Radboud University Nijmegen

Faculty of Science

Entity Linking for Greek

Thesis MSc Computing Science

Supervisor:
Dr. Faegheh HASIBI

Author:
Stergios MORAKIS

Second reader:
Prof. Dr. Martha LARSON

Daily Supervisor:
PhD (c) Keon DERCKSEN

August 2021

Contents

Abstract

Entity linking refers to the task of linking entity mentions found in a text to their corresponding entities in a knowledge base. Entity linking is essential for many natural language processing tasks and various approaches have been proposed to address this task, mainly for the English language. The focus of this research is to develop an entity linking approach for the Greek language. To this end, we extend the Radboud Entity Linker (REL) toolkit to support modern Greek. REL employs a modular entity linking approach that consists of mention detection, candidate selection, and entity disambiguation components. Using a limited amount of annotated data in Greek, we investigate three different mention detection approaches using spaCy, Flair, and BERT and conclude that the mention detection step is the main hindrance to the development of an accurate Greek entity linking system. We also show that the disambiguation approach employed in REL can achieve high accuracy for the Greek language. This thesis furthers research on making REL a multilingual entity linker and can be used to extend REL to the Greek language.

Acknowledgments

I would first like to thank *Koen Dercksen*, a brilliant Phd student, for his contribution and immeasurable support in this work. I would also like to thank my supervisor *Faegheh Hasibi* for her guidance and feedback, as well as *Martha Larson* for her invaluable support. The last, well-deserved place is respectfully given to *Duck*. Duck is not a duck. He is my cactus. I named him this way because I got him when I first heard the term "Duck Debugging" and wanted some company during the lonely corona times. To the best of my knowledge, he has been a very supportive cactus. :)

Chapter 1

Introduction

1.1 Background & Motivation

The application of neural networks in the field of natural language processing has reduced the semantic gap between humans and technological systems by a large margin. This fact is partially owed to knowledge bases and their large amounts of stored data. A non-exhaustive list of such knowledge repositories includes Wikidata [28], DBpedia [35] and YAGO [37], where each of them has its own knowledge graph schema. Most knowledge bases support extensively the English language, as the data is usually derived from the English editions of digital encyclopedias they parse, such as Wikipedia [38]. For that same reason however, the resources they provide for other languages are restricted by the amount of voluntary effort that was invested in the corresponding language and the quality of language-dependent information extraction tools used. These knowledge repositories contain structured information about entities. An entity present in a knowledge base is an object representing a concept, such as a person or a location. The task of recognizing mentions of entities in text and disambiguating them to the corresponding entities in a knowledge base (KB) is called *Entity Linking* [27].

Common entity linking systems consist of three components: mention detection, candidate selection and entity disambiguation [12, 3].

- *Mention Detection*:
  The task of capturing a text fragment that is part of a given piece of text as a potential entity's mention. For instance, in the given sentence "*Paris is a capital.*", the system must be able to identify the word "*Paris*" as a mention referring to an entity.

- *Candidate Selection*:
  The candidate selection step aims to generate a subset of entities for each mention. This is usually achieved by ranking KB entities using a feature-based metric, such as the probability of a mention being linked to a specific entity, also known as *Commonness* [25]:

  $$Commonness = P(e \mid m) = \frac{n(m, e)}{\Sigma_{\acute{e}} n(m, \acute{e})} \tag{1.1}$$

  where the number of times entity e is the target of mention m is divided by the total number of times that the mention refers to any entity. In the example above, the candidate selection component should generate for the given mention "*Paris*" a set of candidate entities that includes "*Paris*" and "*Paris Hilton*", representing the capital of France and a person respectively.

- *Entity Disambiguation*:
  The purpose of this task is to disambiguate a given text fragment (mention), by mapping it to a single entity in a knowledge base or none. The mention's context usually plays an important role during the disambiguation step. In our example, the mention "Paris" should be linked to the entity "*Paris*", and not other entities like "*Paris Hilton*".

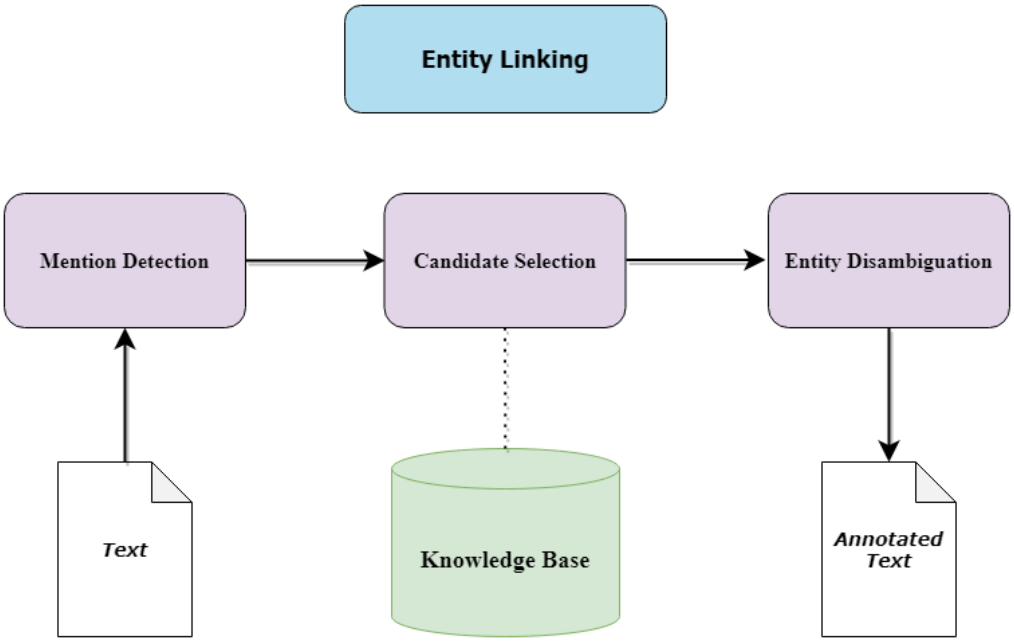


Figure 1.1: Abstract end-to-end entity linking process.

In this thesis we focus on developing an entity linking approach for modern Greek language, which is less studied in the literature compared to English or even other languages like German and Spanish [16]. The importance of the Greek language is immeasurable and we shall demonstrate it using a simple yet elegant example. In the modern Greek language, the meaning of the verb "σχολάω" (pronunciation: "scholao") is to finish your work. The root of that word comes from the Ancient Greece, where people would gather and discuss politics and philosophy after finishing their work, in the sense of self-education. Many years later, the Greek word "σχολείο" (pronunciation: "scholeo") was formed. Influenced by the Greek language, we have the Latin word "scholae". From the Latin, we have the English word "school" or even the French word "école" and many others. From this example, it is evident that the Greek language has not just lent words but whole concepts to the rest of the world.

Being a lesser resource language, modern Greek data is quite limited compared to English. Specifically, the latest Greek edition of Wikipedia's archive has a significantly small amount of documents at its disposal, compared to the respective English edition. Because of this, knowledge bases possess a limited portion of Greek language-dependent information, while some do not even support the language at all. Over the past years, researchers have published annotated Greek datasets, although small in size. This issue poses the question of whether entity linking suffers due to the lack of available data for this specific language and if so, which of its components are affected the most.

On its core, the Greek language differs significantly from English with respect to the grammatical functions of its nouns. In particular, the Greek language is polymorphic with regards to the suffixes of its nouns and the position of its diacritical mark, which depend on the noun's grammatical case. Table 1.1 shows the grammatical function of the Greek name "Αλέξανδος" (pronunciation: "alexandros"), meaning "Alexander". Consider the sentences "Alexander the Great has been influential in human history." and "Human history has been influenced by Alexander the Great.". Their respective translations to Greek are "Ο Μέγας Αλέξανδρος είχε επιρροή στην ανθρώπινη ιστορία." and "Η ανθρώπινη ιστορία έχει επηρεαστεί από το Μέγα Αλέξανδρο.". From this example, it is evident that an English based EL system has to consider a single case ("Alexander"), whereas a Greek one must consider multiple instances of the same entity ("Αλέξανδρος", "Αλέξανδρο", ...). These language-specific characteristics may have a negative impact on the candidate selection's utility, due to data sparsity and noise.

Case	**Singular**	**Plural**
Nominative	Αλέξανδρος	Αλέξανδροι
Genitive	Αλεξάνδρου	Αλεξάνδρων
Accusative	Αλέξανδρο	Αλεξάνδρους
Vocative	Αλέξανδρε	Αλέξανδροι

Table 1.1: The grammatical rules for the Greek name "Alexander" for different cases.

1.2 Objectives

In the context of this thesis, we aim to optimize an end-to-end entity linking system for the modern Greek language. Using the latest Greek archive of Wikipedia as our knowledge base, the system consists of mention detection, candidate selection and entity disambiguation components. We make use of the recently developed Radboud Entity Linker (REL) toolkit [3] and adapt it to the Greek language.

We formulate the following Research Questions:

- **Main Research Question:**
  - *How can we extend a state-of-the-art entity linking (REL) approach to support the modern Greek language?*
- **Sub-Research Questions:**
  - *Is the amount of Greek resources sufficient for a reliable mention detection component?*
  - *What is the impact of the entity disambiguation methodology used in REL on modern Greek text?*

1.3 Contributions

- We perform exploratory analysis on the latest (Feb. 2021) Greek Wikipedia and compare it to the respective English edition.

- We use Flair, BERT and spaCy Named Entity Recognition (NER) models and train them for the mention detection task.
- We adapt REL's Candidate Selection (CS) and Entity Disambiguation (ED) components to the Greek language and report the results.
- We propose changes to the REL codebase in various aspects:
  modularity, scalability, readability, testing

1.4 Structure

The rest of this thesis is structured as follows:

Chapter 2 covers several lines of related work. We provide a brief overview of knowledge bases, different components of an EL system, and related NLP models that can be used for developing an entity linking model.

Chapter 3 focuses on our overall approach. The chapter is divided in three parts. First, we focus on an experimental procedure, aimed to select an appropriate entity recognition model that performs well on the Greek language. We then cover our approach for candidate selection, and the last section focuses on the entity disambiguation step.

Chapter 4 introduces our datasets and describes the final results of our work for each individual component. We also include an exploratory analysis between the Greek and English editions of Wikipedia, an evaluation of our end-to-end system, and an analysis on some test cases.

This thesis ends on Chapter 5, where we address our research questions and propose future directions for Greek entity linking.

Chapter 2

Related Work

The background literature in the domain of entity linking is quite rich indeed. We discuss various approaches for each separate component and cover how these components can be combined in a single framework.

