

RADBOUD UNIVERSITY NIJMEGEN



FACULTY OF SCIENCE

---

# Thesis Title

THESIS SUBTITLE

---

THESIS MSc COMPUTING SCIENCE

*Supervisor:*  
Faegheh HASIBI

*Author:*  
Gizem AYDIN

*External Supervisor:*  
S. Amin TABATABAEI

*Second reader:*  
name SURNAME

date

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Task Definition . . . . .	3
1.2	Related Work . . . . .	4
1.2.1	Funding Information Extraction . . . . .	4
1.2.2	Entity Linking and Subtasks . . . . .	4
1.2.3	Domain-Specific Systems . . . . .	6
1.2.4	Entity Representation . . . . .	8
1.2.5	Using Pretrained Language Models in Domain-Specific Applications	9
<b>2</b>	<b>Experimental Setup</b>	<b>10</b>
2.1	Data . . . . .	10
2.1.1	Pretraining BERT . . . . .	11
2.1.2	Knowledge Repository . . . . .	11
2.2	Evaluation . . . . .	11
2.3	Domain-Adaptation of BERT . . . . .	11
2.4	Named Entity Recognition . . . . .	12
2.4.1	Preprocessing . . . . .	12
2.4.2	Approach . . . . .	13
2.5	Hardware . . . . .	13
<b>3</b>	<b>Results</b>	<b>14</b>
3.1	BERT <sub>SC</sub> . . . . .	14
3.2	Named Entity Recognition . . . . .	14
<b>4</b>	<b>Discussion</b>	<b>15</b>
4.1	BERT <sub>SC</sub> . . . . .	15
4.1.1	BERT or Flair for NER . . . . .	15
<b>5</b>	<b>Conclusion</b>	<b>16</b>
<b>A</b>	<b>NER Data Preprocessing</b>	<b>23</b>

## **Abstract**

This is the abstract

# Chapter 1

## Introduction

Automatic extraction of funding information from academic articles has been an interesting subject for researchers, and various approaches have been proposed for this purpose [26, 53, 7]. Annotating articles with their corresponding funding information adds significant value to the research community, such as enabling organizations to track the outcome of the research they funded [26] and aiding the compliance of open access rules [7]. However, this task is far from trivial, and there is still room for improvement.

Funding information extraction contains subtasks in itself. These can roughly be summarized as:

- (i) isolating the piece of text that contains the funding information from the articles,
- (ii) extracting mentions of funding organizations and grant numbers from the selected text,
- (iii) linking the funding organization mentions to the corresponding entities in a specific Knowledge Repository to determine which funder is being acknowledged,
- (iv) linking grant numbers to the respective funder mention to decide which grant number belongs to which funder.

In this thesis, the aim is to develop a neural Entity Linker for funding domain, hence tackling subtasks (ii) and (iii).

Entity Linking (EL) is the task of annotating text with corresponding entity identifiers from a Knowledge Repository (KR) [2]. EL entails Named Entity Recognition (NER) and Named Entity Disambiguation (ED), where the former corresponds to detecting mentions and their respective types from the text and the latter corresponds to linking the mentions to a KR [2].

A large amount of literature addresses EL, NER and ED using neural approaches, which includes the state-of-the-art methods [47, 57, 34] as well. However, the bigger part of this work focuses on performing these tasks in general-domain, often times considering Wikipedia pages as entities [10]. Hence, there is no guarantee that these approaches will perform well in funding domain.

There are significant amount of differences between performing EL in general-domain and in funding domain. First of all, a different knowledge repository is needed as most of the smaller funders do not exist in general-purpose knowledge repositories. Another challenge is the fact that some information that is deemed highly important in general-domain ED may not be as informative in this domain. For example, [40] reported that ED would almost be solved if entity type information could be estimated better. However, in funding domain, some of the most ambiguous mentions are the ones referring to ministries, as a lot of countries may have a ministry with the same name. In these cases, the type of the mentions being, for example, governmental organizations would not provide any clue.

Funding domain introduces additional challenges for the NER task as well. For example, the mentions of organizations that are not funders should not be extracted. Another challenge is the limited amount of labelled data. In general-domain setting, Wikipedia could be exploited to extract millions of samples [6]. Supervised neural architectures for NER require a large amount of training data to obtain high performance [55].

Considering the mentioned differences, this thesis aims to answer the following research question:

*RQ: Is it possible to use neural approaches for Entity Linking in funding domain, where labelled data is limited and a domain-specific knowledge repository is used?*

Explain how the research question will be answered and that the data/resources from Elsevier will be used.

Explain what each section entails. Add stuff from Radboud Writing Lab session

## 1.1 Task Definition

In this thesis, the aim is to perform Entity Linking (EL) in funding domain. Typically, EL consists of two subtasks, Named Entity Recognition (NER) and Entity Disambiguation (ED).

Given a piece of text  $d = \{x_1, \dots, x_N\}$  of  $N$  tokens, the task of NER is to identify a set of spans  $M = \{m_i \mid m_i = \{x_s, \dots, x_e\}, s \geq 1 \wedge s \leq e \wedge N \geq e\}$ , where each span corresponds to an object, person or even a piece of information of interest. These spans are called named entities, or mentions. Usually, detecting the types of the extracted mentions is also part of the NER task. The available mention types differ among systems based on their objectives. In this thesis, each mention  $m_i$  corresponds to either a *Funding Organization* (*ORG*) or a *Grant Number* (*GRT*). Hence,  $type(m_i) \in \{ORG, GRT\}$ . Throughout this research, it is assumed that the mentions are not nested and are not overlapping.

