Master Thesis

ON THE LIMITS OF ENTITY LINKING ON DOMAIN-SPECIFIC DATA

Quincy Liem

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Sophie Arnoult, Henk Laloli
2^{nd} reader: Lisa Beinborn

Submitted: June 30, 2023

Abstract

For the purpose of optimizing their search engine, Centraal Bureau voor de Statistiek (CBS) intended to implement a knowledge graph in the background, which would interconnect data from their various databases. My contribution in this effort involved using Entity Linking methods to map mentions of sources in semi-structured texts to the correct entry in a pre-existing database of sources. Entity Linking (EL) detects ambiguous entities in text and links them to the correct entity from a set of candidate entities, retrieved from a knowledge base (KB). Where previous endeavours in Entity Linking have focused on datasets containing highly recognizable entities with relatively little perceived ambiguity for humans, the entities found in the CBS datasets were more domain-specific and suffer from greater ambiguity across entities. The goal of this project was to translate the Entity Linking methods to the domain-specific entities of the CBS datasets. To this end, supervised and unsupervised entity linking systems were created. These systems worked in conjunction with a custom CBS-specific knowledge base. System performance proved sub-optimal for both the supervised and unsupervised Entity Linking systems. Furthermore, comparative analyses were made of system performances between the CBS datasets and a standard Entity Linking dataset (VoxEL). Both the supervised and unsupervised EL systems yielded better results on the VoxEL dataset, across all performance metrics.

Declaration of Authorship

I, Quincy Liem, declare that this thesis, titled *ON THE LIMITS OF ENTITY LINKING ON DOMAIN-SPECIFIC DATA* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 30 June 2023

Signed:

Acknowledgments

I would like to thank CBS for providing me with a pleasant internship environment. And in particular, I would like to thank Henk Laloli and Sophie Arnoult for their guidance and support. Above all, I'd like to express my gratitude towards my parents who have been my rock throughout this all.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 What is Entity Linking?

On November 7, 2020, as the Trump administration neared its conclusion, former US president Trump's lawyer held a press conference at the Four Seasons (Gabbatt, 2020). Contrary to what you may have imagined upon reading the words 'Four Seasons', the press conference was not held at the renowned Four Seasons luxury hotel venue. Instead, it was held in the parking lot of the Four Seasons Total Landscaping company, a small and rather dingy looking landscaping company in a rundown neighborhood in Philadelphia (PA), located next to a sex shop and a crematorium. Not the most obvious choice of venue for a presidential press conference.

The absurd mix-up of the incident above is a prime example of the semantic ambiguity issue that Entity Linking aims to address. Because, in human communication, the same term could refer to several different entities in the real world. Context is key in inferring the intended referent of an ambiguous term. For example, the string 'Four Seasons' could refer to a famous hotel chain, a landscaping company in Philadelphia, or a composition for violin by Vivaldi. In the context of a presidential press conference, an upscale hotel is a more obvious choice of venue than a rundown landscaping company. Still, the Four Seasons debacle proved that making inferences is not an exact science.

In essence, the task of Entity Linking (EL) aims at creating an algorithm that approximates what humans do instinctively: predicting the most probable real world referent of an ambiguous term (Figure 1.1). Li et al. (2020) define the Entity Linking task as follows: "Entity Linking (EL), the task of identifying entities and mapping them to the correct entries in a database."

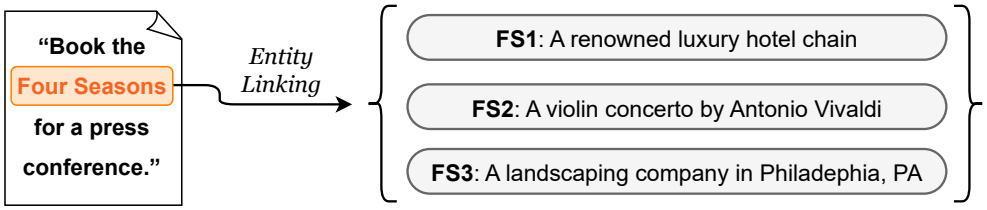

Figure 1.1: An example of the Entity Linking task

1.2 Entity Linking for CBS

Centraal Bureau voor de Statistiek (CBS) is a government institution that collects and publishes statistical information about a huge range of topics in Dutch society. Topics include birth, death, and migration rates per region, milk supply and dairy production by dairy factories, consumer price indices, consumption and production of renewable energy sources, and much more. These are all published in their online Statline database (Centraal Bureau voor de Statistiek, 2023).

Before figures are published on Statline, CBS conducts statistical research by gathering and processing datasets, which they collect from external sources: institutions, companies, municipalities, and people, among others. These external datasets, which are listed in their CBS source catalog, are used to conduct statistical research. The researchers publish short descriptions of the methods and sources they employed to conduct their research; these are called 'korte onderzoeksbeschrijvingen' (Centraal Bureau voor de Statistiek, 2020). A small fragment of a research description is presented in Figure 1.2. I elaborate on the contents of the research descriptions and the source catalog in Chapter 3.

During my internship at CBS, I was asked to help the team who were responsible for making improvements to the CBS search engine. They expressed their desire to boost the power of the CBS search engine by implementing a knowledge graph in the background. The knowledge graph was intended to inter-connect information between Statline tables, research descriptions, and the above-mentioned external datasets from the source catalog (Figure 1.2).

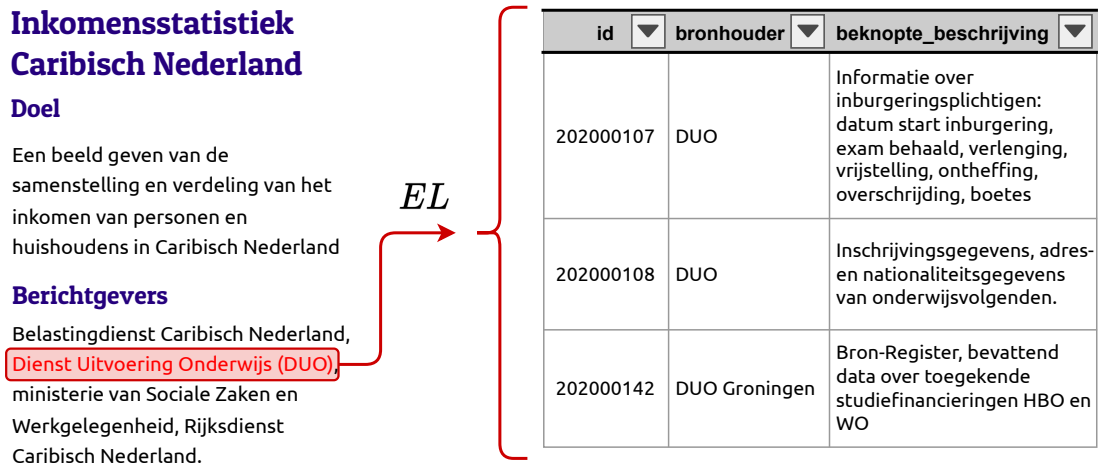

Figure 1.2: Entity Linking task for CBS sources. Left: a fragment of CBS research description. Right: a selection of source datasets from the CBS source catalog.

In particular, CBS asked me to help connect the research descriptions to the source catalog (Figure 1.2). Most research descriptions list the sources from which the researchers had collected their datasets. I was asked to link those sources to the correct

dataset from the source catalog. This might seem like a straightforward task, but it presented me with a slight complication: the research descriptions do not make any mention of the specific datasets they used in their research, only mentioning the name of the source. Furthermore, on the source catalog side, there are many sources who provided multiple, separate datasets. In other words, research descriptions ***mention*** a source, and the source catalog lists several ***entities*** (datasets from a source) for that source name, which are all potential ***candidates*** to be ***linked*** to that mentioned source name.

In Figure 1.2, I present an example of the CBS task. In the research description on the left side, "Dienst Uitvoering Onderwijs (DUO)" is mentioned as one of the researchers' sources. An excerpt of the source catalog on the right side lists several datasets, which were collected from said source. Each dataset has been given a unique identifier. The link between the source mention and the correct dataset candidate is not immediately obvious, without taking into account the context of the research description. The sources mentioned in the research descriptions are, for that reason, ambiguous. Hence, I considered Entity Linking a suitable method to address the task that CBS had given me. The aim is to indentify source entities in the research descriptions (Mention Detection) and mapping them to the correct source dataset entries in the source catalog database (Entity Disambiguation).

1.3 Outline

In Chapter 2, I explore different approaches that have been used to address Entity Linking. I present an overview of traditional Entity Linking systems, which are comprised of separate modules for Mention Detection and Entity Disambiguation, as well as commonly used EL features. I also briefly touch on newer end-to-end EL systems.

In Chapter 3, I present the data which have been used in this project.

In Chapter 4, I present the two EL systems which I used to perform Entity Linking on the CBS corpora, as well as on a standard EL dataset (VoxEL). These EL systems include a supervised EL system and an unsupervised EL system.

In Chapter 5, I present the performance results of the EL systems. I compare performance between the supervised and unsupervised models, and I compare differences in performance between the CBS corpora and the VoxEL corpus.

In Chapter 6, I delve deeper into the specific complications in translating EL methods to the CBS task. This is accompanied by Appendix B, in which I present 4 additional research descriptions as case studies.

Chapter 2

Approaches to Entity Linking

At present, there is no gold standard approach for Entity Linking. Entity Linking is still very much an active field, for which on-going efforts are being made to solve it. A whole plethora of approaches and designs have been proposed to address the EL problem. Over the past decade, Entity Linking has seen a gradual evolution from traditional approaches, which address Mention Detection and Entity Disambiguation separately — to more comprehensive end-to-end approaches in recent years, which address the before-mentioned components in a joint manner. In Chapters 2.1 and 2.2, I elaborate on the structure of traditional and end-to-end EL systems, respectively. In Chapter 2.4, I explain the basis of the EL systems which I have used for the CBS corpora.