Knowledge Bases

A knowledge base stores information in a structured form. Interlinked descriptions of entities, events and situations or abstract concepts can be represented in a knowledge graph using an RDF schema [34]. Here we describe two of these knowledge bases:

- **Wikidata**
  Wikidata was launched in 2012, allowing its users to edit in a collaborative setting, query and retrieve data in a fully multilingual form. It is hosted by the nonprofit *Wikimedia Foundation* and is used as a source of open data for other Wikimedia projects, while directly interacting with the Wikipedia project. A study on Wikidata's knowledge graph [8] highlighted that while Wikidata supports various languages simultaneously, many languages have little or no coverage at all. Despite this, Wikidata aspires to provide structured data for all Wikimedia projects in all languages.

- **YAGO**
  The Yet-Another-Great-Ontology (YAGO) project [37] is an open source semantic knowledge base developed in Max Planck Institute for Computer Science, Saarbrücken. Its content is automatically extracted primarily from Wikipedia, as well as other sources (WordNet, WikiData, GeoNames, etc) and is structured using the RDF Schema. Moreover, its data is interlinked to DBpedia and Suggested-Upper-Merged-Ontology (SUMO) ontologies' data, participating in the vision of Linked Data [33]. Earlier versions of YAGO were manually evaluated, proving a confirmed accuracy of 95%. Most importantly, its latest version YAGO4 extracts data from the whole Wikidata, providing additional data for many languages, including the modern Greek language.

Wikification

The concept of Wikification, using Wikipedia for entity linking, was introduced in 2007 through the paper "Wikify! Linking Documents to Encyclopedic Knowledge" [36]. Entity linking refers to the task of recognizing mentions in a text and associating them to

their corresponding entries in a knowledge base. In the *Wikify!* project, the researchers intended to link mentions to their corresponding English Wikipedia articles, where each Wikipedia article's title reflects a unique entity. Their experiments were evaluated using the precision, recall and f1-score metrics:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2.1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2.2}$$

$$F1.score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.3}$$

where *Positive* and *Negative* labels refer to the outcome of the model's predicted class. If the predicted class does not oppose the ground truth, that prediction is marked as *True*, otherwise *False*.

First, as a mention detection component, the researchers Mihalcea and Csomai applied a greedy algorithm for capturing all text fragments (n-grams) in the input text that match any of Wikipedia's document titles. The product of this *Candidate Extraction* process is a list of candidate entities per text fragment, which can then be forwarded to the next step, namely the Ranking step, for retrieving the final set of candidate entities per mention.

For ranking candidate entities per text fragment (mention), a numerical value is assigned to each mention-entity pair. The authors experimented with three methods:

- *TF.IDF*

  $$tf.idf(t, d, N) = tf(t, d) * log(\frac{N}{df(t, N)}) \tag{2.4}$$

  where t, d and N denote a specific term, document and the amount of documents available respectively. In this formula, the number of occurrences of term t in a given document d is multiplied with the log-smoothed inverse of the number of documents where that term appears.

- *Chi-Square Test of Independence*
  A broadly used measure to see whether distributions of categorical variables differ from each other. In this case, the authors used it as a means for determining whether a text fragment occurs in the document more frequently than it would occur by chance.

- *Commonness*
  Can be interpreted as an entity's popularity given a mention associated with it, as explained in the previous chapter.

When evaluating the three different approaches, Commonness achieved the highest F1-score (54.63), as opposed to *tf.idf* and x^2 *test* (42.82 and 42.30 respectively). This was an interesting finding indicating that a noisy yet simple estimate can be the most efficient way of collecting candidate entities. As Wikipedia's recommended style for filling documents with information has been the same since then, the computation of Commonness scores remains an effective way for retrieving the most likely entities for each mention.

During the entity disambiguation step, the researchers applied two different disambiguation algorithms for identifying the most likely meaning for a word in a given context:

- Knowledge-Based approach (Unsupervised)
  Using a contextual overlap measure between the context of a given mention's paragraph and the context of a candidate Wikipedia page. The mention is disambiguated to the Wikipedia page achieving the highest score.

- Machine Learning approach (Supervised)
  A Naive Bayes classification model using words and part-of-speech tags present in the given mention's local context and a list of most descriptive keywords detected in that mention's global context as features.

As the focus of this study was to evaluate the quality of the disambiguation system independently, this component was evaluated in a separate setting, meaning that the experiment was conducted under the assumption that the candidate selection stage produces 100% precision and recall. It is noteworthy that this is not realistic in an end-to-end entity linking system. Nevertheless, the latter approach outperformed the former and a new state-of-the-art system result was set.

Overall, the *wikify!* project is an exemplary end-to-end entity linking system. Not only has it emphasized the various bottlenecks and set the groundwork for such systems but most importantly, it has proven that Wikipedia, the largest digital encyclopedia to date, can be used as a reliable resource for entity linking. Would this statement hold true for the Greek edition of Wikipedia as well, considering its relatively small size shown in Table 2.1?

Language	Wiki	Articles	Total Pages
English	en	6.326.835	53.686.429
Greek	el	195.246	585.736

Table 2.1: English and Greek Wikipedia edition details as reported on Wikipedia's platform.

Entity Recognition

As can be noticed in the *wikify!* project, mention detection is a crucial ingredient for end-to-end entity linking. With regards to that project's approach to mention detection, the methodology of exact n-gram matching presents an obvious trade-off between recall and computational complexity, as the input text is parsed once for each entity's text label. Moreover, this approach would not be as effective for the Greek language, due to the various grammatical functions being applied on nouns. Therefore, the methods proposed for the task of named entity recognition can better suit mention detection for Greek.

Named entity recognition (NER) is defined as the task of identifying and classifying named entities present in text into a given set of class labels [23]. These text fragments are ofen deteced by employing a machine learning sequence model that classifies each token's type. Figure 2.1 illustrates the NER annotator's output, when applied on a sequence of tokens. The NER sequence model is applied on the tokenized sentence and annotates the text fragments "*Alexander the Great*" and "*Pella*" as person and location entities, respectively. A NER annotator tool that successfully predicts whether a sequence of tokens (words) refers to an entity or not, can be used for the mention detection step.

Named entity recognition has been extensively addressed in NLP research. Undoubtedly, the amount of precisely annotated English NER datasets is large [22, 40, 39, 32]. It has come as a surprise that the conducted research in that task for the modern Greek

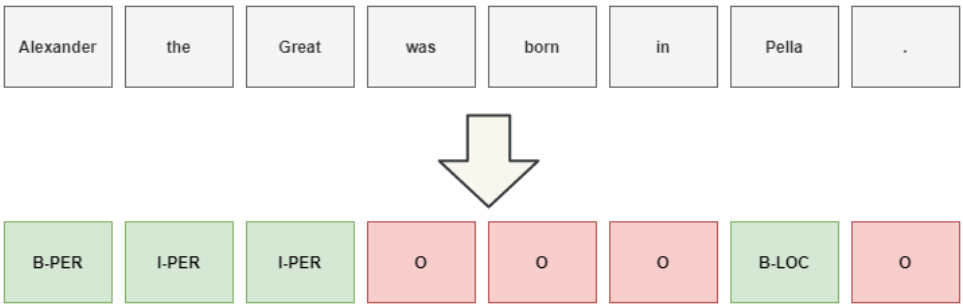


Figure 2.1: Input (top) and output (bottom) of NER sequence tagging using BIO (Beginning, Inside, Outside) format for prefixing entity types (Person, Location). As a result, two distinct Named Entities were successfully identified.

language is extremely limited [11, 1, 41]. In fact, to the best of our knowledge, the first well-structured and publicly available Greek NER dataset was published in 2018, as part of a project for the Google Summer of Code 2018 [1]. That project was a modern Greek language integration for spaCy [2]. Among its several deliverables, there existed a NER component that was trained on a fairly small corpus deriving from Greek newspaper articles. That dataset was annotated by I. Darras using Prodigy [3] on its (ORG, PERSON, LOC, GPE, EVENT, PRODUCT) class labels. This work has influenced many researchers and was quickly followed up in 2020 by Bartziokas, Mavropoulos, and Kotropoulos [1], who further expanded this work by introducing a manually annotated corpus of Greek newswire articles, specifically for the Greek named entity recognition task.

Word Embeddings

A wide variety of state-of-the-art NER tools make use of word embeddings to capture word semantics in text. An embedding is but a low-dimensional space in which high-dimensional vectors are represented, i.e. whole concepts represented by words. Consequently, it is possible for words with similar meanings to be placed closer together in the designated vector space, based on their contextual similarity. As different types of embeddings will constantly appear throughout this research, we feel obliged to provide a brief summary on this subject. In general, there is a class distinction on embeddings, i.e. distributed and contextualized representations.

The first class of embeddings comes from models designed to learn a distributed representation for words. The well-known Word2Vec [30] model, which is based on the continuous bag-of-words and skipgram models, falls in this category. Word2vec's neural network architecture consists of two layers and can be trained to reconstruct linguistic contexts of words in a vector space, so that words that share common contexts in the corpus are located close to one another. Then, we have the GloVe model [29], which uses matrix factorization techniques on the word-context co-occurrence matrix, where distant terms between contexts get penalised. One last popular type of distributed embeddings are generated using the Fasttext model [20], which creates word representations based on the sum of n-gram vectors in the corpus.

Contextualized representations could be characterized as dynamic word representa-

[1] `https://github.com/eellak/gsoc2018-spacy`
[2] `https://spacy.io/`
[3] `https://prodi.gy/`

tions. In this class, a word vector can dynamically change its values, depending on the given context. For example, in the sentences "He is running a company" and "He is running a marathon", the word "running" will adjust different values for each sentence, as opposed to distributed representations of words we previously mentioned. The models used for generating these types of embeddings are still advancing and continuously achieve new state-of-the-art results on various NLP tasks, such as sentiment analysis, question answering, named entity recognition. A non-exhaustive list of such models is Elmo [15], GPT2 [9], GPT3 [2] and BERT [7]. In this research, we make use of BERT as a deeply bidirectional language model and apply it on the task of NER. More details about this model is provided in chapter 3.

AIDA

The AIDA project [31] provides an approach for entity disambiguation. In this research, the authors emphasized the potential of combining the Wikipedia, YAGO and DBpedia knowledge bases. They used entities from YAGO and DBpedia, where both provide short names and paraphrases for their entries (in YAGO this data can be rerieved via the *means* relation which is an extension of Wikipedia's entities). Following the process of collecting KB entities, they cross-referenced them to Wikipedia articles using the *SameAs* class relation, forming a graph representation with unique, aggregated entities as nodes.