Given the set of mentions  $M$ , the aim of ED is to link each mention to their corresponding entity  $e_i \in \mathcal{E}$  in a knowledge repository. A knowledge repository is a collection of entities, and may contain information on the entities and relations between them. The contents and specifications of the knowledge repository used is task-dependent. The task of ED is not trivial as many different mentions may be used to refer to the same entity, or the same mention may be used to refer to different entities. When the correct link of a mention  $m_i$  is entity  $e_i$ , this will be denoted as  $link(m_i) = e_i$ . Another thing to note is that some mentions may not correspond to any entity in the target knowledge repository. These mentions are usually referred to as *NIL Mentions*. To denote these cases, a NIL entity  $\emptyset \in \mathcal{E}$  will be used. In this thesis, a knowledge repository of funding organizations is used, hence, only the mentions with type *ORG* will be considered for ED.

Sometimes, the set of entities  $\mathcal{E}$  may be extremely large. In that case, a Candidate Selector (CS) may be used to limit the search space. The aim of the CS is to extract a set of candidate entities  $C_i = \{e_1, \dots, e_K\} \subset \mathcal{E}$  for a given mention  $m_i$ . Usually the size of  $C_i$  is much smaller than that of  $\mathcal{E}$ . This enables using more complex algorithms for ED as it reduces the number of entities to consider.

Lastly the task of EL, which is the aim of this thesis, is to extract a set of mention-entity pairs from an input text  $d$ . In this thesis, as only the *ORG* mentions will be linked, the objective can be defined as extracting the set  $T = E_{ORG} \cup M_{GRT}$  for each input  $d$ , where,

$$E_{ORG} = \{(m_i, e_i) \mid m_i \in M \wedge e_i \in \mathcal{E} \wedge link(m_i) = e_i \wedge type(m_i) = ORG\} \quad (1.1)$$

and

$$M_{GRT} = \{m_i \mid m_i \in M \wedge type(m_i) = GRT\}. \quad (1.2)$$

## 1.2 Related Work

There is a large amount of literature on Entity Linking and its subtasks, Named Entity Recognition and Named Entity Disambiguation. The bigger part of this literature focuses on performing these tasks in the general-domain setting, often times considering Wikipedia pages as entities [10]. While this line of research introduces the state-of-the-art approaches, there is no guarantee that these approaches will perform well in a domain-specific setting with a custom knowledge repository. Hence, domain-specific literature is also investigated to get insights on adapting general-purpose methods to specific domains.

Section 1.2.1 reviews the literature on automatic funding information extraction from academic articles. In Section 1.2.2, state-of-the-art general-purpose NER, ED and EL solutions are presented. Domain-specific neural NER, ED and EL approaches are demonstrated in Section 1.2.3. Section 1.2.4 concentrates on entity representations in neural ED, mainly entity embeddings, and lastly, Section 1.2.5 reviews the literature on using pretrained neural language models in domain-specific applications.

### 1.2.1 Funding Information Extraction

One of the most notable work on automatically extracting funding information from text is FundingFinder [26]. FundingFinder is a two-step pipeline that utilizes NLP techniques. In the first step, the paragraphs that contain funding information are determined, and in the second step, NER is performed using an ensemble of different Sequential Learning approaches. The authors also created a publicly available benchmark dataset for this task. The approach used in this thesis builds upon this work, keeping the first step intact while improving the second step, and adding the ED capability.

Before FundingFinder, not much literature existed on extracting funding information from text automatically, and the existing work mostly utilized regular expressions [26]. Recently, there have been more approaches presented to tackle this problem. In 2020, Wu et al. proposed AckExtract [53], which extracts funder organization mentions from the COVID-19 Open Research Dataset [49]. For NER, they use a pretrained neural model from the package Stanza [39], which uses Contextual String Embeddings [1]. However, their method does not include any ED, whereas in this thesis, one of the tasks is to link the funder mentions to their corresponding entities in the domain-specific Knowledge Repository. Another approach is proposed in 2021, GrantExtractor [7], which extracts funding information from articles in biomedical literature, in the form of grant numbers and their corresponding organizations. For extracting grant numbers, they train a BiLSTM-CRF [23] architecture. Using a multi-class classifier, they determine which organization the extracted grant number belongs to. They do not use any neural approaches for extracting organization mentions. Also, the focus of GrantExtractor is on linking grant numbers to their respective organizations, while the focus of this thesis includes extracting all funding organizations that financially supported the corresponding research, even though no grant information is acknowledged.

### 1.2.2 Entity Linking and Subtasks

In this section, the current state-of-the-art methods for EL and its subtasks are investigated <sup>1</sup>.

---

<sup>1</sup>[http://nlpprogress.com/english/entity\\_linking.html](http://nlpprogress.com/english/entity_linking.html)

## Named Entity Recognition

In the past couple of years, Deep Learning has been a popular choice to tackle the NER problem, and the corresponding research has improved the state-of-the-art results [55]. In 2018, Akbik et al. proposed Contextual String Embeddings [1], which represents words using a character-level neural language model, and is able to produce different word embeddings depending on the context. By utilizing a BiLSTM-CRF architecture that takes the concatenation of Contextual String Embeddings and pretrained GloVe embeddings [37] as input, they report state-of-the-art results in both German and English NER, in the CoNLL-2003 [46] setup. In 2019, Devlin et al. introduced BERT [8] which obtained new state-of-the-art results on several tasks. In English CoNLL-2003, they obtained an F1 score of 92.8% using the cased version of BERT<sub>BASE</sub> [8], performing very close to [1], which obtained 93.1%.