2.1 Traditional EL systems

Traditional Entity Linking systems consist of two independent components: Mention Detection, also called Named Entity Recognition (NER), and Entity Disambiguation (Kolitsas et al., 2018). Furthermore, Piccinno and Ferragina (2014) add a 'pruning' component, which Shen et al. (2014) refer to as 'unlinkable mention prediction'. Shen et al. (2014), moreover, split the entity disambiguation component into two subcomponents: candidate entity generation, and candidate entity ranking. Aside from differences in the particular subdivision of Entity Linking components among researchers, the overall structure of a traditional EL system can be summarized as follows:

1. **Mention Detection / Named Entity Recognition (NER)**: The first step involves detecting mentions of entities in a text. This is usually accomplished independently from further EL components downstream, with a NER system.

2. **Candidate Entity Generation**: For a given entity mention detected in the previous step, a set of candidate entities is generated, to which this mention might refer to. Candidates are retrieved from a knowledge base (KB), which, in addition to candidate entities, might store information about different aliases of entities, as well as a priori popularity of entities. A knowledge base can be constructed especially for the specific EL task, or it can be a pre-existing database such as Wikidata (Vrandečić and Krötzsch, 2014).

3. **Candidate Entity Ranking**: After detecting a mention and generating a set of candidates for that mention, a link must be established between the mention and

the most probable referent from the set of candidates. This component usually involves scoring and ranking candidates. Different supervised and unsupervised ranking methods are used, e.g. independent binary classification, learning to rank, and Vector Space Models.

4. **Unlinkable Mention Prediction**: It is possible that there is no probable link between the detected entity mention and any of the generated candidates. Some Entity Linking systems, therefore, have a built-in mechanism that allows the system to predict no link between the mention and any of the candidates. This is usually referred to as 'NIL' (not in lexicon). To this end, Bunescu and Pasca (2006) use a score threshold: if the top ranked candidate score falls below a predefined threshold, the EL system will output NIL.

   Other Entity Linking systems fully omit this component. Such models make the assumption that there is always a correct link among the candidates. Rao et al. (2013), for example, simply selects "the highest ranked entry as correct, no matter its score".

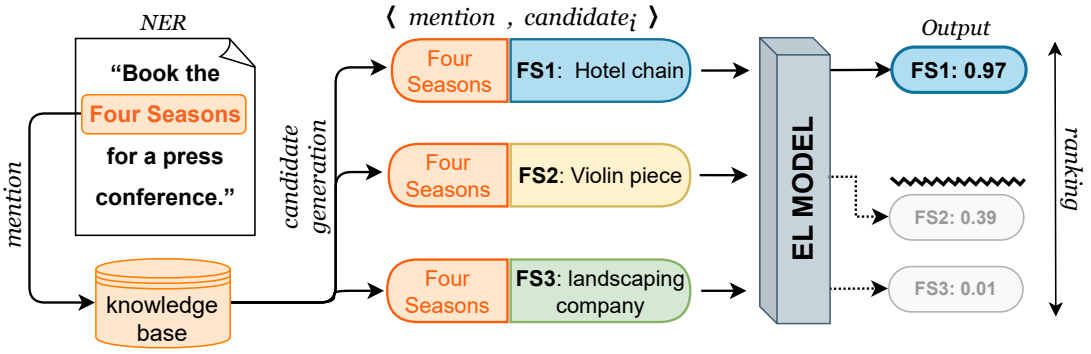

Figure 2.1: Traditional Entity Linking system

2.1.1 Ranking methods

For the ranking component of an Entity Linking system, both supervised and unsupervised methods are used, with unsupervised methods having the advantage of not requiring annotated training data. Popular Entity Linking systems have used Support Vector Machines (SVM) for ranking (Shen et al., 2014). More recently, neural architectures such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Attention-based methods have been used (Liu et al., 2019; Sun et al., 2015; Martins et al., 2019).

Binary classification involves presenting an Entity Linking system with pairs of a mention (m) and one candidate (c_i) out of the set of candidates. Each $\langle m, c_i \rangle$ pair is to be given a score denoting the link probability between the mention and the candidate. The model evaluates each pair independently of the other candidate pairs. In Rao et al. (2013) this method requires ranking all $\langle m, c_i \rangle$ pairs for a given mention, based on the link scores, and selecting the highest ranked $\langle m, c_i \rangle$ pair as the output link prediction (Figure 2.1). In a supervised binary classification Entity Linking system,

most commonly a Support Vector Machine (SVM) is trained on a binary classification of $\langle m, c_i \rangle$ pairs, where a training pair is labeled 1 to denote a link, and 0 to denote no link (Shen et al., 2014). One such EL system is used by the popular NLP module spaCy (Honnibal and Montani, 2017).

Learning to rank methods do not consider each $\langle m, c_i \rangle$ pair independently as with binary classifiers. Rather, they take into account all the relationships between all the candidates (Shen et al., 2014). Zheng et al. (2010) lists two main types of learning to rank methods: pairwise and listwise. The pairwise ranking method is trained on instances which consist of a pair of candidates with a binary classification label denoting correctly ranked or incorrectly ranked. After pairs of all interlinked candidates are scored on ranking, a final ranking of all candidates is eventually established. Instead of evaluating pairs of candidates, listwise ranking ranks the entire set of candidates in its entirety. As for the output of an EL system that utilizes a learning to rank method, the top ranking candidate is predicted to be the link, the ranking of other candidates is not relevant (Cao et al., 2007; Hofmann et al., 2013).

Unsupervised ranking A simple approach to unsupervised ranking is described by Shen et al. (2014): Vector Space Model (VSM). VSM-based approaches function are similar to the above-mentioned binary classification models. A VSM takes as input a $\langle m, c_i \rangle$ pair and generates a link score for each pair independently. The mention and a single candidate from the set of candidates are represented as two vector representations. Link scores are generated by calculating the cosine similarity between both vectors (Zeng et al., 2018). The candidate with the highest similarity score is linked to the mention.

Many Entity Linking systems use embeddings as semantic vector representations of the mention context and the candidate descriptions (Shen et al., 2021). For example, Sevgili et al. (2019) used Doc2Vec embeddings. The idea behind this, is that you compare the context surrounding an entity mention to the description of a candidate. A candidate description that is semantically very similar to the context of the mention is a more probable candidate than a candidate from a completely different domain. Likewise, as embeddings are used as semantic representations of text, it is expected that the embedding of the mention context and the embedding of a candidate description from a very similar domain, would occupy a similar vector space, as represented by a higher cosine similarity score (Thijs, 2020).

2.1.2 Features

Shen et al. (2014) describe two principal categories of Entity Linking features: context-independent, and context-dependent features. Context-independent features are constructed from characteristics of the mention itself, e.g. string similarity between the mention and the candidate entity name, named entity label, and prior probability. These features do not consider the context surrounding the context (Shen et al., 2014; Landeghem, 2019; Liu et al., 2013; Guo and Barbosa, 2014).

Conversely, context-dependent do take the mention context into account. One of the most important context-dependent features is based on textual similarity between the context surrounding the mention, and the descriptions of the candidates. As I illustrated with the Four Seasons example from Chapter 1, context provides valuable

information for candidate disambiguation. Vectorial representations of the mention context and candidate descriptions have been created in different ways, e.g. Bag-of-Words and embeddings (Shen et al., 2014; Dredze et al., 2010; Sevgili et al., 2019). Below follows a non-exhaustive list of commonly used features for Entity Linking.

- **Context-independent features:**
  - *String similarity:* How similar is the string of the mention to the string of the candidate entity name?
  - *NER label*: entity label of the mention, such as PERSON, ORGANIZATION, LOCATION, etc.
  - *Prior probability*: This feature, which is also called entity popularity, indicates how popular or obscure a candidate entity is. A candidate with a higher prior probability is, in general, more likely to be the intended entity link, than an obscure candidate with a very low prior probability. Indeed, Guo and Barbosa (2014) state that prior probability offers a strong baseline for Entity Linking.
- **Context-dependent features:**
  - *Bag-of-words:* A sparse vector which represents counts of words in the mention context. A disadvantages of the bag-of-words is that word order is not preserved.
  - *Embeddings:*
    - *Word embeddings:* Whereas bag-of-words vectors treat words as completely independent from each other. Word embeddings are dense vectors that capture more semantic information of the word. Word embeddings are trained on windows of context words surrounding the target word. The resultant dense vectors. Words occurring in similar contexts usually have similar meanings, and hence, semantically similar words have word embeddings that occupy similar vector spaces (Rudkowsky et al., 2018).
    - *Doc2Vec embeddings:* Document-level embeddings. These are dense vectors that can represent the entire mention context in a single vector.

2.2 End-to-end EL systems

In the previous section I stated that traditional Entity Linking systems treat Mention Detection and Entity Disambiguation as two separate components. Because of this, the Mention Detection component creates a performance bottleneck, as errors in the Mention Detection will have a negative impact on Entity Disambiguation further downstream. To mitigate this issue, Kolitsas et al. (2018) have proposed an end-to-end Entity Linking system, in which Mention Detection and Entity Disambiguation are addressed in a joint manner. Their EL system employs bidirectional LSTMs and pre-trained entity embeddings. Martins et al. (2019) likewise used stacked biLSTMs, but incorporated an additional attention mechanism. Broscheit (2020) and Li et al. (2020) propose, instead, a BERT-based architecture for end-to-end Entity Linking.