As knowledge graphs can be viewed as a *Directed Acyclic Graph* of classes, the authors considered contextual-like relations, such as the *Type* and *SubclassOf* class relations, for computing similarity scores between entities. In addition, they computed Commonness score similarities between entities and their mentions using the frequencies of Wikipedia link anchor texts. A third similarity score was computed as a weighted word overlap between the context of an entity's mention and that entity's related keywords. By combining these three similarity scores, representing all entity-entity and entity-mention relations in a single objective function, they formed a graph that would be used for inference, with mentions and entities as nodes and weighted, undirected edges between all entity-entity and mention-entity pairs.

The researchers also introduced a new entity disambiguation dataset by manually annotating the named entities present in the *CoNLL* NER dataset with their respective YAGO2 entities. Their methodology, along with other variants of their approach, were evaluated on that dataset. An interesting finding of that research was their system's underperformance when solely relying on the computation of Commonness for retrieving rare candidate entities, as opposed to other variants of their proposed function of combining entity popularity, similarity, and graph-based coherence.

Neural Attention for Entity Disambiguation

In 2017, a novel neural network-based approach for entity disambiguation was published by Ganea and Hofmann [18]. In this work, the authors introduced an *attention* mechanism for local entity disambiguation, responsible for capturing the most informative words for the disambiguation process and obtained close to state-of-the-art results.

One year prior, there was extensive research being conducted on ways of generating entity embeddings for entity disambiguation that best capture the semantic similarities between them [24, 21, 26]. That line of work was followed by Ganea and Hofmann. The researchers made use of word2vec pretrained vectors and extended them for creating entity embeddings from their canonical entity pages and local context of their hyperlink annotations. Their utility was enhanced by combining them with word embeddings in

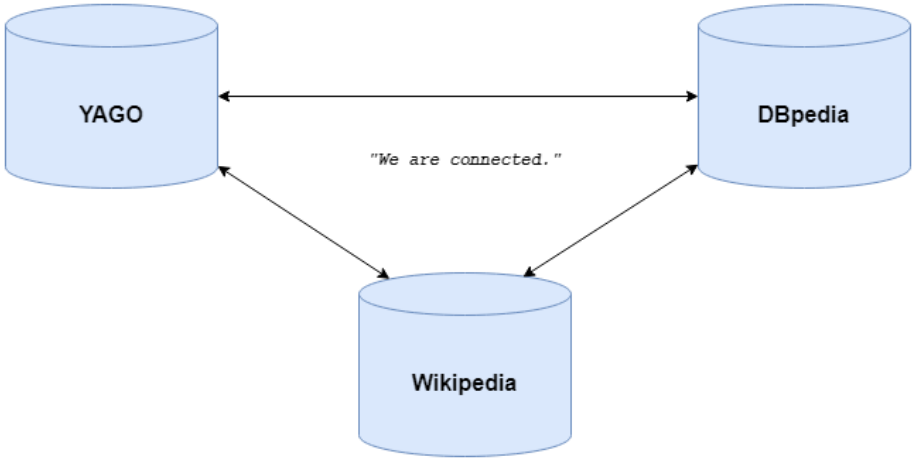
the same vector space. This approach was based on the assumption that ambiguous mentions can be resolved using just a few informative words present in their context. The combination of these context-based scores with the entities' Commonness scores were used for training their local model with neural attention, which would be further optimised using *Adam* optimiser [19].

As it happens in any end-to-end entity linking project, an obstacle in the researched approach has been the recall score for the candidate selection component. In order to reduce the number of disambiguating possibilities, they pruned entities that would not exceed a certain minimum boundary of Commonness score when evaluating their model. As a result, during the entity disambiguation step, a correct entity may not be listed in the set of candidates. Nevertheless, they share excellent results on the AIDA-CoNLL dataset among others (MSNBC, AQUAINT, ACE2004, WNED-WIKI, WNED-CWEB), which is considered the hardest according to the authors based on the Commonness baseline scores.

Radboud Entity Linker

The Radboud Entity Linker (REL[4]) project [3] was developed in 2020 as a promising open-source product of the Radboud University. Following a modular architecture, that project employs state-of-the-art approaches for mention detection, candidate selection and entity disambiguation, and generates a single end-to-end entity linking system. For instance, although REL's approach to mention detection is dependant on Flair's NER tagger [5], it is possible to effortlessly replace it with either exact n-gram matching against a dictionary of entities or spaCy's NER tagger. Additionally, REL allows using different versions of Wikipedia for training its candidate selection and entity disambiguation components, which are inspired by [18] and [14]. The experiments were performed on the 2014 and 2019 versions of Wikipedia's English edition and achieved competitive results on the GERBIL platform [17]. This Master's thesis is greatly influenced by this project and the numerous lines of work that preceded it. REL's strength of combining different neural approaches and harmonically balancing the throughput among them, undoubtedly suits the needs of this research.

[4] `https://github.com/informagi/REL`
[5] `https://github.com/flairNLP/flair`



Chapter 3

Approach

In this Master's thesis, we make use of REL's modular architecture and extend its general approach to entity linking for the Greek language. Although this was not the main focus of this research, a significant amount of time was invested in engineering solutions of major importance for that project, including flexible language dependencies. Following REL, our project's workflow consists of the mention detection (MD), candidate selection (CS) and entity disambiguation (ED) components, for combining and evaluating the system's sub-parts. We develop our approach using Wikipedia's Greek edition (version Feb. 2021), YAGO4 Knowledge Base and a Greek NER dataset, elNER) [1]. We expect the underlying ideas, designs, and suggestions encourage future work in the domain of Greek entity linking.

3.1 Mention Detection

As described before, the mention detection component is a core idea in end-to-end entity linking systems. Its purpose is the detection of text fragments in unstructured text that would represent potential entities. We shall refer to these text fragments as mentions. Since this project focuses on an end-to-end pipeline, it is preferable to avoid an excessive amount of mentions, as they would lead to noise. At the same time, we require as many mentions as possible, to provide high recall for the rest of the components. In an attempt to strike that balance between precision and recall, we consider three different alternatives of sequence labeling tools and compare them on the task of mention detection. The list of candidate tools to be used for the MD component is the following:

- *spaCy*
  SpaCy is an open-source library. It is efficient, robust and achieves close to state-of-the-art performance on a variety of NLP tasks. Although spaCy is designed specifically for production use instead of research purposes, its creators claim that spaCy is built on the latest research, providing a variety of practical tools and deployable models for text processing, supporting multiple languages.

- *Flair*
  Flair [6] is a robust open-source framework for NLP. With a unique architecture of promoting code readability and ease-of-use, it allows users to combine very different types of word embeddings while abstracting the process. Embeddings are word representations and are typically pre-trained by optimizing an auxiliary objective in a large unlabeled corpus, such as predicting a word based on its

context. In fact, the elegant combination of embeddings is the core concept of Flair. Backing this library is a curated collection of disributed embeddings available for the community, such as GloVe embeddings[29], Byte embeddings [13], FastText embeddings [20], Flair embeddings and more. Among other things, Flair can be used for training, optimizing and deploying models for the task of Named Entity Recognition.

- *BERT*
  BERT [7] stands for Bidirectional Encoder Representations from Transformers and is yet another sequence annotator tool worth investigating. Generally, BERT is used to pretrain deep bidirectional representations (embeddings) from unlabeled texts by jointly conditioning on both left and right context in all layers. As a result, a word vector can take different values depending on its context. By doing so, it learns its mask language model (MLM), which can be described as the process of predicting terms in hidden text fragments, where the prediction is a probability distribution over BERT's fixed vocabulary of words. A final NER output layer can then be added before fine-tuning the model for mention detection. Deployed BERT models in downstream NLP sub-tasks achieve excellent performance, especially when applied on NER, as BERT's representations capture useful and separable information about a term using other words in the vocabulary.

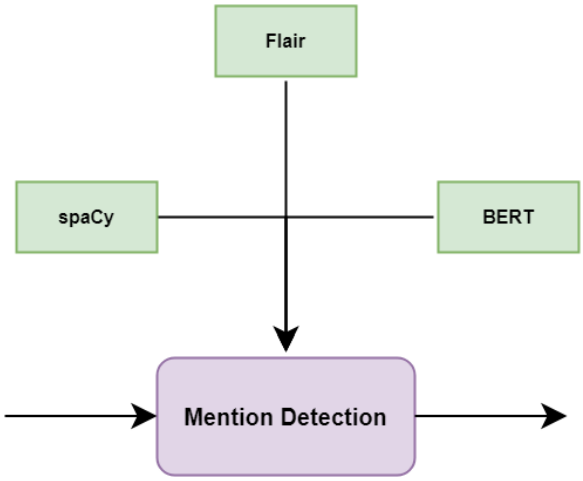


Figure 3.1: Candidate models for the task of mention detection.

The spaCy NER tagger is a black-box implementation with a transition-based chunking model approach to named entity recognition, directly constructing representations of multi-token entities. Its sequence model was trained using the Google Summer of Code 2018 project's NER dataset and was optimized on the task of identifying a total of six (ORG, PERSON, LOC, GPE, EVENT, PRODUCT) entity types in text. We use that exact NER sequence model as one of the candidate MD tools which is reported to achieve an F1-score of 0.77 (unknown whether this score refers to Micro or Macro F-measure).

Flair's entity recognition tagger was extensively used in [3] as the main part of MD component. Following that project's approach, we trained our own Conditional Random Field model for the task of mention detection using the elNER dataset. Moreover, we adjusted the NER tags so that the output labels represent a token being predicted as an entity or not, following the BIO format (B-E, I-E, O, where "E" stands for entity).

As a result, the intention of Flair's sequence model is not aimed for retrieving a token's specific class type, but rather the detection of a named entity's mention. For training this model we used Flair's *StackedEmbeddings*, stacking Greek fastText embeddings (where words are represented by the sum of the n-gram vectors) on top of Greek Byte embeddings (precomputed representations on the subword-level). The model's training and optimization (using Adam [19]) parameters can be found in Appendix A.1.

For the BERT NER tagger's training process we made use of the huggingface [10] open-source library. The huggingface package consists of various state-of-the-art Transformer architectures ready to be used, along with a collection of publicly available pretrained models. We used Greek-BERT's [4] masked language model for optimising our NER sequence model on the same dataset as in Flair's case. Greek-BERT is a pretrained model available in the huggingface repository that was trained on the Google Summer of Code 2018 dataset, plus another slightly larger dataset which is not publicly available anymore. In addition, we followed Greek-BERT's preprocessing steps of lowercasing the text and removing Greek accents from the data available. The parameters used for training and optimizing this model are shown in table A.2.

For inference, the mention detection component is bound to an entity recognition tool using the Strategy design pattern, a behavioral software design pattern that enables selecting an algorithm at compile or run time. For each of the ER tagger's detected mentions, the component uses the Natural Language Toolkit [1] package's tokenizer for capturing 100 tokens surrounding a detected mention. A rule-based approach follows, in which the mention is compared against a rich set of previously encountered mentions during the training process of the candidate selection component, in an attempt to satisfy the need of finding its set of candidate entities. This procedure is necessary, as CS component is unable to provide entities on unseen mentions. The rules are applied recursively and apply minimal transformations on the detected mention's text fragment, e.g. uppercase, lowercase, upper casing the first letter of each word, removing accents, etc. The collected information along with the program's control is then forwarded onto the next component, that is the candidate selection component.