Some approaches for NER utilize external resources, such as a list of entity names, which may be called a dictionary or a gazetteer. This may boost the performance of the system, but may also hurt the generalization ability [55]. However, there have been various models presented [31, 52] that incorporate this information while performing comparable to [1]. With this approach, both papers [31, 52] aim to improve the performance on entities that do not appear in the training set or that are rare.

Recently, Yamada et al. (2020) proposed LUKE [57], a contextualized representation for both words and entities, to be used in entity-related tasks. LUKE is based on the bidirectional transformer [48], however, it treats words and entities as independent tokens. For this purpose, the authors propose a modified attention mechanism as well as a new training methodology based on BERT’s [8] masked training. They pretrain LUKE using a large entity-annotated Wikipedia corpus. By using the proposed embeddings, they report the new state-of-the-art results for NER, improving upon [1].

## Named Entity Disambiguation

In 2018, Raiman and Raiman proposed DeepType [40], an ED approach. A neural network that is constrained by the predicted type information for a given entity. By using the type information, they also reduce the complexity of disambiguation from polynomial to linear. DeepType produced the state-of-the-art results in three ED datasets, by obtaining scores of 92.36%, 94.88% and 90.85% on WikiDisamb30 [12], CoNLL (YAGO) [21] and TAC-KBP-2010<sup>2</sup> respectively. The authors also note that DeepType can reach 99.0% and 98.6% accuracy on CoNLL (YAGO) and TAC-KBP-2010, when the type information is provided by an Oracle. Based on that, they claim the ED problem can almost be solved if the type classifier is improved. However, in the case of funding domain, most of the ambiguities in mentions cannot be solved by using the type information. For example, the mention “Ministry of Health”, can be resolved to different entities corresponding to ministries in different countries, however, the types of these entities would be the same.

The paper proposed by Mulang’ et al. (2020) [34] slightly advances the state-of-the-art ED results for CoNLL (YAGO) [21] dataset by obtaining a score of 94.94%. The authors introduce the idea of incorporating context derived from Knowledge Graphs (KG) to pretrained transformers with the aim of improving their performance for ED. They extract triplets from the KG, verbalize them into natural language form, and append them to the input sentence and mention before passing it through the transformer. When they replace the Wikipedia description used in the DCA-SL model [59] with the structured KG context they extracted, they obtain the above-mentioned score.

---

<sup>2</sup><https://tac.nist.gov/>



Another interesting approach is proposed by Wu et al. in 2020 [54], which outperforms DeepType [40] in TAC-KBP-2010 by obtaining a 94.5% accuracy. Their method also achieves state-of-the-art results in the zero-shot Entity Linking dataset derived from WikilinksNED [36]. To perform ED, they only use textual information and architectures that utilize pretrained BERT [8] transformers. They represent the mention using itself and its context, and the entity using its description. Using a bi-encoder [24], they encode the mention and entity representations in the same space, which they later use to extract candidates for a given mention using approximate nearest neighbor search. They make the final decision by passing representations of the candidate entities and mentions through a cross-encoder [24].

## Entity Linking

Kolitsas et al. (2018) [27] proposed the first end-to-end neural EL system in 2018, and recorded state-of-the-art results in AIDA CoNLL dataset [21]. By tackling NER and ED jointly, the authors aim to utilize the dependency between these two tasks. They suggest that this has several benefits, such as improved mention boundary recognition. Their method first extracts all possible mention spans from the input. Then, the model computes a score for each mention - candidate entity pair, using pretrained entity embeddings [16], context-aware mention representations, commonness and long range attention scores. The final output for the input text is based on these scores and global entity coherence.

In 2020, van Hulst et al. proposed REL [47], an EL toolkit that utilizes state-of-the-art NLP research, outperforming [27] in terms of micro-F1 score in AIDA CoNLL dataset [21]. REL tackles the EL problem in three steps: NER, candidate selection and ED. For NER, they utilize Flair, namely, the sequence labelling architecture and Contextual String Embeddings proposed by [1]. For each mention, up to 4 candidates are selected using commonness, and up to 3 candidates are selected based on the similarity between the context of the mention and entity embeddings. Entity embeddings are provided by [16], the same ones used in [27]. For ED, Ment-norm by Le and Titov [29] is used. Apart from obtaining state-of-the-art results, REL also offers a modular architecture, allowing easy replacement of components and it does not require a GPU during inference time [47].

Broscheit (2020) [5] proposed an architecture that jointly does NER and ED using BERT [8]. In this approach, the task of EL is framed as a per-token multi-class classification problem. The model utilizes a pretrained BERT model and an output classification layer on top of it. Even though this approach is a big simplification on the EL task, it performs only a few percents off compared to [27]. However, as each entity is cast as a class, the model cannot disambiguate unseen entities. In real-world, knowledge repositories keep growing, and hence it is important for the system to be extendable for entities that do not exist in the training set [19]. Moreover, some training datasets may not cover the whole entity vocabulary, such as the one used in this thesis.

### 1.2.3 Domain-Specific Systems

In this section, the research that aims to tackle EL, NER and ED in a domain-specific setting with neural architectures will be reviewed.

For NER, there is a great amount of research in general domain, however, more research on domain-specific solutions is expected for supporting real-world applications [61]. Existing NER approaches rely on a large amount of annotated data, which may not be available for the domain-specific setting [43]. Hence, Shang et al. (2018) [43] proposed AutoNER, a neural architecture that is designed to learn from data that is created

by distant supervision, without any human effort. AutoNER uses domain-specific dictionaries to automatically generate labelled data with distant supervision. The authors also introduce the “Tie or Break” tagging schema, that is based on predicting whether two adjacent tokens belong to the same mention or not. They reason that this tagging scheme is suitable to use noisy labels generated by distant supervision. They show the effectiveness of their work in multiple datasets, two of them being the BC5CDR [30] and NCBI-Disease [9] datasets from the biomedical domain, in which AutoNER achieves 84.8% and 75.52% F1-score respectively.

