S. Tai, R. Socher, and C. D. Manning. *Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks*. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 1, pages 1556–1566, 2015.

[16] D. G. Bobrow. *Natural language input for a computer problem solving system*. 1964.

[17] E. Charniak. *CARPS: a program which solves calculus word problems*. 1968.

[18] E. Charniak. *Computer solution of calculus word problems*. In Proceedings of the 1st international joint conference on Artificial intelligence, pages 303–316. Morgan Kaufmann Publishers Inc., 1969.

[19] C. R. Fletcher. *Understanding and solving arithmetic word problems: A computer simulation*. Behavior Research Methods, Instruments, & Computers, 17(5):565–571, 1985.

[20] Y. Bakman. *Robust understanding of word problems with extraneous information*. 2007.

[21] S. Roy and D. Roth. *Solving General Arithmetic Word Problems*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1743–1752, 2015.

[22] S. Roy and D. Roth. *Unit dependency graph and its application to arithmetic word problem solving*. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[23] S. Roy and D. Roth. *Mapping to declarative knowledge for word problem solving*. Transactions of the Association of Computational Linguistics, 6:159–172, 2018.

[24] D. Huang, J. Liu, C.-Y. Lin, and J. Yin. *Neural Math Word Problem Solver with Reinforcement Learning*. In Proceedings of the 27th International Conference on Computational Linguistics, pages 213–223, 2018.

[25] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. *Parsing natural scenes and natural language with recursive neural networks*. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 129–136, 2011.

[26] R. Socher, J. Bauer, C. D. Manning, et al. *Parsing with compositional vector grammars*. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 455–465, 2013.

[27] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. *Enhanced LSTM for Natural Language Inference*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1657–1668, 2017.

[28] Y. Bengio, P. Simard, and P. Frasconi. *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, 5(2):157–166, 1994.

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

[30] D. Kingma and J. Ba. *Adam: A method for stochastic optimization*. In Proceedings of the International Conference on Learning Representations, San Diego, USA, 2015.

[31] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

B

Predicting Psychological Health from Childhood Essays The UGent-IDLab CLPsych 2018 Shared Task System

In this chapter we describe our contribution to CLPsych 2018 shared task where we achieve competitive results using an ensemble consisting of multiple models to predict depression and anxiety in textual surveys.

⋆⋆⋆

K. Zaporojets, L. Sterckx*, J. Deleu, T. Demeester and C. Develder

5th Annual Workshop on Computer Linguistics and Clinical Psychology (CLPsych 2018) at NAACL-HLT 2018

Abstract This paper describes the IDLab system submitted to Task A of the CLPsych 2018 shared task. The goal of this task is predicting psychological health of children based on language used in hand-written essays and socio-demographic control variables. Our entry uses word- and character-based features as well as lexicon-based features and features derived from the essays such as the quality of the language. We apply linear models, gradient boosting as well as neural-network based

regressors (feed-forward, CNNs and RNNs) to predict scores. We then make ensembles of our best performing models using a weighted average.

B.1 Introduction

The goal of the CLPsych 2018 shared task is to predict the psychological health of children based on essays and socio-demographic control variables. The provided data stems from the National Child Development Study (NCDS) which followed a number of people born in a single week of March 1958 in the UK [1]. The psychological health of this group of individuals was monitored in intervals of several years. At the age of 11, participants were asked to write an essay describing where they saw themselves at age 25. Simultaneously, their psychological health was evaluated by their teachers based on metrics defined by the Bristo Social Adjustment Guides (BSAG) [2].

Given the written essays and social control variables (gender and social class), CLPsych participants are to predict three types of BSAG scores: (i) `total` BSAG score, (ii) the `depression` BSAG score, and (iii) the `anxiety` BSAG score. In order to predict these scores, participants are allowed to use the social control variables next to the features extracted from the essays themselves.

Our system uses several types of features: bag-of-word and bag-of-character features, features derived from lexicons and term lists, and features based on text statistics (see Section B.3.2 for more details). Using these features, we apply several types of regressors: linear models, gradient boosting and neural-network based models. For each of the regressors, we explore different combinations of features to predict each of the BSAG scores. Subsequently, these models are combined using weighted average ensembling. Two sets of predictions were made: the first one is based on the single best models, a second uses an ensemble of models for each of the three scores (`depression`, `anxiety` and `total` BSAG scores).

Our ensemble of models gives a competitive result, positioning our system on the second place with only 0.01 points under the winner of this shared task. We think that this good performance is mostly due to the different nature of our individual models which complement each other when ensembled.