2.3 Evaluation metrics

In previous sections I explained that Entity Linking systems encompass two sub-tasks: Mention Detection (MD) and Entity Disambiguation (ED). The MD task is usually performed by Named Entity Recognition, and there are various methods for candidate ranking in the ED task. As an Entity Linking system involves two different subcomponents, there are also multiple different evaluation metrics which differ in how they prioritize evaluation of the sub-tasks.

Ling et al. (2015) list several evaluation metrics. The NER-style F1 metric places a heavy emphasis on the MD component of the EL system: a correct link will only count towards the final evaluation if the entity span also matches the gold entity span. Conversely, the Micro Accuracy metric prioritizes the ED component. Indeed, this metric ignores the entity spans of the MD component entirely, and only measures percentages of correctly predicted links from the ED component.

For the CBS-specific EL task, I chose to use the Micro Accuracy evaluation metric, as my focus lay with the Entity Disambiguation component of Entity Linking.

2.4 EL approach for CBS

As for the EL approach I employed for the CBS-specific EL task, I set up a supervised and an unsupervised EL system. The supervised EL system was inspired by spaCy's Entity Linker. It follows a traditional EL approach, in which mention detection and entity disambiguation are addressed separately. The Mention Detection component of the system is a separate NER model. The Entity Disambiguation component of the system was trained as a binary classifier of $\langle m, c_i \rangle$ pairs. This worked in conjunction with a custom CBS-specific knowledge base. For testing, the model assigns probability scores to each candidate in the set of candidates for a given mention, and then outputs the candidate with the highest score, provided this score exceeds a threshold. Input features include the commonly used embeddings, prior probability, and NER label. These are all described in previous sections.

The unsupervised EL system likewise evaluates independent $\langle m, c_i \rangle$ pairs. It was not trained on annotated data. The unsupervised EL system follows the Vector Space Model (VSM), described above. Furthermore, the only input features are embeddings.

Chapter 3

Data

CBS provided me with two datasets to work with: a corpus of research descriptions (*korte onderzoeksomschrijvingen*), which I describe in Chapter 3.1, and a source catalog with information about datasets used by CBS for their research (Chapter 3.2). These two datasets were were supplemented by two manually annotated datasets to train a custom NER model, and to train and evaluate my EL systems (Chapter 3.3). Additionally, the EL systems were tested on the VoxEL dataset, for performance comparison (Chapter 3.3.3).

3.1 CBS Research descriptions

CBS publishes a short research description (*korte onderzoeksomschrijving*) of each study they conduct, which is publicly available on their website (Centraal Bureau voor de Statistiek, 2020). The research descriptions provide information about the objective of the study, the type of research, method of observation, sources, etc. (Fig. 3.1) Each research description is written by the researchers involved, and while there is a general template, there is considerable variation as to what information is provided in the description, as well as how detailed the information is. This presented difficulties for Entity Linking, even if performed by a human annotator, which I elaborate on in Chapter 6.

The objective is to detect mentions of sources in the research descriptions and link them to a specific source dataset in the source catalog (Chapter 3.2). For example, the research description 'Inkomensstatistiek Caribisch Nederland' mentions 'Dienst Uitvoering Onderwijs (DUO)' as one of the sources. This source mention can be linked to multiple possible candidate datasets in the source catalog. The research descriptions list sources that are named entities, e.g. 'DUO', but also non-specific entities, e.g. 'Gemeenten' (municipalities). Especially the non-specific entities hindered Entity Linking, because it impeded both the mention detection component, as well as candidate generation from the knowledge base. More on these difficulties, I explain in Chapter 6.

To be able to work with the data, I scraped the research description html pages from the CBS website (Centraal Bureau voor de Statistiek, 2020) and converted them into a json format. Out of the 601 available research descriptions, 368 were still actively referenced by Statline tables.

Inkomensstatistiek Caribisch Nederland

Wat behelst het onderzoek

Doel

Een beeld geven van de samenstelling en verdeling van het inkomen van personen en huishoudens in Caribisch Nederland.

Doelpopulatie

De bevolking in particuliere huishoudens aan het einde van het onderzoeksjaar.

Statistische eenheid

Personen en huishoudens.

Aanvang onderzoek

2011.

Frequentie

Jaarlijks.

Publicatiestrategie

Ongeveer anderhalf jaar na afloop van elk onderzoeksjaar komen voorlopige gegevens beschikbaar. Een jaar later worden deze vervangen door definitieve gegevens.

Hoe wordt het uitgevoerd

Soort onderzoek

Het onderzoek is op basis van integrale registraties.

Waarnemingsmethode

Koppeling, integratie en bewerking van diverse registraties.

Berichtgevers

Belastingdienst Caribisch Nederland, Dienst Uitvoering Onderwijs (DUO), ministerie van Sociale Zaken en Werkgelegenheid, Rijksdienst Caribisch Nederland.

Figure 3.1: CBS research description

3.2 CBS Source catalog

CBS has curated a source catalog which comprises a list of datasets, which have been collected from many different sources over the years. Sources include government institutions, municipalities, companies, organizations, people, etc. It is with the datasets that have been gathered from these sources that CBS conducts their research projects.

The source catalog lists 336 datasets. Each dataset in the catalog has been given a unique identifier. Additional columns in the catalog list the name of source which had provided the dataset to CBS, as well as a short description of the type of information contained in the dataset. An excerpt of the source catalog is shown in Fig. 3.2.

The source catalog was used as the basis for the knowledge base component of the Entity Linking systems. Each entry in the catalog functioned as a candidate source entity (i.e. a dataset) to which source mentions from the research descriptions are to be linked.

However, in contrast to the the Four Seasons example from the Introduction, candidate entities are highly similar to each other. While the Four Seasons hotel chain and the Four Seasons Vivaldi piece are from two completely separate domains, many of the

	2021-03-15 - Export bronnencatalogus (alles)		
id	beknopte_beschrijving	bijzond	bronhouder
202000016	als open data worden gepubliceerd."	geen	Beeldmateriaal.nl
202000130	Bron-Register, bevattend gegevens over energieverbruik bedrijven en huishoudens op maandbasis	Geen	Belangrijkste energienetwerkbedrijven
202000341	Bedrijven met uitstel van belastingbetaling	geen	Belastingdienst
202000125	In de Basisregistratie Inkomen staat van ongeveer 13 miljoen burgers het verzamelinkomen of het belastbaar jaarloon. Overheidsorganisaties gebruiken de BRI om toeslagen, subsidies of uitkeringen te bepalen	burgersen	Belastingdienst
202000027	Bron-Register, bevattend alle toegewezen sofi-nummers en RSIN nummers van eenheden die een economische activiteit bedrijven	Geen	Belastingdienst
202000041	Bron-Register, bevattend alle toegewezen sofi-nummers, A-nummers Economische activiteit zelfstandigen.	burgersen	Belastingdienst
202000042	Bron-Register, bevattend toegekende aangiftebiljetcoderingen per burgerservicenummer of finummer.	burgersen	Belastingdienst

Figure 3.2: CBS source catalog

datasets in the source catalog were collected from the same sources and contain highly similar data. In the following chapters I will explore whether these domain-specific datasets can be disambiguated by my EL systems.

3.3 Annotations

CBS was not in possession of annotated datasets for either named entity recognition or Entity Linking. I decided to manually annotate a NER dataset to train a custom NER model (Table 3.1), as well as an annotated dataset for Entity Linking (Table 3.3).

3.3.1 NER annotations

The standard spaCy Dutch NER model was not suitable for the CBS-specific EL task. It failed to recognize source entities, or mislabeled them as PERSON. More importantly, the spaCy NER model was unable to recognize the non-specific entities that were not *named* entities, e.g. 'gemeenten' (municipalities). By annotating a custom NER annotation corpus, I was able to train a custom NER system that was able recognize these non-specific entities. The annotation corpus consists short texts, which were randomly extracted from the sources section of the research descriptions. Table 3.1 summarizes the NER annotation corpus. The train set consisted of 326 texts, containing 500 source entities: 321 named source entities and 179 non-specific source entities. The test set consisted of 66 texts, containing 117 source entities: 63 named source entities and 54 non-specific source entities. I specified a number of annotation guidelines, which are listed in Appendix A.

Dataset	Texts	Non-specific sources	Named sources	Total source entities
Train	326	179	321	500
Test	66	54	63	117

Table 3.1: NER annotation datasets

3.3.2 Entity Linking annotations

CBS provided me with two datasets: a corpus of short research descriptions, and a source catalog. Neither of these datasets were intended for, or created with the purpose of Entity Linking in mind. To be able to train and evaluate an EL model, a dedicated EL dataset was required. Existing EL datasets, such as the VoxEL dataset (Chapter 3.3.3) would not be optimally suited to the specific requirements of the CBS EL task. Existing EL datasets are usually set up to contain mentions of general entities, such as 'Four Seasons' or 'John Smith'. These are general entities that you would commonly find in daily life. And the candidates for such general entities are usually from highly dissimilar domains that are far removed from each other. Taking the Four Seasons example, the domain of a hotel chain is completely different from a the domain of a classical violin piece. Candidates may also vary significantly in terms of popularity or obscurity (prior probability), as I illustrated with the press conference debacle in the Introduction.

The EL task for CBS is much more domain-specific. Not only are the candidates from the source catalog much more similar to each other, there is no obvious indication that there is much variation in terms of the prior probabilities of candidate datasets. For these reasons, I deemed it unlikely that an EL model trained on an existing EL dataset containing general entities, would translate adequately to the CBS-specific EL task. It was therefore imperative that I should annotate my own Entity Linking dataset to optimally train an EL model.

Dataset	Documents	Mentions
Train	36	36
Test	18	15

Table 3.2: EL annotation datasets

A sample of the research descriptions formed the basis to set up this EL annotation dataset. Out of the 368 actively used research descriptions, a random selection was taken to be annotated (Table 3.3). The EL annotation dataset was set up such that for each research description the entity spans of source mentions were marked, as well as corresponding NER labels, and a links to unique identifiers from the source catalog. Due to annotation difficulties, which I describe in more detail in Chapter 6, I was only able set up a train and test set of 36 and 18 research description documents, respectively.