Another domain-specific NER approach that utilizes a dictionary is proposed by Wang et al. (2019) [50], which tackles the Clinical NER problem in Chinese text. They show the effect of incorporating dictionary knowledge in the BiLSTM-CRF [23] architecture on rare and unseen entities experimentally. Also, they suggest five different methods of using dictionaries in this context and compare the results.

There also exist research on tackling the domain-specific ED problem with neural architectures. In 2019, Mondal et al. [33] proposed a system that is based on string similarity to perform ED on disease names. The authors utilize a two-step solution. First, for each mention, they extract a set of candidate entities based on Jaccard overlap and the cosine similarity between the entity label and the mention. They use word embeddings to calculate the cosine similarity. For multi-word strings, they sum the embeddings for each word. Then, they rank the candidate entities with a Triplet Network [22], that learns to reduce the distance of the mention with the positive candidate, while increasing the distance with the negative candidate. As an input to this network, word embeddings is used again to represent the mention and the candidate entity’s label. With this approach, they obtain 90% accuracy on the NCBI-Disease [9] dataset, outperforming previous approaches.

The input representation of entities vary between different approaches. In another research tackling clinical ED by Schumacher et al. (2020) [42], different from [33], multi-word strings are represented by two different methodologies, maxpooling over the word embeddings and running self-attention. The authors report that running self-attention produces better results compared to maxpooling. Instead of combining different vectors, Sung et al. (2020) [44] represents each mention and each synonym of an entity using two different vectors, one that is based on TF-IDF scores, and another that is based on BERT [8]. For the latter, the [CLS] token is used to represent the whole phrase. Instead of combining these two vectors, they define two different similarity function for each. The final similarity of a mention and an entity synonym is defined based on the weighted average of these two similarity scores.

Architectures utilizing the type information are also present. Zhu et al. (2020) introduced LATTE [63], an architecture for ED in medical domain. The authors emphasize the importance of fine-grained types in their setting, and as this information is not available, they model the fine-grained types as latent variables. For ED, in addition to the latent fine-grained types, they use the similarity between the entity’s label, and the mention and its context. To train their model, they use multi-task learning for both type classification and ED.

Some proposed methodologies tackle the EL problem as a whole in a domain-specific setting using neural architectures. For biomedical domain, Zhao et al. (2019) [62] proposed a joint neural architecture for NER and ED tasks for performing EL. Their architecture utilizes explicit feedback between the two tasks in a multi-task learning setting. With this architecture, they obtain F1 scores of 87.43% and 88.23% in NER and ED tasks of the NCBI-Disease [9] dataset respectively, and 87.62% and 89.17% in BC5CDR [30] dataset. With these numbers, they outperform AutoNER [43] in NER setting for both datasets, and perform comparable to [33] in NCBI-Disease dataset for ED.

Biomedical domain is not the only one in which neural approaches are used for NER, ED and EL. In 2019, Espejo-Garcia et al. [11] proposed a solution to extract named entities that refer to the important parts of phytosanitary regulations, which is related to the agricultural domain. The authors experimented with eight different state-of-the-art neural architectures. For their setting, the best performing architecture was a bidirectional LSTM [18] that utilized a Softmax layer for inference and got the concatenation of pretrained Word2Vec [32] embeddings with character based word representations as input. With this architecture, they obtained an F1 Score of 88.3%. Apart from agricultural domain, Yang et al. (2020) [58] proposed Headword Oriented Entity Linking, an EL setting where the mention scopes do not need to be identified, to extract cosmetic products from blogs and to disambiguate them to a domain-specific knowledge repository. First, using word segmentation techniques, they identify the headwords of the mentions. Then, they apply classification on the mentions to decide whether the mention can be linked to a product that is in the knowledge repository. Lastly, they use a modified version of the architecture proposed by [19] for ED. Another interesting study is by Kurz et al. (2020) [28], where they experiment with different BERT-based [8] architectures to disambiguate mentions of machine parts and errors belonging to German technical service tickets.

#### 1.2.4 Entity Representation

In neural ED, entity representation plays an important role. Some research frames the problem as multi-class classification and represent entities as different classes [5, 51, 62], while other research tends to use architectures that takes properties of entities as input and learns a representation internally during training for ED [54], implicitly or explicitly. Another line of research utilizes entity embeddings [47, 27, 59, 33]. In this section, the literature on entity embeddings will be reviewed.

There is a large body of literature on embedding entities and relations found in knowledge repositories. One of the most notable work is TransE [3], introduced in 2013. TransE generates embeddings for each entity and relation in the input knowledge repository. The idea behind TransE is to model relations as translations in the embedding space. For a given triplet  $(h, l, t)$  in the knowledge repository where  $h$ ,  $l$  and  $t$  denote head entity, relation and tail entity respectively; TransE aims to make the embedding of  $t$  as close as possible to the sum of the embeddings of  $h$  and  $l$ , using an energy-based model. Later on, there has been models that improved upon TransE such as TransH, TransR, CTransR and TransD, each improving upon the previously proposed one respectively [25]. Another interesting work that generates entity embeddings utilizing the triplets in knowledge repositories is RDF2Vec [41]. RDF2Vec extracts graph sub-structures, and treats them as sentences to train a Word2Vec [32] model.