The remainder of this paper is organized as follows: Section B.2 describes the shared task in more detail. Section B.3 presents the features used by the regressors. Section B.4 describes regressors and the general methodology of our approach. Section B.5 describes results we obtained during development on our internal validation set and on the real test set. Finally, we summarize our findings and present future directions in Section B.6.

B.2 Task and data

Input for task A consists of essays written by 11-year-old children describing where they see themselves at age 25, as well as several social control variables:

1. **Gender**: gender of the participant child.
2. **Social Class**: the job hierarchy of the father of the participant child. The domain comprises 6 values representing different job categories: starting with

professional and managerial occupations and ending with unskilled occupations.

3. **Essay**: content of the essay written by the participant child. Originally, the essays were hand-written and later transcribed in digital format. The average length of the essays is 225 characters.

The goal of shared task A is to predict the current psychological health of the children. Psychological health is measured using scores assigned by teachers of the children following metrics defined in the BSAG. These guides score the total psychological health using 12 different syndromes (depression, anxiety, hostility, etc.). CLPsych shared task A requires participants to predict three scores:

1. **Total**: the sum of all the BSAG scores of all the different syndromes.
2. **Depression**: the BSAG score related to the depression syndrome.
3. **Anxiety**: the BSAG score related to the anxiety syndrome.

Participants are given a training set consisting of essays from 9,217 children with corresponding input variables and BSAG scores.

B.3 Features

In this section, we present features used by our models, and experiment with a number of different categories of feature extraction.

B.3.1 Lexical features

We use bag-of-n-gram features both on word- and character-level. The latter provides robustness to the spelling variation found in children's writing. For word-level we experiment with n-grams for n ranging from 1 to 4. At character-level, we experiment with 3- up to 6-grams. These one-hot encodings are weighted using TF-IDF.

B.3.2 Feature engineering

Next to the sparse bag-of-n-grams representations of the essays, we apply several manually designed features.

Social control features These features are given as input in the data and consist of the *gender* and *social class* of the participants. In order to be used in regressors, we encode these features as one-hot vectors.

Lexicon-based features We experiment with features based on two lexicons: the Linguistic Inquiry Word Count (LIWC) described in [3] and the DepecheMood [4]. The LIWC is a psycholinguistic lexicon that allows to measure the emotional health of individuals by providing a set of term categories related to different mental states. In our experiments we use all 73 (partly overlapping) psychological word categories found in the LIWC dictionary.

Similarly, DepecheMood is a lexicon consisting of 37k different words (verbs, nouns, adjectives and adverbs). Each of the words has weights associated to the following 8 mental states: afraid, amused, angry, annoyed, don't care, happy, inspired and sad. In our experiments, we calculate the average of TF-IDF weights for

these categories. These TF-IDF weights are already given inside DepecheMood lexicon and are originally calculated on articles from `rappler.com` based on Rappler's *Mood Meter* crowdsourcing.

Textual statistics features We extract a number of features describing several characteristics of the essays:

- Total number of words
- Average sentence length
- Average word length
- Ratio of spelling mistakes
- Ratio of different words
- Number of words not recognized (illegible) when transcribing the essays from hand-written to digital form.

Sentiment features We reason that the participants' psychological health can partially be detected by evaluating the essay in a positive-negative sentiment spectrum. We use the pretrained sentiment classifier from [5].[1] We hypothesize that individuals with good psychological health will tend to use more positive expressions than individuals with high scores in any of BSAG syndromes.

Language model features Coming from the intuition that mental state may be related to the development of language skills, we include two language model features. Our primary language model feature is the average perplexity of the essays, as it is an often used metric to score the general language quality and coherence of the texts. As a secondary feature, we include the fraction of out-of-vocabulary tokens over the entire essay, with respect to the Penn Treebank data. We use the word-level `AWD_LSTM` language model trained on the Penn Treebank, presented by [6].

B.4 Models description

We train a variety of different regression models predicting the three aforementioned BSAG scores. We include simple linear models as well as gradient boosted trees and neural network-based models. Our best performing models are subsequently combined using ensembling. As a general rule, we try to select different model function types in order to achieve lower correlation between predictions from the different types of models.

B.4.1 Linear models

We experiment with two types of linear regressors: support vector machines (SVMs) and ridge regression. Linear models are trained on two sets of features.

1. *Lexical features* based purely on the text of the essays (see Section B.3.1). Here we use TF-IDF weighted bag-of-word features as well as character features.
2. *Designed features* through feature engineering (see Section B.3.2).