3.3.3 VoxEL dataset

The VoxEL dataset is a multilingual corpus of gold annotated Entity Linking datasets. It is based on 15 news articles from the VoxEurop news website (Rosales-Méndez et al., 2018). The below Examples A and B show schematic representations of two data items from the English language VoxEL dataset. Highlighted are the entity mentions with their gold annotated, unique Q-ID idenfitifer entity link. The Q-ID points to an entry in the Wikidata database (Vrandečić and Krötzsch, 2014).

(**A**) **François Hollande**[Q157]: "**Europe**[Q46] has no need for advice from outside for what it should do".

(**B**) Earlier on Tuesday, **Tusk**[Q946] had asked **EU**[Q458] foreign affairs ministers to propose new sanctions against **Russia**[Q159] at Thursday's meeting.

As the VoxEL dataset was constructed from a news website, most entities refer to either countries/locations or notable people. This highlights a fundamental difference between the CBS and VoxEL datasets: most mentions in the VoxEL database refer to well-known entities with extremely high prior probabilities, whereas the CBS corpus contains more obscure, domain-specific entities with less prominent prior probabilities. Prior probability is one of the most important features for Entity Linking, especially when the domain is news articles. For example, in the above Example B, 'Russia' points to entry Q159 in the Wikidata database, which is the top ranking result (i.e. highest prior probability), and refers to the country of Russia. The probability that Example B refers to a different kind of Russia other than the country, is close to zero. In other words, there is little ambiguity. Entity Linking systems can leverage this information for better performance. Conversely, prior probability is a much less significant feature in the CBS datasets: prior probabilities in the CBS knowledge base are more evenly distributed across source entities. Prior probability is, therefore, a much less potent disambiguating factor for the CBS EL system than for the VoxEL EL system, which is reflected in the performance discrepancies.

Dataset	Texts	Mentions
Train	60	141
Test	20	59

Table 3.3: VoxEL datasets

I used the VoxEL dataset to compare model performances between the CBS and VoxEL datasets (Chapter 5). If the models performed badly on both the CBS and the VoxEL datasets, then it could hint at sub-optimal model architectures. If the model performed well on the VoxEL dataset, which is standard EL dataset, but performed badly on the CBS dataset, it could indicate that the data in the CBS dataset is less optimally suited to Entity Linking.

The VoxEL dataset (Table 3.3) contained 80 English sentences which I split into 60 train and 20 test sentences. These contained 141 and 59 entity mentions respectively.

Chapter 4

Entity Linking systems

As for the CBS-specific EL task, I initially chose to use the spaCy Entity Linker in tandem with a custom, CBS-specific spaCy NER model for Mention Detection (Chapter 4.1). The spaCy model follows the traditional EL approach of separately addressing mention detection and entity disambiguation. The spaCy Entity Linker, however, lacked flexibility in regard to NIL linking, and spelling variations in the candidate generation component (Chapter 4.2). This rendered the model unsuitable for use on the CBS corpora, as they contain abundant cases of non-standard source entities which would not be linkable, as well as entities occurring in many different spelling manifestations.

I mitigated these issues by creating custom supervised and unsupervised Entity Linking systems (Chapters 4.4 and 4.5). These were trained and/or tested on the CBS EL annotation dataset and the VoxEL dataset. Furthermore, for both the CBS and the VoxEL datasets I created separate knowledge bases specific to the dataset (Chapter 4.3).

4.1 Custom NER

As the Mention Detection component in the Entity Linking pipeline, I trained a spaCy Named Entity Recognition model. This was important because the source mentions in the research descriptions often contained sources which were represented in many different forms, spellings, and compounds, which a trained NER model would not be able to recognize, e.g. 'ministeries EZ en I&M'. Furthermore, there were many cases of non-named source entities, which a standard NER model would be unable to recognize.

The model was trained on the custom CBS NER annotation set, which I described in Chapter 3.3.1. The model achieved a precision score of 0.79, a recall score of 0.78, and an F1 score of 0.78. The Entity Linking systems described in Chapters 4.4 and 4.5 were, however, trained on the custom CBS EL annotation set, which already contained gold NER annotations.

It was my intention to use this custom NER model in the eventual deployment phase. But in Chapters 5 and 6 I explain why the EL models could not be trained adequately, and thus the deployment phase was never reached. The custom NER model was consequently not used in further sections.

4.2 spaCy EL module

In her 2019 presentation, Van Landeghem described the architecture of spaCy's Entity Linking module (Honnibal and Montani, 2017; Landeghem, 2019). The model is trained as a binary classifier of $\langle mention, candidate_i \rangle$ pairs. An entity mention is sent to the preconstructed knowledge base to retrieve a set of candidate entities with corresponding descriptions. For each $\langle mention, candidate_i \rangle$ pair, the mention context is vectorized and travels through one-dimensional convolutional and pooling layers, and is then concatenated to the vector of the $candidate_i$ description, the NER label of the mention, and the prior probability of $candidate_i$. Each $\langle m, c_i \rangle$ pair is individually assigned a score that denotes the probability of the link between the mention and the candidate. The output of the model is the $\langle m, c_i \rangle$ pair with the highest link probability.

Initially I intended to use spaCy's EL model for the CBS task of Entity Linking. From my own experience in working with the model, I discovered that two crucial model features are built into the model, which rendered it unsuitable to the CBS-specific task of Entity Linking. Firstly, the spaCy knowledge base retrieves candidates by exact string matching: a mention is matched to a set of entity aliases stored inside the knowledge base, which are then used to retrieve candidates. Exact string matching would have been ill-suited to the CBS data, because of the large amount of spelling variations for the same entity on both the research description side and the source catalog side (Chapters 6.1.4 and 6.2.1). Even a single character mismatch between the mention and the knowledge base alias(es) would lead to a failure in retrieving candidates. Accounting for all possible spelling variations would, therefore, require an almost infinite number of entity aliases to be stored in the knowledge base, if it relied on exact string matching.

The next issue I encountered, was the fact that spaCy's EL model always outputs a link between a mention and one of its candidates, regardless of how low the probability for that link might be. There is no option for the model to output 'NIL'. This feature is problematic for the CBS EL task, because the source catalog is not a comprehensive list of all candidate source entities; it is still a work in progress. It should thus be highly likely that a mention has no actual link to any of the retrieved candidates (NIL).

A third, but comparatively less crucial, issue is that all components of the spaCy EL system have to be within the spaCy ecosystem. This means that the NER component has to be spaCy's own NER model — it cannot be substituted with a different model or customized to one's specific requirements.

Because of the above-mentioned issues I found the spaCy EL model to be ill-suited to the EL task for the CBS corpora. For that reason, I set out to create my own custom EL model that was inspired by the architecture of spaCy's EL model. I set out to customize this model, such that it would overcome the issues of exact string matching, and the lack of NIL linking. I explain this custom model in further detail in Chapters 4.3 and 4.4.

4.3 Knowledge bases

I created two knowledge bases for my EL systems, one for the CBS research description dataset (Chapter 3.3.2), and one for the VoxEL dataset (Chapter 3.3.3). The knowledge base (KB) functions as a look up dictionary in which information is stored about CBS source entities. From the knowledge base, three types of information can be retrieved:

(1) a list of candidate entities for a given entity mention, (2) the prior probability of an entity candidate, and (3) a short description of a candidate entity (Figure 4.1). This knowledge base is used as a component in the pipelines of both the supervised and unsupervised EL systems (Chapters 4.4 and 4.5). The two main KB functions are listed below.

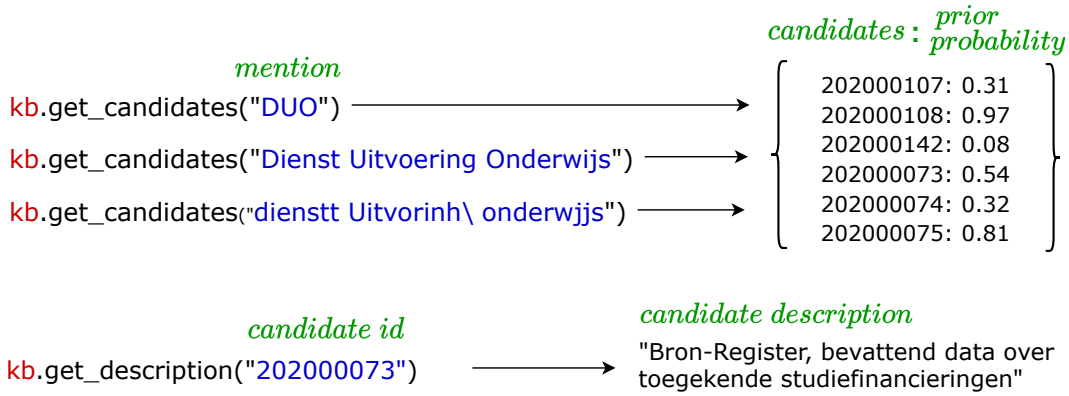

Figure 4.1: Knowledge base functions

`kb.get_candidates(mention).` This KB function takes as input an entity mention and retrieves a set of candidate source entities with their corresponding prior probabilities. It takes into account various entity aliases and it includes fuzzy string matching. Candidates are represented by their unique id.

`kb.get_description(id).` This KB function takes as input a source candidate id and retrieves the corresponding source description.