3.2 Candidate Selection

During the candidate selection step we aim to reduce the number of disambiguating possibilities for a given mention. Our approach for providing a finite set of entities most related to the inputted mention relies on the computation of the probability $P(E|M)$, where E stands for entity and M is the mention. We consider the top 100 entities for each mention based on their Commonness scores, which will then be forwarded to the entity disambiguation component.

Following the construction of the $P(E|M)$ index in REL, an extended version of WikiExtractor [2] was used and applied on the latest Greek wikipedia archive (April, 2021). WikiExtractor is a tool used for extracting and cleaning text from a Wikipedia database dump. By extending its functionality, we generated (a) multiple files containing clean Wikipedia document texts with annotated per-document information (document title, ID and URL) and hyperlinks in HTML format, and (b) three additional files as follows:

- "*wiki_name_id_map.txt*"

  Containing 278.267 ***document_title*** - ***document_ID*** pairs, as extracted from the Greek wikipedia dump. Placeholder **document_title** is the Greek Wikipedia

[1] `https://www.nltk.org/`

[2] `github.com/attardi/wikiextractor`

article's decoded and stemmed URL (usually this field represents the document's title), while **document_ID** is a unique ID that can be used via the "*https://el.wikipedia.org/wiki?curid=****document_ID*** " URL for accessing that article.

- "wiki_redirects.txt"

  Containing 93.791 triples of ***source_document_title*** - ***destination_document_title*** - ***source_document_ID***.
  On the Greek Wikipedia's server side, both decoded and stemmed URLs refer to the same article (**source_document_title**) although that article can be accessed by clients using either one of them. Note that some **source_document_title** and **source_document_ID** entries were not present in the "wiki_name_id_map.txt" file.

- "wiki_disambiguation_pages.txt"

  Containing 8 ***document_ID*** - ***document_title*** pairs. These Wikipedia articles have multiple sections, each dedicated to different concepts. For instance, the Greek ambiguous word for "stretching" may refer to a technique used on newly acquired ropes, to a stomach's condition or even to the warm-up before exercising. All these distinct concepts are explained in that same Wikipedia article.

Excluding the Wikipedia documents present in the disambiguation pages, all distinct Wikipedia document generated titles were used as indexes for instantiating our Ground Truth entity objects. In the case of "wiki_redirects.txt" file's entries, we considered the source documents if their linked redirect target (**destination_document_title**) already exists as an entry in the "*wiki_name_id_map.txt*" file. Moreover, as each redirect relation present in the "wiki_redirects.txt" file can be represented as an Directed Acyclic Graph, for each **source_document_title** entity present in the "wiki_redirects.txt" file, we cached that entity's ID in its targeted entity's object. This way, it is possible to retrieve recursively the root of an entity's redirect. Throughout this research, all encountered entities are replaced by their root version, if such version exists. We report 189.115 root entities out of the 278.296 in total parsed Ground Truth entities.

In addition, as the majority of hyper references found in the Greek Wikipedia documents were noisy (typos, HTML tags, etc.) and could not be matched to the Knowledge Base's entities, we cached a simplified version of each article's title. The following algorithm shows the rule-based approach we used for simplifying the documents' titles:

1. Lowercase the current entity's name.
2. Transform HTML code, e.g. "&" to "&".
3. Replace underscores with spaces.
4. Remove consecutive spaces and spaces in the beginning or end of that entity's name.
5. Remove accents and apostrophes.
6. Transform the first character to Greek,
   if there is only one English character (that first one) and Greek characters exist. It is known that some Greek letters look identical to English ones once capitalized. This last step aims to solve the issue of avoidable mismatches caused by the authors' accidental typos in Wikipedia URLs.

This step-by-step, ad-hoc approach improved the resulting recall scores for entity matching when constructing the $P(E|M)$ index, with 2.671.254 entities being matched correctly to KB entities (an improvement of 3.715 matches as opposed to not using that algorithm), while 391.727 could not be matched and were marked as invalid. The majority of invalid matching attempts have been either Wikipedia documents that had not been filled yet or external URLs redirecting the user to e.g. old Greek newspaper articles. This algorithm is being used if and only if the first attempt of matching the original form of hyper references to ground truth entities proves futile (as there could be cases where simplified title overlaps appear).

As a next step, we parse all Wikipedia documents in our disposal. We track the hyperlink counts linked to mentions using that mention as an index, as long as that hyperlink can successfully be mapped to a KB entity. In case it is infeasible to match it to a knowledge base entity using its current form, we apply the rule-based transformation described above and attempt yet again to match its updated form to any simplified KB entity's hyperlink. This procedure for computing the Commonness score per mention encountered is explained using pseudo-code in algorithm 1.

An improvement was engineered in REL's codebase, as its approach to parsing Wikipedia hyperlinks could be considered inaccurate. Specifically, there were some rare cases where the Wikipedia document would provide an HTML hyperlink in between "«" and "»" symbols. This format would either not be considered or create unintentional hyperlink overlaps. These symbols get generated by the WikiExtractor project when replacing consecutive arrows ("<<", ">>") with them. The cases in which these symbols appeared were mainly related to documents focusing on pronunciation of letters or words.

```
Loading KB entities...
Loaded KB entities successfully.
        Total KB entities: "278296"
Parsing extracted wikidump...

--------
Parsing file "wiki_01"...
Parsing file "wiki_00"...
        Processed "500000" lines, valid hyperlinks: "601252", failed entity links: "79488"
        Processed "1500000" lines, valid hyperlinks: "1338542", failed entity links: "177293"
        Processed "2000000" lines, valid hyperlinks: "1659211", failed entity links: "229321"
        Processed "2500000" lines, valid hyperlinks: "1964761", failed entity links: "279416"
        Processed "3000000" lines, valid hyperlinks: "2254415", failed entity links: "330792"
        Processed "3500000" lines, valid hyperlinks: "2525069", failed entity links: "369745"
--------

Parsed extracted wikidump successfully.
        Total valid links: 2671254
        Total invalid links: 391727
Computing Wikipedia P(e|m) values...
P(e|m) values have been computed successfully.
P(e|m) index computed successfully.
        Total distinct mentions: "375519"

Test case:
        "Ηνωμένο Βασίλειο"
{'Ηνωμένο Βασίλειο': 0.989, 'Συμμετοχή του Ηνωμένου Βασιλείου στη Eurovision': 0.006, 'Ηνωμένο Βασίλει
```

Figure 3.2: Output when parsing Wikipedia documents and looking up entities ranked by Commonness for the term "Ηνωμένο Βασίλειο" (translation: "United Kingdom"). Commonness scores are rounded to 3 decimals following [3].

Algorithm 1: Commonness computation using Wikipedia hyperlinks.

```
Result: P_Wiki(E|M) Index
P_Wiki(E|M) = new HashTable();
for document in Wikipedia dump do
    for hyperlink in document do
        if hyper-reference found in KB
        (document titles as index) then
            initialize P_Wiki(E|M)[Mention] with anchor text;
            initialize P_Wiki(E|M)[Mention][Entity] with hyper-reference;
            P_Wiki(E|M)[Mention][Entity] += 1;
        else
            simplified hyper-reference = transform hyper-reference;
            if simplified hyper-reference found in KB
            (simplified document titles as index) then
                initialize P_Wiki(E|M)[Mention] with anchor text;
                initialize P_Wiki(E|M)[Mention][Entity] with hyperlink;
                P_Wiki(E|M)[Mention][Entity] += 1;
            else
                continue;
            end
        end
    end
end
for Mention in P_Wiki(E|M) do
    Mention size = total Entities linked to this Mention;
    for Entity in Mention do
        P_Wiki(E|M) [Mention][Entity] = Entity count / Mention size;
    end
end
```

The REL project makes use of a uniform probability $P_{YAGO}(E|M)$ deriving from [18], providing a file called *aida_means* with one-to-one relations between mentions and YAGO entities in English, based on YAGO2's *means* relation. Although we find ourselves uncertain on the way this file was generated, we investigated ways of generating our own mention-entity pairs for Greek. Unfortunately, the *means* relation is deprecated in more recent YAGO versions, while the older versions supporting the *means* relation do not support the Greek language. In addition, we are restricted to mapping the YAGO entity to a Wikipedia article, as Wikipedia is our only available Greek dataset for the Entity Disambiguation step. We deduced that the *means* relation is replaced by the *AlternateName* and *label* relations in the most recent YAGO version's *labels* class. We made our best effort for coming up for a solution of using the YAGO Knowledge Base and applied an ad-hoc approach for generating our own Greek $P_{YAGO}(E|M)$.

In this approach, we first identified all YAGO entity-mention pairs found in YAGO's *labels* class. That class contains mappings between YAGO entities and mentions via the *alternateName*, *label* and *comment* relations. We report 1.2% of these pairs to have a Greek relation. For each YAGO entity having either *alternateName* or *labels* relation to Greek mentions, we find its linked Wikidata entity (a Wikidata entity has a unique ID starting with capitalized Q, followed by a sequence of numbers) using the *sameAs* class. As a last step, we make throttled, well-tuned API requests to Wikidata's server for retrieving each Wikidata entity's most related Wikipedia article. After collecting data for approximately 2 months, we ended up with 102.101 mention - Wikipedia entity pairs and computed YAGO's Commonness score for each collected entity.

Following REL, we combine the Wikipedia and Yago $P(E|M)$ indexes into a final $P(E|M)$ index, resulting in 375.519 in total mentions. Additionally, in order to support future researches of combining multiple $P(E|M)$ indexes, we extend REL's formula $min(1, P_{Wiki}(E|M) + P_{YAGO}(E|M))$, by replacing the upper bound of 1 with the fraction of the total $P(E|M)$ indexers to be combined divided by 2.

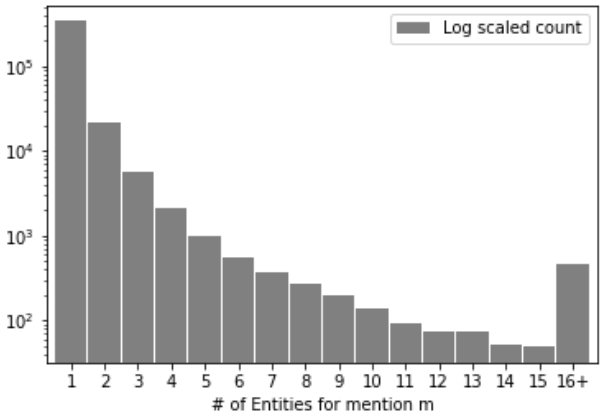


Figure 3.3: Log-scaled count for aggregated mentions, grouped by the number of entities assigned to them. The last bar from the left ("16+") represents the number of mentions containing 16 or more entities.