[1] The python library can be found at: https://pypi.python.org/pypi/sentiment_classifier

To avoid overfitting, we tune the regularization parameter α on a validation set. For SVM models this parameter corresponds to squared L2 penalty. For ridge models, it corresponds to the strength of L2 regularization term. We experiment with selecting models based on lowest RMSE error as well as the ones with highest disattenuated Pearson correlation score.

B.4.2 Gradient boosting

We apply gradient boosted tree regressors using XGBoost [7] trained on the *designed features* (see Section B.3.2). To train XGBoost models, we use early stopping by evaluating on a validation set with 10,000 estimators and a logarithmic scale grid search of learning rate from 10e$-$5 to 10e$+$5. We experiment with RMSE as well as disattenuated Pearson correlation scores as criterion to perform early stopping.

B.4.3 Feed-forward neural networks

As a second type of non-linear models, we use feed-forward neural networks (FFNNs). We train FFNNs on our *designed features* (see Section B.3.2) expecting that the introduced non-linearity will complement the results of previous models. Our FFNN architecture consists of 3 hidden layers with tanh activation units. We apply dropout regularization of 0.5 between each of the layers. The network has a total of 223 input features in the first layer and 256 neurons in each of the three intermediate hidden layers. We experiment with optimizing for three loss functions:

1. **Mean squared error (MSE)**: this is our default choice used for most of the regressors.
2. **Huber**: Huber loss is less sensitive to outliers which are present in BSAG scores (high BSAG scores for few individuals).
3. **Pearson correlation**: we experiment with correlation loss because it is directly related to the metric used to evaluate the model performance by organizers of shared task A.

B.4.4 Neural sequence encoders

We include two types of models based on neural networks which encode the essays to a low dimensional representation, after which a score is predicted using a feed-forward layer. Essays are encoded using two of the most prevalent neural network architectures for modeling of sequences, convolutional neural networks (CNN) and recurrent neural networks (RNN).

Pretrained embeddings The first layer of NN architectures embeds the one-hot token representations into a vector space of lower dimensionality, which it then fine-tuned through back-propagation. We initialize the embedding layer using embeddings from dedicated word embedding techniques Word2Vec [8] and Glove [9]. This proved to be essential for good performance of the neural sequence models.

CNNs We apply the architecture proposed by [10] which consists of a single convolutional layer with multiple filter sizes, followed by one feed-forward layer over the three-dimensional score vector. We use filters of size 3, 4, 5, 6 and 7 and vary the amount from 64 to 512 filters for each size.

	Anxiety			Depression			Total		
	RMSE	MAE	Diss. R	RMSE	MAE	Diss. R	RMSE	MAE	Diss. R
Development									
Ridge RMSE (lex. feat.)	1.222	0.784	0.2100	1.460	1.076	0.3493	8.356	6.472	0.4532
+Diss. R (lex. feat.)	1.225	0.782	0.2160	1.497	1.138	0.4046	8.643	7.043	0.4783
+RMSE (des. feat.)	1.218	0.773	0.2136	1.446	1.073	0.3781	8.272	6.280	0.4719
+Diss. R (des. feat.)	1.218	0.773	0.2136	1.446	1.073	0.3781	8.272	6.280	0.4719
SVM RMSE (lex. feat.)	1.260	0.690	0.1129	1.517	1.046	0.2542	8.643	5.940	0.4526
+Diss. R (lex. feat.)	1.360	**0.573**	0.1220	1.811	1.007	0.4094	9.047	6.091	0.4624
+RMSE (des. feat.)	1.241	0.723	0.1227	1.470	1.005	0.3736	8.683	6.920	0.3418
+Diss. R (des. feat.)	1.352	**0.573**	0.1026	1.897	1.694	0.3508	8.449	6.019	0.4473
XGBoost RMSE (des. feat.)	1.221	0.769	0.1982	1.452	1.081	0.3624	8.302	6.257	0.4600
+Diss. R (des. feat.)	1.225	0.768	0.1997	1.458	1.073	0.3579	8.312	6.343	0.4557
CNN RMSE loss	1.221	0.772	0.2053	1.473	1.128	0.3863	8.390	6.488	0.4556
RNN RMSE loss	1.228	0.769	0.1630	1.444	1.070	0.3938	8.271	6.206	0.4805
FFNN MSE loss (des. feat.)	**1.216**	0.775	0.2253	1.445	1.073	0.3837	**8.219**	6.310	0.4945
+Huber loss (des. feat.)	1.246	0.697	0.2294	1.483	0.997	0.3921	8.486	**5.884**	0.5000
+Diss. R loss (des. feat.)	1.288	0.616	0.2010	1.675	**0.959**	0.3488	11.556	7.743	0.4290
Ensemble	1.223	0.743	**0.2660**	**1.435**	1.035	**0.4246**	8.252	6.047	**0.5191**
Test Runs									
Submission 1 (Ensemble)	1.119	**0.476**	**0.1946**	**1.393**	**1.004**	**0.4536**	**7.843**	5.691	**0.5667**
Submission 2 (Single Model)	**1.022**	0.697	0.1760	1.403	1.019	0.4192	8.134	**5.688**	0.5140

Table B.1: Results on internal evaluation set for best individual models; "lex. feat." refers to the lexical features (see section B.3.1), whereas "des. feat." are the designed features (see section B.3.2).