4.3.1 Aliases

In creating the knowledge base for the CBS corpus, I used the source catalog (Chapter 3.2) as a proto-KB. With the `kb.get_candidates()` function, the knowledge bases generates a set of candidate sources, given an entity mention. Matching a mention to a candidate from the KB, however, becomes more difficult when there are multiple different spelling or form variations of the same name (aliases). Many of the sources are listed in various aliases: as an abbreviation (DUO), in its full form (Dienst Uitvoering Onderwijs), or as a combination of both (Dienst Uitvoering Onderwijs (DUO)). A wide variety of aliases referring to the same entity was especially prevalent in the research descriptions. I expand on these issues in Chapter 6.1.4.

It was therefore crucial that the knowledge base was able to recognize multiple aliases. I accomplished this by using regular expressions to process source names from the source catalog into multiple entries to be added to the knowledge base. For example, the source catalog lists a source with the name “Uitvoeringsinstituut Werknemersverzekeringen (UWV)”. It was given a unique identifier (202000094). With regular expressions I disentangled the compound name and processed it into 4 aliases to be stored in the knowledge base, which I repeated for all source names in source catalog:

- Uitvoeringsinstituut Werknemersverzekeringen (UWV): 202000094

- UWV (Uitvoeringsinstituut Werknemersverzekeringen): 202000094
- Uitvoeringsinstituut Werknemersverzekeringen: 202000094
- UWV: 202000094

Each alias is linked to a unique identifier of the source entity. As the source catalog lists multiple source entities with the same name (candidates), the aliases in the knowledge base were further expanded with all the identifiers of other candidate source entities. Figure 4.1 shows the end result of this process. As "DUO" and "Dienst Uitvoering Onderwijs" are aliases of each other, both mentions should retrieve the same set of candidate source entities from the knowledge base. Apart from abbreviation aliases, the source catalog also lists source names that refer to multiple institutions. For example, a source (202000258) from the source catalog is listed under the name "ERB/KvK,Belastingdienst". This name actually refers to three separate institutions. With regular expressions I split source names into their constituent institutions, to be added individually as aliases to the knowledge base.

4.3.2 Fuzzy string matching

In order for the knowledge base to retrieve a set of source candidates for a given mention, it must first recognize the mention by matching it to one the KB stored aliases. This is described above. Still, the knowledge base would fail to match a mention to a stored alias, if the strings do not match exactly. In Chapter 6.1.4, I explain how sources are represented in many different spelling variations, in both the research descriptions and the source catalog. Therefore, relying on aliases alone is insufficient for proper KB functionality — a single character mismatch between mention and a KB alias would result in 0 retrieved candidates.

To solve this issue, I added a fuzzy string matching component to the knowledge base: the Damerau-Levenshtein algorithm (Fairchild, 2013; Chaabi and Allah, 2022). The algorithm takes as inputs two strings, and calculates the character-based string similarity score [0,1] between those strings. It takes into account character deletions, insertions, substitutions, and transpositions. When the `kb.get_candidates(mention)` function is called, fuzzy string similarity scores are calculated between the mention and all aliases stored in the knowledge base. Both the mention and the alias are decapitalized prior to similarity scoring. Only those aliases with scores above the user defined threshold (0.75) are matched to the mention. The output is the union of the sets of candidates of all matched aliases.

4.3.3 Prior probability

$$P_{prior}(e_i) = \frac{count(e_i) + 1}{\sum_{e_j \in E_c} count(e_j)} \tag{4.1}$$

Prior probability is an important feature for Entity Linking. It is a measure which indicates how frequent the entity occurs comparatively to other candidate entities. In the case of the Four Seasons, the renowned hotel chain is *a priori* a more probable reference than some obscure landscaping company. I calculated the prior probabilities for every entity by counting the absolute frequency of mentions in the CBS annotation set (e_i), and divided it by the sum of the frequencies of other candidate entities (e_j) from the set of candidate entities (E_c). The formula for calculating the prior probability

was adapted from Shen et al. (2014) and Liu et al. (2013). Furthermore, I incorporated add-one (or Laplace) smoothing to account for entities with a mention frequency of 0, as suggested by Blanco et al. (2015).

4.3.4 VoxEL knowledge base

A similar approach to the knowledge base creation procedure as described above was applied to creating a knowledge base for the VoxEL dataset. Where I used the source catalog as a proto-KB for the CBS knowledge base, I used online wikidata.org database (Vrandečić and Krötzsch, 2014) to construct the VoxEL knowledge base.

Entity mentions from the VoxEL database were used to send a search query to the wikidata.org website, which retrieves a list of up to 20 search results. Each result has its own unique Q-ID identifier. I scraped the necessary information from the search result with the Q-ID that matched the Q-ID of the mention. This included a short description of the entity, various aliases and number of sitelinks. The number of site links were used as a frequency for calculating the prior probability. If any of the other 19 search results were an exact string match with the mention, or an exact string match with one of the aliases, then these results were also scraped. The Q-IDs, entity descriptions, aliases, and prior probabilities were then added to the VoxEL knowledge base.

4.4 Supervised EL system

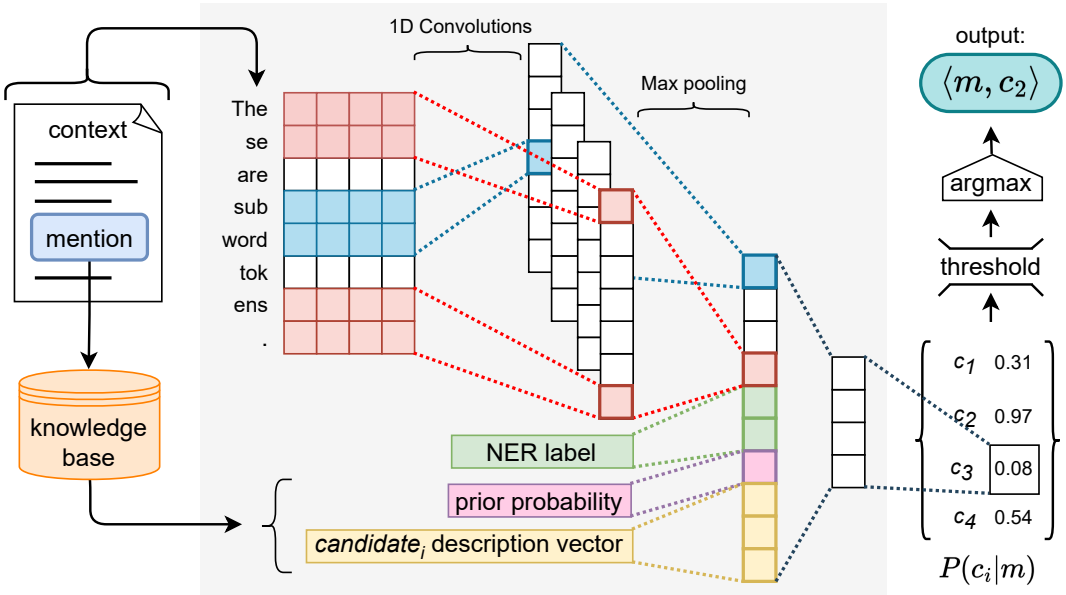

Figure 4.2: Supervised Entity Linking model.

As explained in Chapter 4.2, the spaCy EL model lacks two crucial features: fuzzy string matching and NIL linking. The lack of these features rendered it unsuitable to the CBS-specific EL task. I therefore created a custom model whose architecture is based on spaCy's EL model, but incorporates the above-mentioned features. Figure 4.2 presents a schematic overview of the model architecture, which can be summarized as follows:

1. **Mention Detection / NER**. The first element of the pipeline involves detecting mentions of source entities inside the body of text of a research description through named entity recognition (NER), and classifying these entity mentions with a NER label.

2. **Generating candidates**. The mention is matched to a set of candidate entities from the knowledge base, taking into account spelling variations and aliases. Each candidate contains a short entity description and a prior probability. Each $\langle mention, candidate_i \rangle$ pair is fed to the model independently as a single training instance.

3. **Text vectorization**. The context surrounding the entity mention is split into a maximum of 512 subword tokens and converted into BERT embeddings (728 dimensions). The context is thus represented by a matrix in the shape of (512, 728). A candidate description is likewise converted into BERT embeddings. However, instead of a stack of n embeddings for n tokens, the entire candidate description is instead represented by a single overall sentence embedding. Choi et al. (2021) explain that taking the embedding from the `[CLS]` token can function as a reasonable sentence-level context representation. The difference in vector representation between the context and the candidate description is in line with spaCy's EL architecture.

4. **Feature concatenation**. The stack of context embedding vectors pass through one-dimensional convolutional layers (filter size 3), pooling layers (size 2), and dropout layers ($p = 0.25$). The resultant vector is then flattened and concatenated to 3 additional input features: the NER label of the mention, the description embedding vector of $candidate_i$, and the prior probability of $candidate_i$.

5. **Candidate scoring**. The concatenated vector from the previous step goes through a fully connected layer ($n = 100$) to generate a probability score: $P(c_i|m) = [0, 1]$. This probability denotes how likely it is for there to be a link between between the mention and $candidate_i$.

6. **Candidate ranking**. The algorithm outputs the $\langle m, c_i \rangle$ pair with the highest probability score, provided that the score exceeds a threshold of 0.5. If this threshold is not met, the output is 'NIL', which would indicate no link between the mention and any of the candidates. The threshold is a crucial component of the model architecture - without it, the model would always output a link between a mention and a candidate, even if the candidate score is extremely low.

My custom CBS-specific NER model was intended to be used as the Mention Detection component in the EL pipeline. But as I explain in Chapters 3.3.2 and 6, the EL task proved unfeasible for the available data, so I did not apply to the custom NER model to the entire dataset of research descriptions. I, instead, only used gold annotations.