In 2017, Gupta et al. [19] proposed a neural architecture that can generate entity embeddings that jointly encodes the information on the entity’s description, the context of its mentions and its type. The architecture consists of three models that encode the different information. The parameters of the models and the embeddings are jointly learned based on the sum of four different losses that ensure the entity embeddings and the encoded information is similar. The summation guarantees that entity embeddings can be generated even though some information is missing, such as the description [19]. They also proposed an ED model based on these embeddings. As mentioned in Section 1.2.3, a modification of this architecture is used to create entity embeddings in [58]. Another interesting neural approach for generating entity embeddings was proposed by Ganea and Hofmann in 2017 [16], and, their pretrained entity embeddings have been used by research that reported state-of-the-art results in EL [47, 27]. In their approach, the words and entities are embedded in the same space by extending a pretrained Word2Vec

model [32] to cover entities as well. They generate the entity embeddings such that they are close to the vectors of the words that occur in the corresponding entity descriptions and in the mention contexts that belong to the entity.

Gillick et al. (2019) [17] proposed a dual encoder architecture for ED, which also learns entity embeddings. The model has one encoder to encode the mention and its context, another encoder to encode the entity using its description and categories. Then, the model is trained to maximize the cosine similarity between the encoded representations of the correct mention - entity pairs. After training, the entity embeddings are precomputed using the entity encoder and stored for inference.

Another successful method to get entity embeddings is Wikipedia2Vec [56]. They also have pretrained embeddings available for use. Wikipedia2Vec encodes words and entities in the same space and utilizes Word2Vec [32]. To train Wikipedia2Vec, they use three models. Word-based skip gram model puts the embeddings of words that occur in similar context close, anchor context model puts the embeddings of entities close to embeddings of words that occur near the anchor texts of the entity, and lastly, link graph model puts the entity embeddings close based on Wikipedia’s hyperlink graph.

### 1.2.5 Using Pretrained Language Models in Domain-Specific Applications

BERT [8], and other pretrained language models have been used extensively in various NLP applications and have obtained state-of-the-art results in benchmark tasks [35]. However, for some domains, they may be too generic and may not be able to cover specific needs [4]. Hence, it may be worthwhile to adapt the pretrained language model that will be used to the specific domain. Gururangan et al. (2020) [20] shows that a second-round of pretraining of a pretrained language model improves the performance in both low and high resource settings, using different domains and tasks. Also, Fraser et al. (2019) [14], reports that language models which are pretrained with domain-specific text perform better on the task of NER in biomedical domain. However, it should be noted that training a neural language model from scratch for a specific domain can take weeks [60].

There is emerging research on cheap domain adaptation of pretrained language models. Gururangan et al. (2020) [20] proposed Task-Adaptive Pretraining (TAPT), which corresponds to pretraining the neural language models further with the unlabeled data of the task at hand. They report that this approach performs comparable to pretraining with larger amount of text from the same domain. Tai et al. (2020) proposed exBERT [45], a low-cost method to add new domain-related words to the vocabulary of BERT [8] while not changing its weights. In addition, Poerner et al. (2020) [38] introduced a CPU-only domain adaptation method, where a Word2Vec [32] model is trained on the domain-specific data, and the embedding vectors of the pretrained language model is aligned based on that.

## Chapter 2

# Experimental Setup

To tackle the Entity Linking problem in funding domain, different experiments were defined. First, a pretrained BERT [8] model was further pretrained using the relevant sentences. After that, separate Named Entity Recognition and Named Entity Disambiguation components were developed, and the possible entity representations were investigated. Lastly, the end-to-end performance of the best performing components were measured, and is compared with a single neural end-to-end architecture. The following sections present the data used and the experiments conducted.

### 2.1 Data

The dataset for funding data extraction and the knowledge repository for ED used in this research is provided by Elsevier B.V.<sup>1</sup>. The dataset consists of a set of labelled articles annotated by humans. To create this dataset, each article is annotated by three people. First, two annotators extracted the funding information from the articles independently. Then, a third annotator harmonized the decisions of the previous two annotators, resolving the conflicts if necessary.

For developing models and evaluating various approaches, the dataset is divided into five subsets: *Training*, *Validation*<sup>1</sup>, *Validation*<sup>2</sup>, *Test*<sup>1</sup> and *Test*<sup>2</sup>. The *Training* split is used to train the models. *Validation*<sup>1</sup> split is used to monitor the progress of training, while *Validation*<sup>2</sup> split is used to select the best approach for each task. *Test*<sup>1</sup> is used to evaluate the performance of each selected component and *Test*<sup>2</sup> is used to evaluate only the best end-to-end Entity Linking solution. The reason why there is such distinction in the testing sets is because *Test*<sup>2</sup> is the official evaluation split for funding data extraction pipeline at Elsevier, and hence is used very carefully. Another important point to note is that there may be a difference in data distributions between *Test*<sup>1</sup> and the other splits, as the former has been annotated 2 years after the creation of the rest of the dataset.

The second column of Table 2.1 shows the number of articles contained in each split.

As a preprocessing step, the methodology of [26] is used to identify the sentences containing funding information. The third column of Table 2.1 shows the number of sentences extracted from each dataset split, and the second column shows the number of articles with at least one positive sentence. Furthermore, the number of organization and grant mentions on each split can be found in the fourth and fifth columns respectively. The last column shows the number of organization mentions that are linked to their corresponding entities in the knowledge repository.

---

<sup>1</sup>??

Dataset Split	#Articles	#Articles Min.1 Sent.	#Sentences	#Org. Mentions	#Grt. Mentions	#Org. Links
Training	37,484	22,720	26,132	67,671	45,263	?
Validation <sup>1</sup>	1,000	1,000	1,284	4,333	2,770	?
Validation <sup>2</sup>	4,000	4,000	5,012	16,355	10,112	?
Test <sup>1</sup>	5,921	4,640	5,500	13,383	10,420	?
Test <sup>2</sup>	19,920	13,851	15,590	37,495	25,349	?