RNNs We experiment with two types of RNNs to encode the essays, long short-term memory networks (LSTM) [11] and gated recurrent units (GRU) [12]. After encoding the essay in forward and backward direction, we use the concatenated sequences of hidden states to predict scores. To reduce the dimensionality of this representation, we use max-pooling and self-attention to obtain the final essay encodings [13]. We experiment with single-layer bidirectional RNNs with hidden state vectors of 64, 128 and 256 dimensions. A fully connected layer of 32 and 64 nodes is used to predict scores.

B.4.5 Model ensembling

To produce weighted averages of predictions, we use the *forward model selection* algorithm that greedily selects the combination of models that maximizes the disattenuated Pearson correlation on the evaluation set. We use 100 iterations and choose the best model if there is no improvement after 30 iterations on the evaluation set.

B.5 Experiments

B.5.1 Training details

We divide the training set of 9,217 individual evaluations into two parts: (i) a *train set* consisting of 7,835 examples, and (ii) an *evaluation set* consisting of the rest

	Anxiety	Depression	Total
Ridge RMSE (lex. feat.)	0.2698	0.0625	0.1825
Ridge RMSE (des. feat.)	-	-	-
SVM RMSE (lex. feat.)	-	-	-
SVM RMSE (des. feat.)	0.0688	0.1563	0.0584
XGBoost RMSE (des. feat.)	0.2646	0.0469	0.0949
CNN RMSE loss	0.0423	0.1250	-
RNN RMSE loss	-	0.3281	0.2993
FFNN MSE loss (des. feat.)	-	0.2813	0.0365
FFNN Huber loss (des. feat.)	0.3545	-	0.3285
FFNN Diss. R loss (des. feat.)	-	-	-

Table B.2: Weights of the ensemble components.

(1,382 examples). For SVM, Ridge and XGBoost models, we select the best models on our evaluation set using two metrics: (i) models with the lowers RMSE score, and (ii) models with the highest disattenuated Pearson correlation score. For feed-forward neural nets we experiment with three loss functions: (i) MSE, (ii) Huber, and (iii) disattenuated Pearson correlation. Finally, for neural sequence encoders, we use MSE as a loss function. In order to build an ensemble of models, we further subdivide our evaluation set in two equal parts:

1. **Validation set**: the validation set is used to choose the best combination of models using forward model selection (see Section B.4.5).
2. **Test set**: the test set is used to verify that a given model combination does not overfit the evaluation set.

Before extracting features from the text of input essays, we perform basic text preprocessing functions: lowercasing, removal of punctuation and extra spaces. For TF-IDF and embedding lexical features we also remove the stop words. Additionally, we use TextBlob (https://textblob.readthedocs.io/) in order to correct the spelling mistakes.

Feed-forward neural networks are trained for 100 epochs with learning rate of $1e{-}5$. We also apply a weight decay (L2 penalty) of $1e{-}6$ on the Adam optimizer. Most of the models converge after training approximately for 20 epochs with a batch size of 8.

CNN and RNN models are trained with Adam and early stopping based on disattenuated Pearson correlation. Models converge after training for approximately 10 epochs, with batch size 32. For RNN models we apply a dropout with probability 0.3 on the embedding layer and the output layer. For both CNN and RNN models we apply dropout on the fully connected layer with probability 0.15.

B.5.2 Results

Table B.1 summarizes results for different models on our validation set. For linear models, we notice that SVM models are sensitive to optimizing towards RMSE or disattenuated correlation score. We also observe that SVM models have lower disattenuated correlation scores for the anxiety BSAG metric. For feed-forward neural nets, use of the Huber loss obtains the best performance. We speculate that

this is because this method is not as influenced by outliers as other loss functions. The rest of the models has approximately similar performance.

A large boost in performance is observed when creating ensembles of models. We gain between 0.02 and 0.04 points on our validation set for the disattenuated correlation metric. We don't see this improvement on RMSE and MAE metrics since our ensemble is greedily built to optimize for Pearson correlation between predicted and ground truth results.