Additionally, following [24], we train word and entity embeddings, using as parameters 300 dimensions and a sliding window of size 8, based on the open-source Wikipedia2Vec [5] model. These embeddings capture the semantics in a word or an entity's context based on the Wikipedia dump's text and link structure that is given as input. The resulting embeddings were stored along with the final $P(E|M)$ index locally, using an SQLite [3] Database, signaling the end of the preprocessing steps for Candidate Selection.

During the e2e entity linking system's inference, we follow the lines of work REL was based on. Specifically, following [18], for a text fragment spotted as a potential mention to an entity during the MD step, we select up to $k_1 + k_2$ $(= 7)$ candidate entities. The k_1 $(= 4)$ candidate entities are selected from the top 4 entities for that given mention, ranked by Commonness scores. Following [14], the list of candidate entities is then completed with the top k_2 $(= 3)$ candidate entities, by computing the contextual similarity between the given mention and the top 30 entities potentially being linked to that mention, based on their obtained Commonness scores. That contextual similarity score is computed by $e^T \Sigma_{w \in c} w$, where $\mathbf{c}$ is an n-word $(n = 50)$ context surrounding the given mention, while $\mathbf{e}$ and $\mathbf{w}$ are word and entity embedding vectors. As a result, the next task, i.e. the entity disambiguation component, has to rank each of these candidates and assign the most promising one as the given mention's origin.

3.3 Entity Disambiguation

The entity disambiguation component intends to disambiguate a given mention to its ground truth entity, which is present in the list of candidate entities proposed by the

[3] `www.sqlite.org`

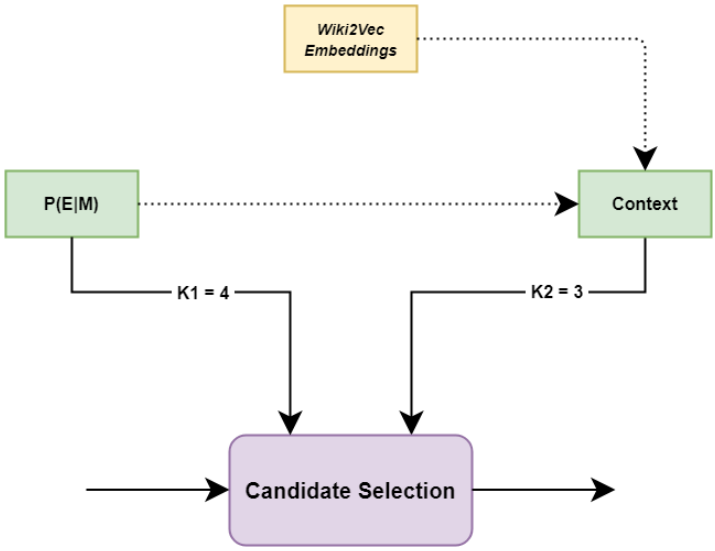


Figure 3.4: Inference during the candidate selection step for retrieving k_1 $(= 4)$ + k_2 $(= 3)$ = 7 candidate entities given a mention and its context.

Candidate Selection component. In our case, the set of ground truth entities derive from Greek Wikipedia document titles. Thus, with the addition of this ED component to our end-to-end entity linking system, a Greek Wikification process can be completed.

By following REL's implementation on Le and Titov's approach [14] on entity disambiguation, we first trained Greek GloVe (Global Vectors for Word Representation) [29] embeddings using the latest Greek Wikipedia archive. Similar to the Wikipedia2Vec model's embeddings, GloVe embeddings are useful for reconstructing linguistic contexts of words, using matrix factorization techniques on a word-context matrix. These embeddings were stored in the local SQLite database, in the same way as in Wiki2Vec embeddings' case.

As defined in REL, the linking decisions for a given document are based on the combination of local compatibility and coherence of other entity linking decisions:s

$$E^* = \underset{x \in C_1 * \ldots * C_n}{\arg\max} \sum_{i=1}^{n} \psi(e_i, c_i) + \sum_{i \neq j} \phi(e_i, e_j, D), \tag{3.1}$$

where C_i is the set of candidate entities for mention m_i and $E = \{e_1, ..., e_n\}$, the ψ function (following [18]) captures the coherence similarity between entity e_i and its local context c_i and the ϕ function (following [14]) captures the coherence between all entity linking decisions in the current document D.

Same as in REL, equation 3.1 is optimized using loopy belief propagation and the final score for each candidate entity given of the input document's mention is obtained by a two-layer neural network that combines Commonness score with max-marginal probability of the entity. During the training process, our model minimizes the maximum-margin loss function and is optimised using Adam [19].

With respect to REL's codebase, we report that no language dependencies were required for this component, apart from the generation of the Greek GloVe embeddings. We detected some trivial oversights on its codebase, related to the tokenization of the context where spaces would be recognized as distinct words by the system. With minor modifications on its dataset parsing techniques, its component was effortlessly used with the proposed parameter settings found in A.3.

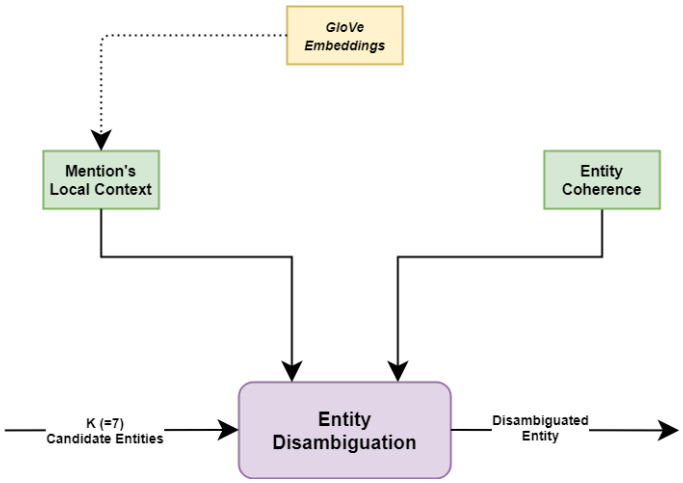


Figure 3.5: Inference during the entity disambiguation step for identifying a given mention's true entity using its context and past disambiguation decisions.

Chapter 4

Evaluation

4.1 Datasets

elWikiEL21

This subsection introduces our elWikiEL21 dataset, a dataset generated using the Greek edition of Wikipedia's 2021 version for entity linking. From that specific edition of Wikipedia (version Feb. 2021) we used, which consists of 276.024 in total documents, we worked on Wikipedia articles that did not redirect to other Wikipedia pages (i.e. were root articles and thus, we proceeded by discarding their duplicates) and do contain at least one paragraph. From that sample, we held-out randomly 1.000 articles for the development set and 500 articles for the test set. The documents present in the test set were excluded from all components, i.e. mention detection, candidate selection, entity disambiguation, as well as the Wiki2vec and GloVe embedding training process, and were used only once for obtaining the final evaluation of the system. The remaining documents were used as part of the training set, whereas the development set's articles were used for optimization purposes. This methodology is inspired by Jonathan and Olivier Raiman [16], who evaluated their English, French, German and Spanish Entity Linking classifiers on 1.000 held-out Wikipedia documents per Wikipedia edition.

The English Wikipedia has $10^5 > x$ active user base, while the French, German and Spanish Wikipedia editions have $10^5 > x > 10^4$. In contrast, Wikipedia's Greek edition is reported to have $10^4 > x > 10^3$ active users. Taking these differences into consideration, we applied an ad-hoc augmentation technique on the data used for the Entity Disambiguation component as an attempt to minimize potential underfitting. First, a list of hyperlink-mention pairs present in the document is kept in memory, as long as each pair's hyperlink is a valid KB entity. The document's title is then added to that list, as both an entity and a mention. Starting from the largest in length mention, all text spans in that document's first paragraph that match the current mention and are bounded by non-alphanumeric characters are recursively annotated using that mention's hyperlink, without allowing overlaps. As a last step, each document is replaced by its first, now augmented, paragraph for normalizing the varying article lengths. Following this approach we increased the total number of hyperlinks in every document's first paragraph that could be linked successfully to entities from 748.438 to 906.686, resulting in the data set splits shown in Table 4.1.

	Training set	Development set	Test set
Total articles	185.946	1.000	500
Total articles with at least one hyperlink	176.942	954	478
Total entities	912.072	4.858	2.577
Total distinct entities	131.063	3.169	1.822
Average entities per document	5.15	5.09	5.39
Standard deviation	3.60	3.65	3.80

Table 4.1: elWikiEL21's training, development and test set sizes, deriving from Greek Wikipedia articles.

elNER

The elNER dataset [1] is a publicly available manually annotated Greek corpus for facilitating research on the task of Greek named entity recognition. For its preprocessing and NER annotation, the spaCy and Prodigy tools were used. We combined its set of class labels (ORG, PERSON, LOC, GPE, EVENT, PRODUCT) into a single class (E, for entity), modifying its classification's purpose from multiclass to plain binary entity recognition. Table 4.2 provides an overview on that set's available data.

	Training set		Development set		Test set	
Count	17.132		1.904		2.116	
	Tokens	Entities	Tokens	Entities	Tokens	Entities
Mean	29.52	4.43	28.98	4.37	29.61	4.43
Std	17.76	4.05	17.60	4.06	17.78	3.97
Min.	1	1	1	1	1	1
Median	27	3	27	3	27	3
Max.	331	6	153	59	156	46

Table 4.2: elNER's dataset inspection

4.2 Wikipedia edition analysis

The latest Greek Wikipedia dump is compared to the English latest one in Table 4.3 and Figure 4.2. It is worth noting that some hyperlinks of Greek Wikipedia's referenced articles do not yet exist and will be filled in its future versions. Despite the small size of Wikipedia's Greek edition, the results suggest that the Greek edition of Wikipedia could overall be characterized as "compact", i.e. its authors attempt to reference a lot of Wikipedia articles in each of its documents. Nevertheless, when manually inspecting Greek articles, numerous text fragments, i.e. references to entities, were neglected during the annotation process.

	elwiki-latest	enwiki-latest
Total documents	280.980	15.788.310
Hyperlink count	3.103.222	108.106.829
Average document length (characters)	1718.7	924.98
Documents with length < 100 chars	34.45%	63.03%
Documents without hyperlinks	35.29%	62.8%
Average hyperlink count per document	11.04	6.85
Average hyperlinks per first paragraph	2.9	1.67

Table 4.3: Comparison on the latest Greek (el) and English (en) Wikipedia editions.