Table 2.1: Dataset splits and statistics. For each split; number of articles, number of articles with at least one funding sentence, number of sentences, number of mentions and links are shown.

### 2.1.1 Pretraining BERT

Following previous research (see Section 1.2.5), a pretrained BERT model [8] is pre-trained further for domain adaptation with Masked Language Modeling (MLM) [cite](#), using the sentences that are identified to be containing funding information. In this stage, apart from the articles in the *Training* split, an additional 22,115 articles are used, increasing the number of training sentences to 40,088. This was made possible by the fact that MLM does not need labeled data. In fact, it is possible to utilize even more articles, however, this is left to future work due to time constraints and computational requirements.

### 2.1.2 Knowledge Repository

## 2.2 Evaluation

As this research contains training models for various tasks, different evaluation metrics are used. To pretrain BERT for the problem domain, an MLM objective is used. That is, for each sentence, some words were masked randomly and the model was taught to predict the masked word using the context around it [8]. To monitor the quality of this training, Perplexity is used. This metric corresponds to the inverse probability of the dataset based on the model [15], and it is the most popular metric to evaluate language models [15].

To evaluate the Named Entity Recognition task, precision recall and F1 scores are used for each entity type, *Organization* and *Grant*. These metrics are defined in terms of True Positives (TP), False Positives (FP) and False Negatives (FN). Precision is defined as the fraction of TPs among all mentions extracted by the system, and recall is defined as the fraction TPs among all ground truth mentions. F1 score is the harmonic mean of precision and recall metrics. A mention is considered to be a TP if and only if both the extracted span and type information is correct. A FP corresponds to a mention that is extracted by the system wrongly, and a FN corresponds to a mention that is not extracted by the system while being present in the ground truth. This scheme is chosen as it is inline with evaluation of the CoNLL-2003 NER task [46].

## 2.3 Domain-Adaptation of BERT

Previous work suggests that pretraining a BERT model with in-domain data improves the overall performance of the model for the task at hand [\[1\]](#). For this purpose, instead of using a BERT model that is trained on a generic corpus, a more domain-relevant BERT, denoted by BERT<sub>SC</sub>, is trained. The choice of terminology is attributed to the

fact that the training data consists of a subset of articles that can be found in Scopus<sup>2</sup>.

To pretrain BERT<sub>SC</sub>, Task-Adaptive Pretraining (TAPT) schema proposed by [20] is used. The idea behind TAPT is to pretrain a BERT model, which was pretrained on a generic dataset, further using unlabelled data from the specified task with MLM objective. The authors compare this approach with Domain-Adaptive Pretraining (DAPT), which they define as pretraining a BERT model from scratch using documents from a specific domain. DAPT is much more expensive in terms of both data and computational power compared to TAPT, and yet the authors show that TAPT performs comparable to DAPT. Although there are other works on adapting BERT to a specific domain in an inexpensive way [1], TAPT was chosen due to its easy-to-use, open-source implementation<sup>3</sup>.

The weights of BERT<sub>SC</sub> were first initialized from "bert-base-cased"<sup>4</sup>, and then using a MLM prediction head [1], the whole model was fine-tuned end-to-end with the extended *Training* split (see Section 2.1.1), using the hyperparameter settings recommended by [20]. On top of that, at the end of each epoch, the resulting model was saved, and the model which had the lowest perplexity score on *Validation*<sup>1</sup> split was chosen to be BERT<sub>SC</sub> at the end. To reduce the runtime, instead of pretraining for 100 epochs as suggested, training was stopped when no improvement in *Validation*<sup>1</sup> was made for 4 consecutive epochs. The choice of a case-preserving model is due to the fact that case information can provide important information to the NER task, for example, it is common in English to capitalize organization names.

## 2.4 Named Entity Recognition

To extract mentions of funding organizations and grant numbers from sentences, a Named Entity Recognition system is developed. Section 2.4.1 introduces the methodology to preprocess the data to convert it to a format suitable for the NER task, and Section 2.4.2 explains the models that were experimented with.

### 2.4.1 Preprocessing

The NER problem is cast as a token classification task using the IOB tagging schema. In this schema, the initial tokens of the mentions are tagged with a "B" ("Beginning") and the remainder tokens are tagged with an "I" ("Inside"). The tokens that are not a part of any mention are tagged with "O" ("Outside"). In NER, there may be different types of mentions. In that case, the type information is appended after the "B" and "I" tags. Since there are two types, *Organization* and *Grant*, a total of 5 tags are used: {B-ORG, I-ORG, B-GRT, I-GRT, O}.

In the dataset used for this task, the annotations are not done in terms of tokens, but in terms of character spans of the input text. That is, each gold mention is provided using their character offsets with respect to the article text. Hence, first the input text is tokenized and the labels are assigned to tokens based on some predefined rules to tackle some edge cases that mostly correspond to annotation errors. These rules are extracted based on empirical results to maximize the correctness of the annotations. The experiments were done on a portion of the training set, and all the edge cases found were present for less than 0.5% of the investigated dataset. In Appendix A you may find the details of the labelling step.