The EL architecture further down the pipeline, in essence, was trained as a binary classifier, where each $\langle mention, candidate_i \rangle$ pair was classified independently. Hence, a research description with a single entity mention, which retrieved five candidates from the knowledge base, would yield five $\langle m, c_i \rangle$ pairs to be used as individual training instances. A training instance consisted of (a) a BERT vector representation of the

context surrounding a mention, (b) a BERT embedding of the description of one of the retrieved candidate entities from the knowledge base, (c) the prior probability of the candidate as a float, (d) the NER label of the mention (one-hot), and (e) a binary classification indicating whether a link between the mention and the candidate exists (1), or does not exist (0).

The supervised EL model was trained on 36 annotated research descriptions from my gold EL annotation dataset (Chapter 3.3.2), which were preprocessed into 157 training instances. Since the model was trained on newly created dataset, it was necessary to train the model on a benchmark dataset Comparing the performance of the model on the CBS dataset to its performance on a benchmark dataset, it ensured that any discrepancies between performance are a result of the quality of the datasets that the models were trained on, and not of the architecture. Thus, a separate model with identical architecture was trained on the VoxEL EL dataset (Chapter 3.3.3) along with the VoxEL-specific knowledge base (Chapter 4.3). This model was trained on 60 VoxEL sentences, which were preprocessed into 474 training instances.

4.5 Unsupervised EL system

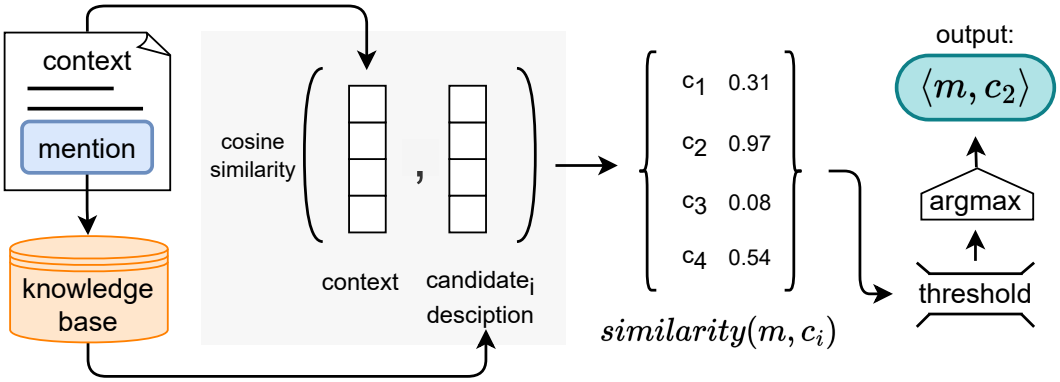

Figure 4.3: Unsupervised Entity Linking model.

In addition to my supervised model, I created a simpler unsupervised EL model, which was based on the Vector Space Model approach described in Chapter 2.1.1. A major advantage of an unsupervised model is that does not require any labeled training data. This is an especially important advantage considering the difficulties with annotations (Chapter 6) and the small training set as a consequence of that.

The simpler unsupervised model requires fewer types of input data than the supervised model: only embedding representations of the mention context and that of a candidate description. It lacks other input features that would require annotated training data, e.g. prior probability. Furthermore, the model forgoes the CNN architecture of the supervised model. Instead, it employs a simpler cosine similarity measurement between embeddings.

Apart from simpler architecture and input features, it works fundamentally the same as the supervised model. Because, it evaluates independent $\langle mention, candidate_i \rangle$ pairs and estimates the probability of there being a link between the mention and a given candidate between 0 and 1. The algorithm involves 5 steps, which is schematically

shown in Figure 4.3:

1. **Mention Detection / NER**. Detecting mentions of source entities inside the body of text of a research description by means of named entity recognition (NER).

2. **Generating candidates**. A list of candidate source entities is retrieved from the knowledge base, taking into account spelling variations and aliases. Each candidate contains a short description. Each $\langle mention, candidate_i \rangle$ pair is fed to the model independently.

3. **Text vectorization**. The context surrounding the mention of a source entity is word tokenized, and stop words are removed. The remaining tokens are converted into a Doc2Vec embedding. The same vectorization procedure is applied to each candidate description. Doc2Vec embeddings of dimension size 300 were trained on Dutch and English corpora from the Leipzig Corpora Collection (Leipzig Corpora Collection, 2020, 2012), which were supplemented with texts from the training datasets to minimize OOV (out of vocabulary) issues.

4. **Candidate scoring**. The link probability for $\langle mention, candidate_i \rangle$ pair is scored between 0 and 1 by calculating the cosine similarity between the context vector and the candidate description vector. A higher cosine similarity would indicate a higher link probability.

5. **Candidate ranking**. The algorithm outputs the $\langle mention, candidate_i \rangle$ pair with the highest probability score, provided that the score exceeds a threshold of 0.5. If this threshold is not met, the output is 'NIL'.

A separate unsupervised model with the same architecture was created for the VoxEL dataset with the VoxEL knowledge base.

Chapter 5

Results

Both the supervised and unsupervised models were tested on the CBS Entity Linking test set, which contained 15 entity mentions. The models were subsequently tested on the VoxEL Entity Linking dataset, which contained 59 entity mentions.

corpus	EL system	precision	recall	F1	mentions
CBS	supervised	0.33	0.27	0.29	15
	unsupervised	0.30	0.27	0.27	15
VoxEL	supervised	0.98	0.98	0.98	59
	unsupervised	0.53	0.29	0.35	59

Table 5.1: Performance of EL systems

In Chapter 2.3, I described several evaluation metrics for Entity Linking. I chose the Micro-Accuracy evaluation metric, which simply measures percentages of correctly linked mentions (Ling et al., 2015). I chose this metric, because other EL evaluation metrics such as NER-style F1, penalize performance results if the system entity span does not match the gold entity span. Absolute correctness of entity spans was not of paramount importance to the CBS EL systems, therefore I did not incorporate this information in the evaluation. Performance results are summarized in Table 5. The supervised model performed significantly better on the VoxEL corpus than on the CBS corpus on all performance metrics. The unsupervised model performed marginally better on all metrics on the VoxEL dataset, compared to the CBS dataset. Across datasets, both supervised and unsupervised models performed better on the VoxEL dataset than on the CBS dataset.

Performance across models. First, let us inspect performances across models. Predictably the much simpler unsupervised model performed worse on both datasets. The performance gap was, however, especially notable on the VoxEL dataset. Inspecting the architectural differences between the two models might offer an explanation. The rudimentary architecture of the unsupervised model only takes into account the context of the mention and compares it to a candidate description, whereas the supervised model

also takes into account other features, such as prior probability. I suspect that context alone is not a sufficiently informative feature for candidate disambiguation, and that prior probability plays a much more significant role. Indeed, a deeper inspection of the VoxEL dataset revealed that the link between mention and gold annotated candidate entity is in 98% of cases the candidate with the highest prior probability. In theory, the model would thus be able to achieve a precision score of 98% from one input feature alone. With prior probability being such a dominant feature, it is uncertain whether the model was able to extract features from the mention context and the candidate descriptions at all. A different dataset, that is less heavily reliant on prior probability, might have given us more insights into context-focused model performances.

Performance across datasets. What could explain the wide discrepancy in performance between both datasets? The model architectures remained identical across datasets, so it is unlikely that the discrepancy can be attributed to technical flaws in the algorithms. The more probable explanation is that information in the CBS annotation dataset was not conducive to the task of Entity Linking. This would explain why both models performed worse on the CBS dataset. The limited scope of information enclosed in the CBS dataset effectively created a performance ceiling, above which the models, however advanced their architecture might be, would not be able to reach. In Chapter 6 I delve deeper into the CBS data to explore whether the sub-optimal performance can be attributed to the quality of the training data.

Chapter 6

Evaluation of the task

In the previous Chapter I explained that a potential cause for sub-optimal system performance could be attributed to the quality of the training data. While it is tempting to try to improve a system by testing different algorithms, adding more modules to the pipeline, and fine-tuning the parameters, we risk losing track of a more global overview on what we are trying to achieve with the task. During the annotation process, I discovered that the task of Entity Linking on the CBS corpus is inherently not feasible for human annotators.

There are two main reasons for this. The candidate descriptions in the source catalog were from highly similar domains. This is in contrast to the example of the Four Seasons, where a global hotel chain and a piece of classical music are from two completely different domains. Having to choose from a set of candidates that are all very similar, makes disambiguation exceedingly difficult. What's more, candidate descriptions did not contain sufficient information to make the correct link obvious. It would require an annotator to have an in-depth familiarity with the research, or have been personally involved in conducting the research, in order to be able to choose the correct candidate.

I should reiterate and stress that the CBS datasets were not initially created with the task of Entity Linking in mind. They were intended for internal use. So it would be unreasonable to expect a high level of compatibility with the task at hand. In the next sections I will elaborate on the above-mentioned issues. In Appendix B, I present 4 research descriptions as case studies, to further illustrate what issues arise when trying to link mentions from a research description to candidates in the source catalog.

6.1 Research descriptions

6.1.1 Do the sources still exist?

Not all studies conducted by CBS are still actively in use by CBS. CBS publishes *Statline* tables which show information about a certain statistics in Dutch society. Each Statline table is based on one or more CBS studies, and the table lists links to the research descriptions of those studies. The 4919 Statline tables only link to 368 of the 601 available research descriptions. The research descriptions without links are often obsolete (even from the 1980s), and are not actively used by CBS anymore.

The issue that arises here, is that obsolete research descriptions often contain obsolete sources that either don't exist anymore or whose names have changed. This

issue is not only present in the obsolete research descriptions, but also in the still active research descriptions. For example, the research description WONINGBESTAND VAN WONINGCORPORATIES: T/M 2001 mentions the *Ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieu (VROM)* as its source. This institution has not existed since 2010[1], and there exists no entry in the source catalog of this institution. It is impossible to link textual source mention to an identifier from the source catalog, if it doesn't exist in the source catalog.