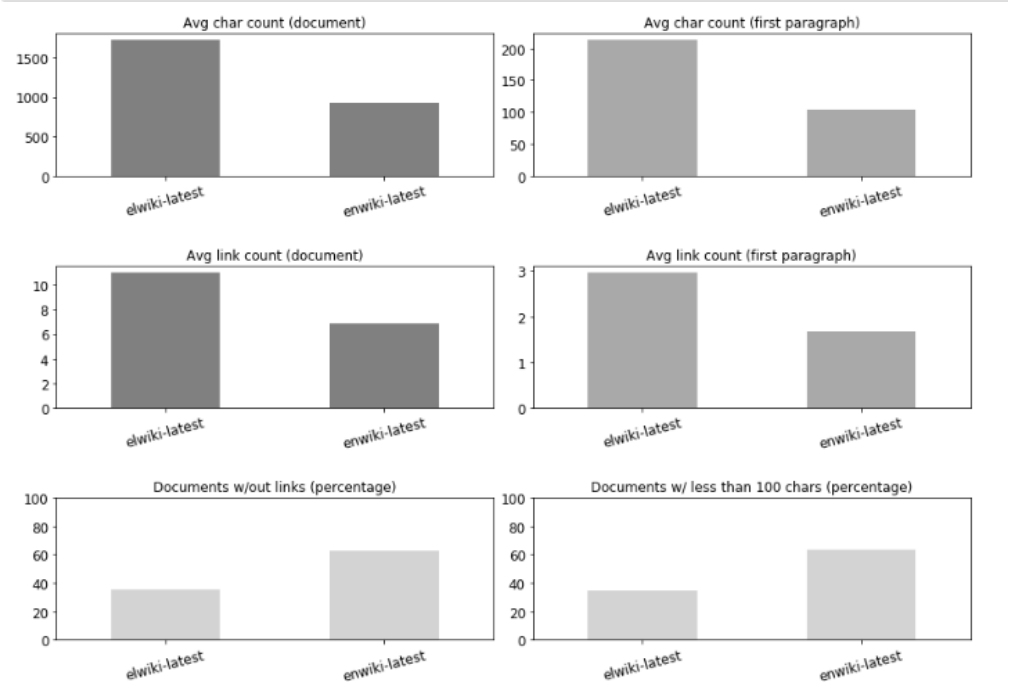


Figure 4.1: The first sub-figure (row 1, column 1) shows the difference on average article length between the two Wikipedia editions, i.e. Greek (el) and English (en). The second sub-figure (row 1, column 2) shows the difference on average first paragraph's length. Respectively, the third and fourth sub-figures show the difference on average hyperlink count on document (row 2, column 1) and first paragraph (row 2, column 2) level. Sub-figure 5 (row 3, column 1) shows the percentage of documents which do not provide any hyperlinks, while sub-figure 6 (row 3, column 2) shows the percentage of documents with exceptionally small document length.

4.3 Results

Entity Recognition

The nature of the mention detection component's role in e2e entity linking is to pass sufficient information to subsequent components, in the form of potential entity mentions. In this subsection, we showcase our results when comparing the spaCy, Flair and BERT sequence models on the task of entity recognition. As a kind reminder, the

spaCy NER model is trained on the Google Summer of Code 2018 data, whereas Flair and BERT are both trained using the elNER dataset's training set, optimised on its development set. We form a new set called *mdWiki*, consisting of the first paragraphs of 1.000 randomly sampled Greek Wikipedia articles deriving from the unprocessed el-WikiEL21 dataset's training set. In addition, we denote that same set as *mdWiki** with the data augmentation technique applied on. This procedure may be necessary in order for us to gain insight on how well the optimised NER models generalise on Wikipedia's semi-structured articles' anchor texts. Table 4.4 shows the models' achieved scores once applied on elNER's text, mdWiki and mdWiki* sets. For both mdWiki and mdWiki* sets, spaCy's senter component for splitting text into sentences was applied, as both Flair and BERT models were trained on a sentence level. In all cases we follow a soft evaluation methodology, where *True Positive* is considered an entity prediction that is a subset of a ground truth mention's span.

	elNER test set			mdWiki set			mdWiki* set		
	P	R	F	P	R	F	P	R	F
spaCy	-	-	-	**0.31**	0.40	0.35	**0.38**	0.43	0.40
Flair	0.94	0.93	0.94	0.18	**0.71**	0.29	0.19	**0.71**	0.31
BERT	**0.95**	**0.96**	**0.95**	0.28	**0.71**	**0.40**	0.34	0.68	**0.45**

Table 4.4: spaCy, Flair and BERT entity recognition models' precision (P), recall (R) and F1-score (F) results applied on elNER's test set, mdWiki set and data augmented mdWiki (mdWiki*) set.

The trained Flair and BERT models using elNER's training and development sets generalise very well on that dataset's test set, with the BERT model performing slightly better. That is not the case for the mdWiki set. In fact, the applied models' (i.e. spaCy, Flair, BERT) performance drops significantly when they encounter raw Wikipedia articles, where the highest F1-score is achieved by BERT at 0.45. A variety of reasons can justify this. First, we need to emphasize the difference between named entity recognition and mention detection. An optimised model on recognizing named entities is not optimised for mention detection in Wikipedia articles. Wikipedia articles contain links to both concepts and named entities, while NER datasets contain only annotations of named entities. Moreover, we need to take into account the mistakes made by the authors, e.g. a single character or a URL referencing an entity. This collection of unforeseen situations is reflected by each model's drop in recall. With regards to the lack of Wikipedia annotations, a notable drop on the models' precision scores is observed. Consequently, training NER models inputs annotated with both concepts and named entities (e.g. Wikipedia), is a promising future direction.

We find ourselves slightly more optimistic when looking up the models' performance on the mdWiki* set. That set could be characterized as mdWiki's variant, as the same paragraphs are used with our data augmentation technique applied on. Therefore, the focus of this evaluation process is the selection of an MD model that would perform well on Wikipedia's augmented data. In this case we would naturally expect an overall improvement of scores for all models, as Wikipedia's recommended style for its authors is not to annotate the document's title in its text, as well as not to annotate the same entity more than once in the same document. That improvement is present in each model's achieved Precision scores (and by extension F1-scores). The influence of our technique on the data had the least impact in Flair's case, which admittedly suffered throughout the evaluation process on both mdWiki and mdWiki* in terms of Precision. SpaCy, being the fastest, achieved the highest Precision score but underperformed in terms of Recall on the same task compared to the rest of the models, reaching the second highest

F1-score. Last but not least, BERT achieved the highest F1-score, with a pleasing balance between Precision and Recall. Based on the computed F1-scores of our models' evaluation on the mdWiki* set, spaCy and BERT models are both reasonable candidates to be used for the mention detection step.

We share the results of the spaCy, Flair and BERT models applied on elWikiEL21's yet unseen test set (Table 4.5). The achieved scores are better this time compared to mdWiki*'s results, which followed the same data augmentation technique and used a twice as large set of documents. Overall, elWikiEL21's test data are easier to be recognized as entities, compared to mdWiki and its variant mdWiki*'s data.

elWikiEL21 test set			
	Precision	Recall	F1-score
spaCy	**0.55**	0.44	0.49
Flair	0.28	0.66	0.39
BERT	0.45	**0.69**	**0.55**

Table 4.5: Mention detection evaluation on spaCy, Flair and BERT models applied on elWikiEL21 test set.

Candidate Selection & Entity Disambiguation

This subsection focuses on the evaluation of the candidate selection and entity disambiguation components. We report that the recall of the candidate selection component is 97.16% (Recall @7). The scores for the ED component are shown in Table 4.6. These results indicate that the disambiguation methodology used in REL and its predecessors can successfully be applied in the process of entity linking for the Greek language. We made sure to avoid any possibility of data leakage during the training process. We expected the polymorphic behaviour of Greek nouns to have a negative impact on these two components. That is not the case for the Greek Wikipedia dataset, in which grammatical functions on nouns have reduced the number of candidate entities and eased the process. Lastly, we report that 51 out of the 61 cases (Nil, denoting the absence of candidate entities for a detected mention) show issues that could be avoided if we removed the detected mentions' suffices and accents, as an attempt to imitate nouns in the English language.

elWikiEL21 test set			
Nil	Precision	Recall	F1-score
61	0.94	0.91	0.93

Table 4.6: Entity disambiguation results on elWikiEL21's test set. Nil corresponds to disambiguation cases in which no candidate entity in the KB could potentially be retrieved for a given mention.

End-to-End Entity Linking

We present the resulting end-to-end entity linking system's pipeline performance on the elWikiEL21's test set (Table 4.7). Out of the three potential tools to be used for the mention detection component, we used spaCy, due to its efficiency and ease of use compared to Flair and BERT respectively. Our computation process for these results, is inspired by the Gerbil platform's [17] *A2KB* type of experiment, i.e. a combination

of the Entity Recognition and Entity Disambiguation tasks. Consequently, we consider *True Positive* a mention detected as a subset of a ground truth anchor text and is successfully disambiguated to the same KB entity as the ground truth mention's entity. In contrast, *False Negative* is either a mention that was not detected successfully by spaCy or was detected but was not disambiguated precisely to its true underlying KB entity (the latter case counts as a *False Negative* as well for further penalising incorrect disambiguations). As the number of unidentified ground truth entities was increased, the decrease on Recall score compared to spaCy's independent evaluation on elWikiEL21's test data (0.44 Recall score) is justified.

elWikiEL21 test set		
Precision	Recall	F1-score
0.55	0.33	0.41

Table 4.7: End-to-End entity linking evaluation on elWikiEL21 test set.

Test Cases

We hand-picked some interesting test cases to shed some light on what is really happening in our system under test (SUT). We accurately underlined the anchor texts in each of the text's hyperlinks and used bold on the detected mention spans when using spaCy's tagger. When listing the candidate entities for a given mention, we underlined the SUT's predicted one as the most accurate disambiguated mention's KB entity.

Test Case 1:

Greek sentence:
"Ο Τομάς Μπαλκάσαρ Γκονσάλες (ισπανικά: 'Tomás Balcázar González¨, γεννημένος στις 21 Δεκεμβρίου 1931 στη **Γουαδαλαχάρα** του Μεξικού και αποβιώσας στις 26 Απριλίου 2020) ήταν **Μεξικανός** πρώην διεθνής ποδοσφαιριστής, ο οποίος αγωνιζόταν ως επιθετικός."

English translation:
"Tomas Balcázar González (Spanish: "Tomás Balcázar González", born in 21 December 1931 in **Guadalajara** of **Mexico** and dead in 26 April 2020) was a former Mexican international footballer who competed as an offensive player."