---

<sup>2</sup>

<sup>3</sup>

<sup>4</sup>

### 2.4.2 Approach

For the NER component, several models were implemented and tested. These models can be separated as Flair-based [1] and BERT-based [8] models. The choice of experimenting with Flair was due to its success in NER task. Moreover, Flair is used by REL [47], which achieved state-of-the-art results on Entity Linking, and AckExtract [53], a system for extracting funder organization mentions. Contextual String Embeddings, the embeddings that is the core of Flair, consist of concatenation of vectors from a Left-to-Right and a Right-to-Left language model. However, [8] reports that BERT is inherently more powerful than such language models as it is using MLM objective, that enables it to train a single representation for which both right and left contexts are used. Because of this claim and the recent popularity of BERT models, it is decided to experiment with BERT-based models as well.

Explain Flair-based model

We tried both BERT<sub>SC</sub> and "bert-base-cased". Lastly, as the recent research showed the importance of domain adaptation [1] BERT<sub>SC</sub> is also used.

For the BERT-based NER models, namely the ones utilizing BERT-base-cased and BERT<sub>SC</sub>, the architecture used consists of a BERT model and a linear classification layer on top of it. These models are fine-tuned end-to-end with different number of epoch and different learning rate schedulers. However, for the hyperparameters, mainly the advice given in [8] is followed. The library transformers [8] is used for implementation. Another thing to note that is, some of the sentences extracted were too long to feed into the BERT model. These sentences were split into smaller chunks as a preprocessing step, and the predictions were merged together again as a postprocessing step. These sentences were rare and covered only an [ ] % of the *Training* dataset. All the experiments were seeded for reproducibility.

For the NER model using Contextual String Embeddings, a BiLSTM layer and a CRF layer were appended on top of the embeddings. During training, only these two layers were trained and the embeddings were frozen. This setting is inline with the one described in [1]. This was implemented using the Flair library [1].

## 2.5 Hardware



## Chapter 3

# Results

### 3.1 BERT<sub>SC</sub>

### 3.2 Named Entity Recognition

To compare the developed model, a Stanford Named Entity Recognizer [13] trained on a similar training set is used as a baseline.

## Chapter 4

# Discussion

Put Limitations here.

### 4.1 BERT<sub>SC</sub>

#### 4.1.1 BERT or Flair for NER

## Chapter 5

# Conclusion

Put Future Work somewhere.

# Bibliography

- [1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018.
- [2] Krisztian Balog. *Entity-oriented search*. Springer Nature, 2018.
- [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.
- [4] Peter Bourgonje, Anna Breit, Maria Khvalchik, Victor Mireles, Julian Moreno-Schneider, Artem Revenko, and Georg Rehm. Automatic induction of named entity classes from legal text corpora. In *International Workshop on Artificial Intelligence for Legal Documents (AI4LEGAL2020)*, volume 2722, pages 1–11, November 2020.
- [5] Samuel 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, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [6] Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics.
- [7] S. Dai, Y. Ding, Z. Zhang, W. Zuo, X. Huang, and S. Zhu. Grantextractor: Accurate grant support information extraction from biomedical fulltext based on bi-lstm-crf. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(1):205–215, 2021.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [9] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10, 2014.
- [10] Jacob Eisenstein. *Natural Language Processing*. GitHub, 2018.

- [11] Borja Espejo-Garcia, Francisco J. Lopez-Pellicer, Javier Lacasta, Ramón Piedrafita Moreno, and F. Javier Zarazaga-Soria. End-to-end sequence labeling via deep learning for automatic extraction of agricultural regulations. *Computers and Electronics in Agriculture*, 162:106–111, 2019.
- [12] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Softw.*, 29(1):70–75, January 2012.
- [13] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 363–370, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
- [14] Kathleen C. Fraser, Isar Nejadgholi, Berry de Bruijn, Muqun Li, Astha LaPlante, and Khaldoun Zine El Abidine. Extracting UMLS concepts from medical text using general and domain-specific deep learning models. *CoRR*, abs/1910.01274, 2019.
- [15] Pablo Gamallo, Jose Ramon Pichel, and Iñaki Alegria. A perplexity-based method for similar languages discrimination. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 109–114, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [16] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [17] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldrige, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [18] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.
- [19] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [20] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, 2020.
- [21] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstena, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- [22] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *Similarity-Based Pattern Recognition*, pages 84–92, Cham, 2015. Springer International Publishing.

- [23] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
- [24] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*, 2019.
- [25] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics.
- [26] Subhradeep Kayal, Zubair Afzal, George Tsatsaronis, Marius Doornenbal, Sophia Katrenko, and Michelle Gregory. A framework to automatically extract funding information from text. In Giuseppe Nicosia, Panos Pardalos, Giovanni Giuffrida, Renato Umeton, and Vincenzo Sciacca, editors, *Machine Learning, Optimization, and Data Science*, pages 317–328, Cham, 2019. Springer International Publishing.
- [27] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [28] N. Kurz, F. Hamann, and A. Ulges. Neural entity linking on technical service tickets. In *2020 7th Swiss Conference on Data Science (SDS)*, pages 35–40, 2020.
- [29] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [30] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016, 2016.
- [31] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy, July 2019. Association for Computational Linguistics.
- [32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [33] Ishani Mondal, Sukannya Purkayastha, Sudeshna Sarkar, Pawan Goyal, Jitesh Pillai, Amitava Bhattacharyya, and Mahanandeeswar Gattu. Medical entity linking using triplet network. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 95–100, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.
- [34] Isaiah Onando Mulang’, Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. Evaluating the impact of knowledge graph context on entity disambiguation models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM ’20*, page 2157–2160, New York, NY, USA, 2020. Association for Computing Machinery.