Table B.2 shows the weight combinations of our ensemble for all three objectives to predict. We only add best RMSE models for Ridge, SVM and XGBoost regressors. The reason is that adding models that had the best performance on Pearson disattenuated correlation score decreased significantly the RMSE and MAE scores of the ensemble. How these models can still be added without producing this drop in performance is left for future work.

The bottom rows of Table B.1 show the results of our two submissions on the official CLPsych test collection. We obtain a considerable improvement using ensembles of models with respect to our single best model submission, resulting in the overall second best submission. We speculate that this is because of different score distributions produced by dissimilar models used in this work. This generates low correlation of individual model predictions, which results in better ensembles. We were surprised to see that disattenuated correlation score was several points higher in depression and total BSAG predictions than on our internal validation set. The anxiety score, on the other hand, is considerably lower. Further analysis is needed to understand these differences, and to investigate the impact of the individual types of hand-designed features.

B.6 Conclusion and future work

In this paper we briefly described the Ghent University – IDLab submission to the CLPsych 2018 shared task A. We found that linear models, gradient boosting as well as neural network based models perform similarly but produce different models that, when combined, can increase the performance on the test set considerably.

For future work, we plan to conduct a careful error analysis (e.g. ablation tests) and examine the best ways to design our train-validation splits in order to decrease the score difference between the validation and test sets. We also plan to experiment with more sophisticated ways of ensembling and stacking techniques.

We consider that in the end, most of the success of this task comes down to designing a good set of features. In particular, one of the features we didn't explore is topic modeling. Additional features can be obtained from topic model distributions as they provide positive results on similar tasks described in [14] and [15].

Finally, another direction we want to explore consists of using word and phrase embeddings, pre-trained on a corpus of individuals with psychological disorders. Some work has already been done to gather this kind of corpus from online resources (Twitter and Reddit in particular) [16] and [17]. We hypothesize that we can get a significant improvement by initializing our CNN and RNN models with these embeddings.

Acknowledgments

We are grateful to Giannis Bekoulis for fruitful discussions on model cross-validation and for providing resources, support and encouragement.

Human Subjects Review

This study was evaluated by the Ethics Committee of the faculty of Psychology and Educational Sciences of Ghent University, which concluded that ethical approval was not needed for the research conducted for this manuscript.

References

[1] C. Power and J. Elliott. *Cohort profile: 1958 British birth cohort (national child development study)*. International journal of epidemiology, 35(1):34–41, 2005.

[2] P. Shepherd. *Bristol social adjustment guides at 7 and 11 years*. Centre for Longitudinal Studies, 2013.

[3] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. *The development and psychometric properties of LIWC2015*. Technical report, 2015.

[4] J. Staiano and M. Guerini. *DepecheMood: a lexicon for emotion analysis from crowd-annotated news*. arXiv preprint arXiv:1405.1605, 2014.

[5] T. Cagan, S. L. Frank, and R. Tsarfaty. *Generating subjective responses to opinionated articles in social media: an agenda-driven architecture and a Turing-like test*. In Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media, pages 58–67, 2014.

[6] S. Merity, N. S. Keskar, and R. Socher. *Regularizing and Optimizing LSTM Language Models*. arXiv preprint arXiv:1708.02182, 2017.

[7] T. Chen and C. Guestrin. *XGBoost: A Scalable Tree Boosting System*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. Available from: http://doi.acm.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. *Distributed representations of words and phrases and their compositionality*. In Advances in neural information processing systems, pages 3111–3119, 2013.

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

[10] Y. Kim. *Convolutional Neural Networks for Sentence Classification*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, 2014.

[11] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural computation, 9(8):1735–1780, 1997.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. arXiv preprint arXiv:1406.1078, 2014.

[13] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. *A structured self-attentive sentence embedding*. arXiv preprint arXiv:1703.03130, 2017.

[14] P. Resnik, W. Armstrong, L. Claudino, and T. Nguyen. *The University of Maryland CLPsych 2015 shared task system*. In Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 54–60, 2015.

[15] A. Cohan, S. Young, and N. Goharian. *Triaging mental health forum posts*. In Proceedings of the Third Workshop on Computational Lingusitics and Clinical Psychology, pages 143–147, 2016.

[16] A. Yates, A. Cohan, and N. Goharian. *Depression and Self-Harm Risk Assessment in Online Forums*. arXiv preprint arXiv:1709.01848, 2017.

[17] G. Coppersmith, M. Dredze, C. Harman, K. Hollingshead, and M. Mitchell. *CLPsych 2015 shared task: Depression and PTSD on Twitter*. In Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 31–39, 2015.