- Mention: "Γουαδαλαχάρα" - "Guadalajara"

Ground truth entity:
"Γουαδαλαχάρα (Μεξικό)" - "Guadalajara (Mexico)"

Candidate entities & selected entity (translated):
"Guadalajara (Mexico)", "CD Guadalajara", "Guadalajara (disambiguation)", "Province of Guadalajara", "CD Guadalajara (Spain)"

- Mention: "Μεξικού" - "Mexico"

Ground truth entity:
"Μεξικό" - "Mexico"

Candidate entities & selected entity (translated):
"Mexico", "Mexican National (Men's Soccer)", "Mexico U23 National (Men's Soccer)", "Mexico U20 National (Men's Soccer)", "Mexico City", "Mexican National (Men's Basketball)"

This Wikipedia paragraph features a combination of errors present in our system. First, it portrays a rare case in which Wikipedia authors link text representing dates to Wikipedia articles. Naturally, the Entity Recognition tagger is not trained on the task of recognizing dates. In addition, this example highlights our greatest concerns addressing the question: "When should a mention be identified as an entity?". Wikipedia's anchor texts are in many times simple words (e.g. "international" referring to Mexico's international football team and "footballer" referring to football as a sport), sometimes not sufficient to justify an entity's recognition. Therefore, it does not feel fair to assume an entity recognition tool's application on a Wikipedia corpus provides a good representation of that tool's performance. Moreover, the system failed to recognize the "Tomas Balcázar" mention, which was generated using our data augmentation technique (as this mention is that article's title), causing an increment of 1 on the total of *False Negatives*.

While "Mexico" completed successfully its end-to-end entity linking process, a happy accident occurred for the second detected by spaCy mention. Although, "Guadalajara" had as its most promising candidate entity its true ground truth entity, the ED component had other plans for it. Based on the mention's context, it computed a higher similarity score for its second most promising candidate entity, namely "CD Guadalajara", a professional football club based in Guadalajara. As the final prediction was wrong, our evaluation methodology counts this second end-to-end process as both a *False Negative* (as the ground truth entity was not retrieved successfully) and a *False Positive* (as the predicted entity does not match to the ground truth's entity). With a grain of humor we may characterise this result as close enough but far from perfect.

Test Case 2:

Greek sentence:
"Η **Άνθεια** είναι οικισμός της **Περιφερειακής Ενότητας Μεσσηνίας**, στην Περιφέρεια **Πελοποννήσου**, με πληθυσμό 248 κατοίκων, σύμφωνα με την Απογραφή του 2011. Διοικητικά ανήκει στην Κοινότητα Άνθειας και υπάγεται στη Δημοτική Ενότητα Θουρίας, του Δήμου Καλαμάτας."

English translation:
"**Antheia** is a settlement of the **Regional Unit of Messinia**, in the Region of **Peloponnese**, with a population of 248 inhabitants, according to the 2011 Census. Administratively it belongs to the Community of Antheia and belongs to the Municipal Unit Thourias, of the Municipality of Kalamata."

- Mention: "Άνθεια" - " Antheia"

Ground truth entity:
Nil

Candidate entities & selected entity (translated):
"Antheia (Argolida)", "Anteia (ancient city)", "Antheia of Patras"

- Mention: "Περιφερειακής Ενότητας Μεσσηνίας" - "Regional Unit of Messinia"

Ground truth entity:
"Νομός Μεσσηνίας" - "Prefecture of Messinia"

Candidate entities & selected entity (translated):
"Prefecture of Messinia", "Administrative division of Peripheral Unit of Messinia"

- Mention: "Πελοποννήσου" - "Peloponnese"

Ground truth entity:
"Περιφέρεια Πελοποννήσου" - "Region of Peloponnese"

Candidate entities & selected entity (translated):
"Peloponnese", "Region of Peloponnese", "Peloponnese theme", "Principality of Achaia", " Elialeti of Moria", " Univeristy of Pelopponisos"

A variety of interesting inference decisions occur in this test case. First, our augmentation technique was ineffective on marking "Atheia" as an entity, as the document's title "Atheia of Messinia" could not be matched as an exact n-gram. Nevertheless, the system was successful in not disambiguating the detected mention to its most popular choice "Antheia of Patras". Instead, based on the provided context, KB entity "Antheia (Argolida)" is considered the most appropriate choice and was satisfyingly selected, the difference being the two geographical locations for these non-identical places while both in Peloponnese. The second sequence of tokens ("Regional Unit of Messinia") that was detected by spaCy did not present competition in-between its two candidate entities and the one with the highest $P(E|M)$ (0.991) was correctly preferred.

The last mention for this test case i.e., "Peloponnese" shows some interesting complications. First of all, the mention "Πελοπόννησος" (nominative case) is extremely popular, as it is the largest peninsula in Greece, representing a geographical location. In contrast, entity "Region of Peloponnese" has a political or even administrative theme and is rarely referenced. During the Entity Disambiguation step, as the Wikipedia page for "Peloponnese" provides rich context and wider variety of uses in contrast to the "Region of Peloponnese" page's minimal information, which in many cases is not being annotated by the authors even though it would arguably be a better choice, the former was preferred. In other words, the system was not successful in selecting the ground truth entity "Region of Peloponnese" as the optimal choice, despite the plethora of administrative information provided, as "Peloponnese" is often the go-to choice by the authors. This example demonstrates how easily an entity linking system is distracted by inaccuracies and underlying bias during the Wikipedia articles' annotation process. Even though the usability of embeddings could overcome the entity's popularity and fill that semantic gap, the data could not support this claim.

Chapter 5

Conclusion

In this Master's thesis an end-to-end entity linking system for the Greek language was developed. That system consists of a mention detection component for detecting references of entities in text, a candidate selection component responsible for reducing the number of potential entities linked to the given mention and an entity disambiguation component for linking that mention to its respective true entity. We used the Feb. 2021 version of the Greek edition of Wikipedia and a subset of YAGO4 entities as ontology data.

With regards to the mention detection component, we tested spaCy, Flair and BERT models. SpaCy's model was originally trained on the Google Summer of Code 2018 Greek NER dataset and we proceeded by training the rest using the elNER dataset. We applied various techniques for comparing the three entity recognition models on the Wikipedia mention detection task. There was a notable difference when using a conventional NER dataset for training and applying them on Wikipedia articles for the task of mention detection. Although BERT performed better (0.55 F1-score), its usability was a non-trivial task and proceeded by embedding spaCy (0.49 F1-score) into our end-to-end system's mention detection component, after considering its ease of use, time efficiency and NLP preprocessing tools provided.

For the candidate selection component, we extended REL's codebase by improving its scalability via the use of design patterns and other object-oriented techniques (Appendix B.1). We used Wikipedia articles as our knowledge base entities and combined Commonness scores with wiki2vec embeddings for retrieving a set of candidate entities for a given mention. As a stand-alone component, candidate selection achieved notably high recall (0.97) on the augmented data of our Wikipedia test set, emphasizing the absence of unforeseen disambiguation decisions. On that same process, our entity disambiguation component that uses GloVe embeddings and a local attention mechanism, achieved a very good 0.93 Micro F1-score by considering the given mention's Greek context and the coherence between disambiguation decisions.

We shall now address the research questions posed in the beginning of this research, the first one being "*How can we extend a state-of-the-art entity linking (REL) approach to support the modern Greek language?*" The main question of our study can be addressed directly through the obtained results of each components' evaluation process, while considering the evaluation of the end-to-end system's application on elWikiEL21 test set. Based on these results, mention detection may be considered a bottleneck in entity linking for Greek. That is because none of the tools we investigated achieved a reliable outcome when applied on Wikipedia's articles. In contrast, the state-of-the-art techniques used in our research for candidate selection and entity disambiguation, perform very well on Greek texts, given sufficient context and well-annotated data.

Therefore, our applied methodology for Greek texts can extend entity linking systems.

The second question of whether the amount of Greek resources is sufficient for a reliable mention detection component can therefore be answered in a similar manner. Based on our results, the induced noise caused during the mention detection step makes a reliable end-to-end pipeline within our reach but far from perfect. As none of the MD models learned to identify entity mentions in Greek Wikipedia articles, we conclude that more data is required. Nonetheless, we have proven that, with the current state of available in Greek data, an entity linking system consisting of a candidate selection and an entity disambiguation component can fulfill the need of disambiguating a mention to its respective ground truth entity. Hence, to address our last research question regarding the impact of REL's ED approach on Greek text, such methodology performs exceptionally well in our case.

Limitations

A major limitation in our research has undoubtedly been the limited amount of well-structured annotated data for the task of entity linking, as we were restricted on using the noisy Greek edition of Wikipedia for this research. The modern Greek language encountered in Wikipedia was not as well annotated as we originally hoped and faced problems with both anchor texts and hyperlinks. In addition, the data that could be used were biased, for instance the excessive use of nominative case in anchor texts and the repetitive usage of articles as hyperlinks when other articles would be more precise hyper-references. Last but not least, we find ourselves unable to retrieve an entity when it does not currently exist in the available Greek knowledge base, i.e. that entity's respective Greek Wikipedia page does not exist or is not filled yet.

5.1 Future directions

1. *Data Annotation*
   Annotated datasets in the modern Greek language for any NLP-related field is currently a necessity. Greek entity linking datasets are in particular highly valued, where entity mentions are manually disambiguated to their corresponding entity in a knowledge base. That knowledge base could be either Wikipedia, Wikidata or YAGO. An alternative would be the Greek DBpedia, which during our research was out of service. We strongly suggest this task a necessary part for future research.

2. *Mention Detection*
   In our research, the mention detection component did not perform as expected on Wikipedia articles due to the NER data used for its training (Google Summer of Code 2018, elNER). We consider Wikipedia's data often inconsistent for this training process and easy to overfit, although it would be worth investigating i.e., training the models on non-overlapping Greek Wikipedia articles. In addition, more experiments should be conducted using using other variations of sequence labeling models (e.g. RoBERTa, DistilBERT, XLNet). Distinction between named entity recognition and mention detection tasks should be reflected by the data used for training, while respecting the end-to-end entity linking process' purpose.

3. *Modifications on Greek suffixes*
   In case future research encounters obstacles concerning the morphology of Greek noun suffices, we propose an experimental setup for comparing different techniques on the lemmatization and stemming NLP sub-tasks, of which the best approach

should be used as a data preprocessing step. The suggestion of these tasks are aimed to normalize distributions and bring state-of-the-art English entity linking techniques closer to Greek texts.

Bibliography

[1] Nikos Bartziokas, Thanassis Mavropoulos, and Constantine Kotropoulos. "Datasets and Performance Metrics for Greek Named Entity Recognition". In: *11th Hellenic Conference on Artificial Intelligence (SETN 2020)*. SETN 2020. Athens, Greece: Association for Computing Machinery, 2020, pp. 160–167. ISBN: 9781450388788. DOI: 10.1145/3411408.3411437. URL: https://doi.org/10.1145/3411408.3411437.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[3] 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*. July 2020. DOI: 10.1145/3397271.3401416. URL: http://dx.doi.org/10.1145/3397271.3401416.