- [35] Keval Nagda, Anirudh Mukherjee, Milind Shah, Pratik Mulchandani, and Lakshmi Kurup. Ascent of pre-trained state-of-the-art language models. In Hari Vasudevan, Antonis Michalas, Narendra Shekokar, and Meera Narvekar, editors, *Advanced Computing Technologies and Applications*, pages 269–280, Singapore, 2020. Springer Singapore.
- [36] Yasumasa Onoe and Greg Durrett. Fine-grained entity typing for domain independent entity linking. In *AAAI*, 2020.
- [37] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [38] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1482–1490, Online, November 2020. Association for Computational Linguistics.
- [39] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.
- [40] Jonathan Raiman and Olivier Raiman. Deeptype: multilingual entity linking by neural type system evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [41] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 498–514, Cham, 2016. Springer International Publishing.
- [42] Elliot Schumacher, Andriy Mulyar, and Mark Dredze. Clinical concept linking with contextualized neural representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8585–8592, Online, July 2020. Association for Computational Linguistics.
- [43] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
- [44] Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020. Association for Computational Linguistics.
- [45] Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1433–1439, Online, November 2020. Association for Computational Linguistics.

- [46] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.
- [47] Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM, 2020.
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [49] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020.
- [50] Qi Wang, Yangming Zhou, Tong Ruan, Daqi Gao, Yuhang Xia, and Ping He. Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition. *Journal of Biomedical Informatics*, 92:103133, 2019.
- [51] Maciej Wiatrak and Juha Iso-Sipila. Simple hierarchical multi-task neural end-to-end entity linking for biomedical text. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 12–17, Online, November 2020. Association for Computational Linguistics.
- [52] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Named entity recognition with context-aware dictionary knowledge. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 915–926, Haikou, China, October 2020. Chinese Information Processing Society of China.
- [53] Jian Wu, Pei Wang, Xin Wei, Sarah Rajtmajer, C. Lee Giles, and Christopher Griffin. Acknowledgement entity recognition in CORD-19 papers. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 10–19, Online, November 2020. Association for Computational Linguistics.
- [54] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics.
- [55] Vikas Yadav and Steven 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, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.
- [56] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online, October 2020. Association for Computational Linguistics.



- [57] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji 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)*, pages 6442–6454, Online, November 2020. Association for Computational Linguistics.
- [58] Mu Yang, Chi-Yen Chen, Yi-Hui Lee, Qian-hui Zeng, Wei-Yun Ma, Chen-Yang Shih, and Wei-Jhih Chen. Headword-oriented entity linking: A special entity linking task with dataset and baseline. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1910–1917, Marseille, France, May 2020. European Language Resources Association.
- [59] Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [60] Qingkai Zeng, Wenhao Yu, Mengxia Yu, Tianwen Jiang, Tim Weneringer, and Meng Jiang. Tri-train: Automatic pre-fine tuning between pre-training and fine-tuning for SciNER. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4778–4787, Online, November 2020. Association for Computational Linguistics.
- [61] Hongzhi Zhang, Weili Zhang, Tinglei Huang, Xiao Liang, and Kun Fu. A two-stage joint model for domain-specific entity detection and linking leveraging an unlabeled corpus. *Information*, 8(2), 2017.
- [62] Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 817–824, 2019.
- [63] Ming Zhu, Busra Celikkaya, Parminder Bhatia, and Chandan K. Reddy. Latte: Latent type modeling for biomedical entity linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9757–9764, Apr. 2020.

# Appendix A

## NER Data Preprocessing

Below, you may find the steps to assign labels to tokens using the gold annotations in sequential order.

- (i) Label tokens of ORG mentions. Sometimes, annotators tend to extract mentions not as a continuous span, but rather a list of individual words. If there more than two characters in-between, take the first continuous set of words. The decision of not taking the mention from the first annotated word until the last is based on the cases where there are too many characters or grant mentions in-between these words. It was observed that the first span mostly contained the important words to be able to identify the organization. Example annotations where underlined text corresponds to a single mention based on the gold annotation:
  - (a) National Institute of Child Health and Human Development
  - (b) the Technological Innovation and Demonstration of Social Undertakings Project fund ( HS2014003 ) of Nantong, Jiangsu, China;
- (ii) Remove duplicate ORG mentions based on their position on text. If there are two mentions with same text in different parts of the input, both are kept.
- (iii) Remove ORG mentions that are too long. Very rarely, the annotators extracted too large of a span as a mention, sometimes even the whole article. ORG mentions longer than 200 characters are discarded.
- (iv) If there are overlapping ORG mentions, keep only the one with the largest span. Example overlapping gold annotations:
  - (a) “National grant no. Science NSC Council”
  - (b) “NSC”
- (v) Label tokens of GRT mentions. Follow the same rule as the first step for mentions that are not continuous spans.
- (vi) Remove duplicate GRT mentions similar to the second step.
- (vii) Discard the grant mentions that are longer than 100 characters.
- (viii) Resolve overlapping GRT mentions similar to the fourth step.
- (ix) Resolve overlapping ORG and GRT mentions. Keep the label of the ORG mention, if there are tokens left on the right-hand-side, label them as GRT.

Text: “supported by the European Community, FP6 036097-2”

(a) ORG Mention: “European Community, FP6”

(b) GRT Mention: “FP6 036097-2”

(c) Span that is labelled as ORG: “European Community, FP6”

(d) Span that is labelled as GRT: “036097-2”

As the candidate models for NER were BERT-based [8] and Flair-based [1] models, the tokenizers these models use were tried for the tokenization of the input text before assigning the NER labels. After empirical analysis, it was decided to use the tokenizer of the “bert-base-cased” model [8], as it was splitting the text to smaller pieces, which was crucial to minimize labelling errors. One drawback of this tokenizer is that it being a word-piece tokenizer. Hence, it also splits some words into smaller pieces based on the vocabulary of the model. As a post-processing step, these wordpieces are merged back together.