6.1.2 Where to find the sources in the text?

Although a template was provided to the authors of the research descriptions, there is considerable variety in terms of which specific bits of information the authors provide, and how detailed that information is. Across the research descriptions, I found 207 different headers. The sources are usually listed under the header 'Berichtgevers', but they could also be listed under 'Belangrijkse bronnen' (KWARTAALREKENINGEN), 'Waarnemingseenheid' (STATISTIEK STIJGERS EN DALERS; OMZETONTWIKKELING), etc. Some research descriptions have no headers at all (BETROKKENHEID BURGER).

The formatting variations make it difficult to determine where precisely to look for source mentions. If there is no standard header, the alternative could be scanning the entire text with NER. But this strategy is not optimal, as the NER system will recognize all mentions of organizations and institutions in the entire text, even if these organizations were not actual sources.

6.1.3 What specific data was obtained from each source?

Most of the research descriptions mention only the names of the sources from which the researchers obtained their data, but lack information about the specific type of data that was obtained from each source. Without this information it is exceedingly difficult to link a source mention to a candidate in the source catalog. For example, *Dienst Uitvoering Onderwijs (DUO)* is listed as a source in the research description INKOMENSSTATISTIEK CARIBISCH NEDERLAND. In order to link it to one of the 7 candidate id's in the source catalog, there has to be sufficient information in the research description about what kind of data from DUO was used. That information is lacking, making it very difficult for a human annotator to assign the correct id code. Moreover, it is possible that there is no link to any of the id's in the source catalog (NIL). Another possibility is that data from multiple datasets were combined, which would mean that one source mention should be linked to multiple id's. There is no way of finding out which of these possibilities is true, without consulting the authors of all the research descriptions.

6.1.4 Acronyms and spelling variations

As with the formatting of the research descriptions themselves, the researchers who wrote them were given complete freedom as to how to spell the sources. Lack of standardization makes it difficult to link a source mention to candidates in the source catalog, if both strings are not matched. I made progress in solving the issue for the most part by adding multiple aliases of the same source entity to the knowledge base,

[1] at the time of writing

as well as implementing fuzzy string matching. Listed below are examples of spelling variations that I encountered.

- Name only:
  - Dienst Uitvoering Onderwijs
  - ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieubeheer
  - Uitvoeringsinstituut Werknemersverzekeringen
- Acronym only:
  - DUO, UWV, DNB, VROM
- Name and acronym between brackets:
  - Dienst Uitvoering Onderwijs (DUO)
  - Uitvoeringsinstituut Werknemersverzekeringen (UWV)
  - Ministerie van Volkshuisvesting, Ruimtelijke ordening en Milieu (VROM)
- Acronym and name between brackets:
  - UWV (Uitvoeringsinstituut Werknemersverzekeringen)
  - DUO (Dienst Uitvoering Onderwijs)
- Partial acronym:
  - Ministerie van VROM
  - Ministerie van EL&I
- Combined sources:
  - Ministeries van OCW en EL&I
- Capitalization variations
  - ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieubeheer
  - uitvoeringsinstituut werknemersverzekeringen
  - Uitvoeringsinstituut Werknemersverzekeringen

6.1.5 Non-specific sources

Many research descriptions list non-specific sources that do not refer to an unambiguous real world entity. The research description LOGIESACCOMMODATIES; BOEKINGSWIJZE, SAMENWERKING, CONCURRENTIE lists as a source *Nederlandse logiesaccommodaties* (Dutch overnight accommodations). First of all, it is difficult for a NER system to recognize such text spans as a source. And secondly, because there is no mention of a specific name, it is difficult to generate candidates from the source catalog or knowledge base.

6.2 Source catalog

The source catalog lists datasets used by researchers in CBS studies. Each entry has a unique id and provides information about the source owner that provided the dataset and a short description of what type of data is in the dataset. The creators of the source catalog asked the CBS researchers to fill in a form about their datasets. The source catalog is the direct result of that. No further filtering or standardization methods were applied, nor was checked whether multiple entries might be the same or overlap. The source catalog was created a few years ago and only includes datasets from recent studies. Hence, many of the sources mentioned in the research descriptions do not occur in the source catalog.

6.2.1 Source owners

As was the case in the research descriptions, the source owners (those who provided the datasets) in the source catalog display a great variety in spelling and representation. This is due to the fact that no standardization or filtering methods were applied during the creation of the source catalog. Because there are spelling differences on both the side of the research descriptions as well as on the side of the source catalog, matching one to the other has become an exceedingly difficult task. What is more, the source catalog contains entries that refer to non-specific sources, e.g. 202000144: Bijna alle ministeries (almost all ministries). Below is a small selection of examples from the source catalog.

- Source owner column contains multiple source entities:
  - 202000118: Kadaster, Landbouw en Innovatie, Rijkswaterstaat, Defensie en ProRailgemeenten, provincies, waterschappen
  - 202000008: CITO (Centraal Instituut voor Toetsontwikkeling) (Centraal Instituut voor Toetsontwikkeling)/ Rijksoverheid
  - 202000258: ERB/KvK,Belastingdienst
  - 202000323: banken/ pensioenuitvoerder
  - 202000268: CTI, Google, etc.
  - 202000120: Gemeenten, provincies, waterschappen en de ministeries EZ en I&M (RWS)
- Source owner column is empty: 202000333, 202000290, 202000334, 202000254, 202000255, 202000260, 202000264, 202000265, 202000274, 202000297, 202000329, 202000231, 202000202.
- Source owner is represented in different spellings:
  - 202000198: CAK
  - 202000169: het CAK
  - 202000120: Gemeenten, provincies, waterschappen en de ministeries EZ en I&M (RWS)
  - 202000024: Dienst Verkeer en Scheepvaart (DVS), Ministerie Infrastructuur en Milieu
  - 202000339: Vektis

  - 202000220: Vektis (informatiecentrum zorgverzekeraars)
  - 202000340: UWV
  - 202000094: Uitvoeringsinstituut Werknemersverzekeringen (UWV)
  - 202000087: Uitvoeringsinstituut voor Werknemersverzekeringen (UWV) WERKbedrijf (v/h CWI)

- Non-specific source owners:
  - 202000144: Bijna alle ministeries
  - 202000183: Containerterminals, waaronder ECT, en regionale overslagcentra
  - 202000130: belangrijkste energienetwerkbedrijven
  - 202000031: het bedrijf zelf

6.2.2 Short descriptions

Each dataset in the source catalog has a short description (korte beschrijving). The short descriptions are short. Domain-specific knowledge is required to have a clear understanding of what type of data is in each dataset listed in the source catalog. Especially for a source mention which generates multiple candidates in the source catalog, it is difficult for a human to judge which one of the candidate datasets should be linked, based on the short descriptions. Some of the short descriptions are very highly similar indeed, as exemplified below.

- 202000045: Bron, bevattend detailgegevens administratieve eenheden.
- 202000046: Bron, bevattend detailgegevens overgang administratieve eenheden.
- 202000057: Bron, bevattend data over ontvangen kinderopvangtoeslagen.
- 202000058: Bron, bevattend data over ontvangen kindertoeslagen.

Chapter 7

Conclusion

As part of optimizing the CBS search engine, I have sought to connect mentions of source entities in the CBS research descriptions, to sources in the CBS source catalog. I have attempted to address this problem with Entity Linking. To this end, I created a supervised and an unsupervised Entity Linking system, as well as a domain-specific knowledge base. The EL systems were trained and tested on data provided by CBS, as well as on pre-existing EL annotations (VoxEL).

The scope of this project differed from previous endeavours at addressing the Entity Linking problem. Whereas EL systems in the past have focused primarily on datasets containing highly recognizable entities with little ambiguity issues, which human annotators can easily disambiguate and link (e.g. 'Russia'), the CBS-specific EL task involved data which was highly domain-specific and more fine-grained. It proved inherently much more challenging for human annotators due to greater ambiguity. Indeed, this is reflected by sub-optimal system performance on the CBS datasets across EL systems. When applied to the standard EL dataset VoxEL, the EL systems showed more favorable results than when applied to the CBS datasets.

In Chapter 6 I explained why the task of Entity Linking is unfeasible for the purpose of linking source mentions in the research descriptions to datasets in the source catalog. The data do not provide sufficient information for human annotators to perform to the disambiguation task. Without this ground truth, any machine learning model cannot accurately learn the task either. A large corpus of high quality data is the key to a well performing Entity Linking system.

For future endeavours, it is crucial that the information in the research descriptions and the source catalog be informative enough such that a human is able to perform the task. It would require restructuring of the CBS data documentation. Firstly, the research descriptions should be standardized such that they all present the same information in a standardized way, as well as standardization of source spellings. Secondly, explicit information is needed about what specific data the researchers used from each source.

The same two principles apply to the source catalog. Filtering and standardization should make sure that there are no overlapping datasets, and that all the sources are spelled in a standardized way. The short descriptions that accompany each dataset should be expanded such that no domain-specific knowledge is required to know what data is present in each dataset.

Appendix A

NER annotation guidelines

- Annotate the maximal span of a named entity source. For example, if a source mention is followed by its acronym between brackets, include the acronym between brackets in the entity span. Assign the label SOURCE.
- If a source is not a *named* entity, e.g. 'rechtbanken', assign the label NONSPECIFIC_SOURCE.
- Do not include articles (de/het/een) in the entity span, unless it is an inseparable part of entity name, e.g. 'De Nederlandsche Bank'.
- Include only the core NP of a non-specific source. Exclude adjectives and relative clauses. For example, the research description 'Landbouwtelling' lists as its source 'Agrarische bedrijven met een economische omvang van 3000 SO of meer' (Agrarian businesses with an economic size of 3000 SO or more). I only included 'bedrijven' in the entity span, so as to increase recall.