[4] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. *GREEK-BERT: The Greeks visiting Sesame Street*. 2020. arXiv: 2008.12014. URL: https://arxiv.org/abs/2008.12014.

[5] 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*. 2020. arXiv: 1812.06280 [cs.CL].

[6] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. "FLAIR: An easy-to-use framework for state-of-the-art NLP". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 2019, pp. 54–59.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[8] Mariam Farda-Sarbas and Claudia Müller-Birn. *Wikidata from a Research Perspective - A Systematic Mapping Study of Wikidata*. 2019. arXiv: 1908.11153. URL: http://arxiv.org/abs/1908.11153.

[9] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. *Language Models are Unsupervised Multitask Learners*. 2019.

[10] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. arXiv: 1910.03771. URL: http://arxiv.org/abs/1910.03771.

[11] I. Angelidis, Ilias Chalkidis, and M. Koubarakis. "Named Entity Recognition, Linking and Generation for Greek Legislation". In: *JURIX*. 2018.

[12] Krisztian Balog. *Entity-Oriented Search*. Oct. 2018. ISBN: 978-3-319-93935-3. DOI: 10.1007/978-3-319-93935-3.

[13] Benjamin Heinzerling and Michael Strube. "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. URL: https://www.aclweb.org/anthology/L18-1473.

[14] 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)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1595–1604. DOI: 10.18653/v1/P18-1148. URL: https://www.aclweb.org/anthology/P18-1148.

[15] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. *Deep contextualized word representations*. 2018. arXiv: 1802.05365. URL: http://arxiv.org/abs/1802.05365.

[16] Jonathan Raiman and Olivier Raiman. *DeepType: Multilingual Entity Linking by Neural Type System Evolution*. 2018. arXiv: 1802.01021. URL: http://arxiv.org/abs/1802.01021.

[17] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. *GERBIL - Benchmarking Named Entity Recognition and Linking consistently*. 2018. DOI: 10.3233/SW-170286. URL: https://doi.org/10.3233/SW-170286.

[18] 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*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2619–2629. DOI: 10.18653/v1/D17-1277. URL: https://www.aclweb.org/anthology/D17-1277.

[19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[20] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. *Enriching Word Vectors with Subword Information*. 2016. arXiv: 1607.04606. URL: http://arxiv.org/abs/1607.04606.

[21] Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. "Entity Disambiguation by Knowledge and Text Jointly Embedding". In: Jan. 2016, pp. 260–269. DOI: 10.18653/v1/K16-1026.

[22] Abbas Ghaddar and P. Langlais. "WikiCoref: An English Coreference-annotated Corpus of Wikipedia inprocceedingss". In: *LREC*. 2016.

[23] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. *Neural Architectures for Named Entity Recognition*. 2016. arXiv: 1603.01360 [cs.CL].

[24] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. *Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation*. 2016. arXiv: 1601.01343 [cs.CL].

[25] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. "Entity Linking in Queries: Tasks and Evaluation". In: *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: Association for Computing Machinery, 2015, pp. 171–180. ISBN: 9781450338332. DOI: 10.1145/2808194.2809473. URL: https://doi.org/10.1145/2808194.2809473.

[26] Hongzhao Huang, Larry Heck, and Heng Ji. *Leveraging Deep Neural Networks and Knowledge Graphs for Entity Disambiguation*. 2015. arXiv: 1504.07678 [cs.CL].

[27] Wei Shen, Jianyong Wang, and Jiawei Han. *Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions*. 2015. DOI: 10.1109/TKDE.2014.2327028.

[28] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. "Introducing Wikidata to the Linked Data Web". In: *The Semantic Web – ISWC 2014*. Ed. by Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble. Cham: Springer International Publishing, 2014, pp. 50–65. ISBN: 978-3-319-11964-9.

[29] 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)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162.

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[31] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. "Robust Disambiguation of Named Entities in Text". In: *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland*. 2011, pp. 782–792.

[32] Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. "OntoNotes: A Large Training Corpus for Enhanced Processing". In: Jan. 2011.

[33] Christian Bizer, Tom Heath, and Tim Berners-Lee. *Linked Data: The Story so Far*. July 2009. DOI: 10.4018/jswis.2009081901.

[34] Vassilis Christophides. "Resource Description Framework (RDF) Schema (RDFS)". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 2425–2428. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_1319. URL: https://doi.org/10.1007/978-0-387-39940-9_1319.

[35] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web*. Ed. by Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735. ISBN: 978-3-540-76298-0.

[36] Rada Mihalcea and Andras Csomai. "Wikify!: Linking Documents to Encyclopedic Knowledge". In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. CIKM '07. Lisbon, Portugal: ACM, 2007, pp. 233–242. ISBN: 978-1-59593-803-9. DOI: 10.1145/1321440.1321475. URL: http://doi.acm.org/10.1145/1321440.1321475.

[37] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: A Core of Semantic Knowledge". In: *16th International Conference on the World Wide Web*. 2007, pp. 697–706.

[38] Susan L. Bryant, Andrea Forte, and Amy Bruckman. "Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia". In: *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work*. GROUP '05. Sanibel Island, Florida, USA: Association for Computing Machinery, 2005, pp. 1–10. ISBN: 1595932232. DOI: 10.1145/1099203.1099205. URL: https://doi.org/10.1145/1099203.1099205.

[39] 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 - Volume 4*. CONLL ’03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 142–147. DOI: 10.3115/1119176.1119195. URL: https://doi.org/10.3115/1119176.1119195.

[40] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. “The GENIA Corpus: an Annotated Research Abstract Corpus in Molecular Biology Domain”. In: Jan. 2002, pp. 82–86.

[41] Iason Demiros, Sotiris Boutsis, Voula Giouli, Maria Liakata, Harris Papageorgiou, and Stelios Piperidis. *Named entity recognition in Greek texts*. Jan. 2000.

Appendices

Appendix A

Model parameters

Parameter	Value
MAX_LENGTH	256
MAX_EPOCHS	300
LEARNING_RATE	0.1
BATCH_SIZE	32
SEED	42

Table A.1: Flair training parameters for Named Entity Recognition.

Parameter	Value
MAX_LENGTH	256
BERT_MODEL	nlpaueb/bert-base-greek-uncased-v1
BATCH_SIZE	16
NUM_EPOCHS	5
SAVE_STEPS	750
SEED	42

Table A.2: BERT training parameters for Named Entity Recognition.

Parameter	Value
prerank context window	50
commonness entities	4
context entities	3
context window	100
dropout	0.3
max epochs	1000
learning rate	1e-4

Table A.3: Entity Disambiguation training parameters.

Appendix B

Class Diagrams

The student has spent a significant amount of time on implementation decisions for improving REL's codebase and would to share some of his work.

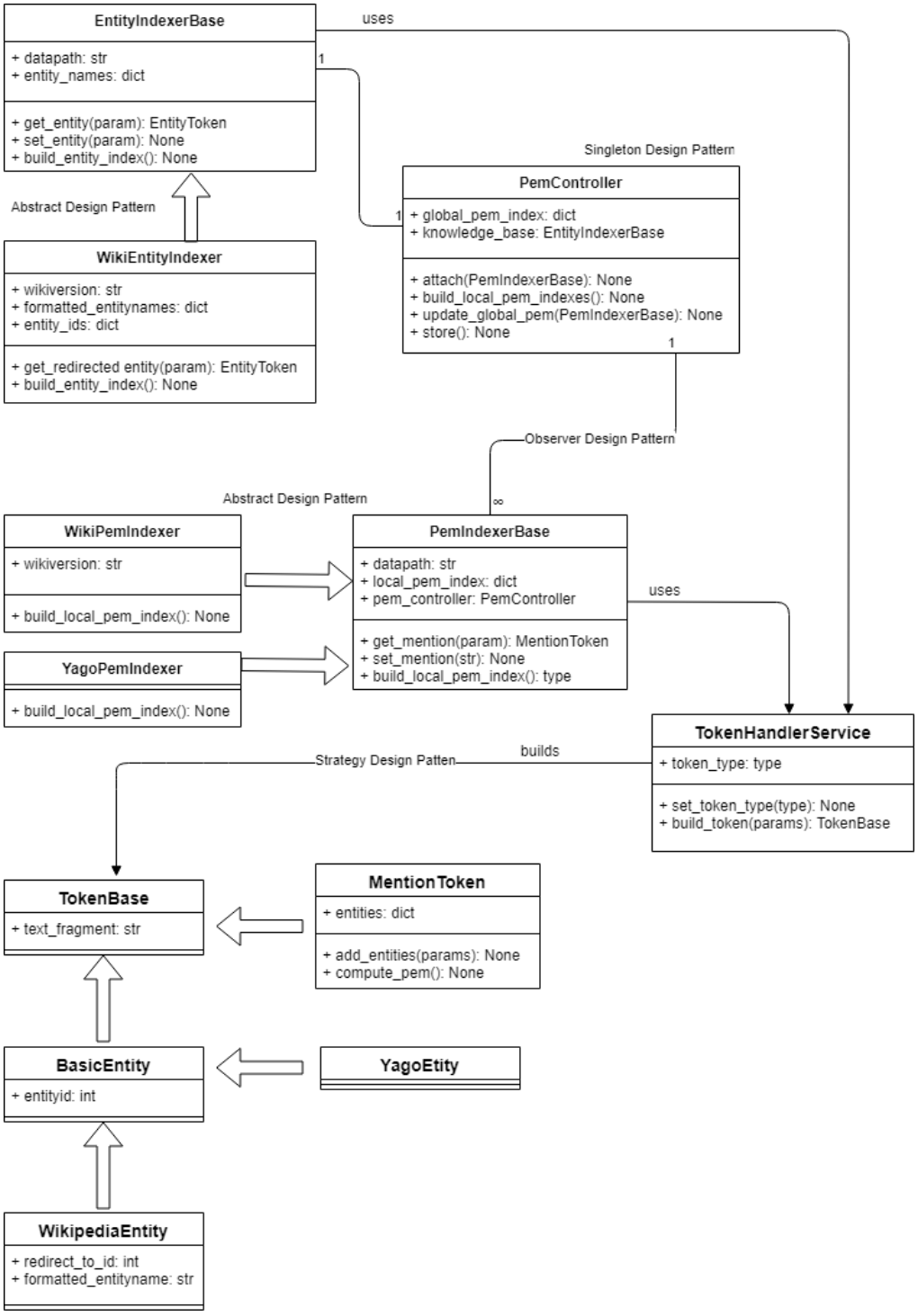


Figure B.1: Abstract implementation decisions for Commonness computation.