Appendix B

Case studies

In this section I examine several examples from the research descriptions. I explain what issues arise in Entity Linking. Because the research descriptions can be quite long, I only show the relevant paragraphs. Named sources are highlighted in green, non-specific sources are highlighted in red.

B.1 Case: Faillissementen

Text in research description:

> ***Doel***
> Een statistische beschrijving van het aantal door rechtbanken in Nederland uitgesproken faillissementen.
>
> ***Berichtgevers***
> De Nederlandse Rechtbanken fungeren als dataleverancier. Elk door een rechtbank in Nederland uitgesproken faillissement wordt openbaar gemaakt. Een overzicht van de uitgesproken faillissementen wordt dagelijks elektronisch ter beschikking gesteld aan het CBS en andere geïnteresseerde partijen.

Candidates from the source catalog:

id	bronhouder	beknopte beschrijving
202000164	Openbaar Ministerie (OM), Rechtbanken/ICTRO	Bron-Register, bevattend gegevens over strafzaken bij het Openbaar Ministerie
202000038	Rechtbank	Bron-Register, bevattend gegevens over echtscheidingprocedures
202000192	Rechtbank	Bron-Register, bevattend gegevens over bestuursrechtzaken.

Issues: The research description FAILLISSEMENTEN mentions *Nederlandse rechtbanken* (Dutch courts) as its source. Firstly, this exact string does not occur in the source catalog. The knowledge base with its fuzzy matching component was, however, able to retrieve 3 candidates from the source catalog (Table B.1). None of the candidates'

descriptions mention anything about bankruptcies (faillissementen). Based on the contextual information in the research description and the candidates' short descriptions, it is impossible to assign a link to the mention *Nederlandse rechtbanken*.

B.2 Case: Dividend beursgenoteerde fondsen

Text in research description:

Doel
Weergave van de waarde van het uitgekeerde dividend onderverdeeld naar sector.

Waarnemingsmethode
Elektronische aanlevering van beursgegevens door Euronext Amsterdam.

Berichtgevers
Euronext Amsterdam.

Candidates from the source catalog: None.

Issues: The source mention *Euronext Amsterdam* does not occur in the source catalog and therefore it cannot be linked.

B.3 Case: Banen en lonen, zeggenschap van bedrijven; SBI'93

Text in research description:

> ***Doel***
> Het publiceren van gegevens over de banen en lonen van werknemers bij bedrijven in Nederland, onderverdeeld naar buitenlandse en Nederlandse zeggenschap.
>
> ***Waarnemingsmethode***
> Een combinatie van bedrijfs- en persoonsenquêtes en registraties.
>
> ***Berichtgevers***
> Bedrijven, instellingen en personen.

Candidates from the source catalog: All or none or some.

Issues: The research description mentions *bedrijven* (companies), *instellingen* (institutions) and *personen* (people) as their sources. These are non-specific sources that are not *named* entities. All datasets in the source catalog were obtained from companies or instutitions, so in theory all the entries in the source catalog could be candidates. It is also possible that there is no actual link with any of the entries, or information from multiple datasets could have been used. 'personen' does not occur in the source catalog, so it cannot be linked.

B.4 Case: Inkomensstatistiek Caribisch Nederland

Text in research description:

Doel
Een beeld geven van de samenstelling en verdeling van het inkomen van personen en huishoudens in Caribisch Nederland.

Waarnemingsmethode
Koppeling, integratie en bewerking van diverse registraties.

Berichtgevers
Belastingdienst Caribisch Nederland, **Dienst Uitvoering Onderwijs (DUO)**, **ministerie van Sociale Zaken en Werkgelegenheid**, **Rijksdienst Caribisch Nederland**.

Candidates from the source catalog: *Belastingdienst Caribisch Nederland* does not occur exactly in the source catalog. 'Belastingdienst' does occur and has 50 candidate datasets. *ministerie van Sociale Zaken en Werkgelegenheid*: Does not occur in source catalog. *Rijksdienst Caribisch Nederland*: Does not occur in source catalog. *Dienst Uitvoering Onderwijs (DUO)* does not occur in the source catalog. 'DUO' does occur as the source owner of 7 datasets:

id	bronhouder	beknopte beschrijving
202000107	DUO	Informatie over inburgeringsplichtigen: datum start inburgering, examen behaald, verlenging, vrijstelling, ontheffing, overschrijding, boetes
202000108	DUO	inschrijvingsgegevens, adres- en nationaliteitsgegevens van onderwijsvolgenden.
202000142	DUO Groningen	Bron-Register, bevattend examengegevens van leerlingen in het voortgezet onderwijs (vo), zoals de examenuitslag, het opleidingsnummer, de code van de onderwijsinstelling en de onderwijssoort van het examen.
202000073	DUO Groningen	Bron-Register, bevattend data over toegekende studiefinancieringen HBO en WO
202000074	DUO Groningen	Bron-Register, bevattend toegekende studietoelagen MBO jonger dan 18 jaar.
202000075	DUO Groningen	Bron-Register, bevattend toegekende studietoelagen MBO ouder dan 18 jaar.
202000004	Ministerie van Onderwijs, Cultuur en Wetenschap/DUO Groningen	Electronische jaarverslagen van schoolbesturen

Issues: *Belastingdienst*: the research descriptions mentions that the researchers used and processed data from multiple registrations. The context does not explain which registrations were used. Without domain-specific knowledge of Belastingdienst registrations, it is impossible to assign a link to a dataset in the source catalog.

Dienst Uitvoering Onderwijs (DUO): The context does not provide enough information to assign a link to a dataset in the source catalog.

ministerie van Sociale Zaken en Werkgelegenheid and *Rijksdienst Caribisch Nederland* do not occur in the source catalog, so they cannot be linked.

Bibliography

R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 179–188, 2015.

S. Broscheit. Investigating entity knowledge in bert with simple neural end-to-end entity linking. *arXiv preprint arXiv:2003.05473*, 2020.

R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. 2006.

Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

Centraal Bureau voor de Statistiek. Onderzoeksomschrijvingen, Apr 2020. URL `https://www.cbs.nl/nl-nl/onze-diensten/methoden/onderzoeksomschrijvingen`.

Centraal Bureau voor de Statistiek. Statline, May 2023. URL `https://opendata.cbs.nl/#/CBS/en/`.

Y. Chaabi and F. A. Allah. Amazigh spell checker using damerau-levenshtein algorithm and n-gram. *Journal of King Saud University-Computer and Information Sciences*, 34(8):6116–6124, 2022.

H. Choi, J. Kim, S. Joe, and Y. Gwon. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 5482–5487. IEEE, 2021.

M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin, et al. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

G. Fairchild. pyxdameraulevenshtein implements the damerau-levenshtein (dl) edit distance algorithm for python in cython for high performance., 2013. URL `https://github.com/gfairchild/pyxDamerauLevenshtein`.

A. Gabbatt. Tourists flock to Four Seasons Total Landscaping after Giuliani debacle. 12 2020. URL `https://www.theguardian.com/us-news/2020/dec/02/four-seasons-total-landscaping-philadelphia-tourism-selfies`.

Z. Guo and D. Barbosa. Entity linking with a unified semantic representation. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1305–1310, 2014.

K. Hofmann, S. Whiteson, and M. de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16:63–90, 2013.

M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

N. Kolitsas, O.-E. Ganea, and T. Hofmann. End-to-end neural entity linking. *arXiv preprint arXiv:1808.07699*, 2018.

S. V. Landeghem. Sofie Van Landeghem: Entity linking functionality in spaCy (spaCy IRL 2019), 7 2019. URL `https://drive.google.com/file/d/1EuGxcQLcXvjjkZ-KRUlwpr_doBVyEBEG/view`.

Leipzig Corpora Collection. Leipzig corpora collection (2020): Dutch corpus based on mixed media material from 2020., 5 2012. URL `https://svn.apache.org/repos/bigdata/opennlp/trunk/leipzig/data/nld_mixed_2012_1M-sentences.txt`.

Leipzig Corpora Collection. Leipzig corpora collection (2020): English newspaper corpus based on material from 2020., 5 2020. URL `https://wortschatz.uni-leipzig.de/en/download/English`.

B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W.-t. Yih. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413*, 2020.

X. Ling, S. Singh, and D. S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.

C. Liu, F. Li, X. Sun, and H. Han. Attention-based joint entity linking with entity embedding. *Information*, 10(2):46, 2019.

X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311, 2013.

P. H. Martins, Z. Marinho, and A. F. Martins. Joint learning of named entity recognition and entity linking. *arXiv preprint arXiv:1907.08243*, 2019.

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.

D. Rao, P. McNamee, and M. Dredze. Entity linking: Finding extracted entities in a knowledge base. *Multi-source, multilingual information extraction and summarization*, pages 93–115, 2013.

H. Rosales-Méndez, A. Hogan, and B. Poblete. Voxel: a benchmark dataset for multilingual entity linking. In *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II 17*, pages 170–186. Springer, 2018.

E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair. More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, 12(2-3):140–157, 2018.

Ö. Sevgili, A. Panchenko, and C. Biemann. Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop*, pages 315–322, 2019.

W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2014.

W. Shen, Y. Li, Y. Liu, J. Han, J. Wang, and X. Yuan. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

B. Thijs. Using neural-network based paragraph embeddings for the calculation of within and between document similarities. *Scientometrics*, 125(2):835–849, 2020.

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

W. Zeng, J. Tang, and X. Zhao. Entity linking on chinese microblogs via deep neural network. *IEEE Access*, 6:25908–25920, 2018.

Z. Zheng, F. Li, M. Huang, and X. Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491, 2010.