INTERNATIONAL
HELLENIC
UNIVERSITY

Extracting Structured Information from Greek Legislation Data

Michailidis Alexios

SID: 3308210026

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

5th January 2023

THESSALONIKI - GREECE

INTERNATIONAL
HELLENIC
UNIVERSITY

Extracting Structured Information from Greek Legislation Data

Michailidis Alexios

SID: 3308210026

Supervisor:	Dr. Berberidis Christos
Supervising Committee Members:	Assist. Prof. Peristeras Vassilios Dr. Magnisalis Ioannis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

5th January 2023

THESSALONIKI - GREECE

Abstract

Customers nowadays have difficulties in finding the relevant information due to information overload, mainly via the usage of the World Wide Web. The amount of data kept on the Internet has grown at an exponential rate in recent years. Furthermore, most information is released in an unstructured format, making it difficult to efficiently extract knowledge. But there is a huge demand for transparency, especially in the public sector, thus it is necessary to extract structured information. Except for the transparency issue, structured data may be examined further to get new information and insights. Although machine learning and NLP approaches, such as named-entity recognition and relation extraction, have recently demonstrated interesting outcomes in the research field of information extraction, the majority of current research has been devoted to English language material.

The purpose of this study is to extract structured information from Greek legislation texts, which are published in PDF format with no metadata, deviating a lot from Tim Berners-Lee's idea for linked open data. To be more detailed, we will first create a named entity recognition model for extracting the entities from the documents. Following that, we will fine-tune a transformers-based model to detect the relationships between these entities. Finally, we will combine them into a pipeline and extract structured information from plain text input.

Concluding, the present research contributes to studies that have already examined information extraction tasks in general, by proposing a particular approach that relies on transformer-based models for deriving entity relationships from text and providing structured information that can be further studied. Especially, to the best of our knowledge, the presented approach is the first initiative to extract relationships from Greek legal documents and might contribute to the country's desired Open Government agenda.

Acknowledgments

I would like to take this opportunity to thank my supervisor, Dr. Christos Berberidis, for all of his assistance and guidance during this project. Furthermore, I would also like to thank Mr. Ioannis Konstadinidis and Ms. Elena Kapantai for their patience, support, and direction in every part of this thesis. Their contribution to the result is substantial. Lastly, I would like to thank Thomas Papadopoulos, my real-life mentor who pushed me to apply to this master's program and encouraged me throughout my studies.

Contents

ABSTRACT **3**

ACKNOWLEDGMENTS **4**

CONTENTS **5**

1. INTRODUCTION **8**

2. THEORETICAL BACKGROUND **11**

2.1 INFORMATION EXTRACTION 11

2.1.1 Information extraction Tasks *12*

2.1.2 General Pipeline of Information Extraction *13*

2.2 PART OF SPEECH (POS) TAGGING 14

2.2.1 Approaches to POS Tagging *15*

2.3 NAMED ENTITY RECOGNITION (NER) 16

2.3.1 Approaches to NER *17*

2.4 DEPENDENCY PARSING 18

2.4.1 Approaches to Dependency Parsing *19*

2.4.2 Evaluation of Dependency Parsing *21*

2.5 COREFERENCE RESOLUTION 21

2.5.1 Anaphoric Types *22*

2.5.2 Anaphoric Constraints *23*

2.5.3 Approaches to Coreference Resolution *24*

2.5.4 Evaluation of Coreference Resolution *25*

2.6 RELATION EXTRACTION 26

2.6.1 Approaches to Relation Extraction *26*

2.6.2 Evaluation of RE *28*

2.7 USEFUL MACHINE LEARNING CONCEPTS 29
2.7.1 The Transformer Architecture *29*
2.7.2 The concept of Weak Supervision Learning *30*
2.7.3 The concept of Active Learning *31*
2.8 BACKGROUND ON NATIONAL PRINTING HOUSE (ET) 31
2.8.1 Structure of National Printing House *32*

3. RELATED WORK **34**
3.1 OPEN DATA INITIATIVES IN EUROPEAN UNION 34
3.2 INFORMATION EXTRACTION OF GREEK LEGAL DOCUMENTS 35

4. BASIC PRINCIPLES AND IMPLEMENTATION FRAMEWORK **39**
4.1 METHODOLOGY AND IMPLEMENTATION FRAMEWORK 39
4.2 DATASETS 41

5. IMPLEMENTATION AND RESULTS **45**
5.1 NAMED ENTITY RECOGNITION 45
5.1.1 Main process *45*
5.1.2 Enhance the Results *47*
5.1.3 Compare the Language Models *47*
5.1.4 Fine-Tune the Fine-Tuned Model *48*
5.2 RELATION EXTRACTION 49
5.2.1 Main process *49*
5.2.2 Enhance the Results *52*
5.3 PIPELINE 53
5.3.1 Remove Abbreviations inside Parenthesis *54*
5.3.2 Perform NER on the inputs *55*
5.3.3 Transform Data into a readable format for RE *56*

5.3.4 Perform RE on the transformed data *56*

5.3.5 Post-Processing of Relationships extracted *56*

5.3.6 Create a Knowledge Graph *58*

6. EXPERIMENTATION ON ENTIRE DOCUMENT **60**

6.1 PROCESS PERFORMED 60

6.2 OUTCOME OF ANALYSIS 62

7. CONCLUSIONS & FUTURE WORK **65**

7.1 CONCLUSIONS 65

7.1.1 Findings *66*

7.2 IMPLICATIONS 67

7.2.1 Theoretical Implications *67*

7.2.2 Management Implications *67*

7.3 LIMITATION AND SUGGESTION FOR FUTURE RESEARCH 68

REFERENCES **70**

1. Introduction

The modern world is built on data, from which important knowledge may be gleaned. Every organization works with its data and seeks the most efficient and effective way of storing them since data are its most important asset. Thus, the tremendous storage of data has created data lakes with unstructured data, where its main format is textual data.

Tim Berners-Lee, the inventor of the World Wide Web, envisioned a world with data available to everyone. For this reason, he has suggested a five-star framework for defining the quality of Open Data [1]. At the first level,, data in any format should be accessible on the Web. When data is provided in a machine-readable format, such as an Excel file, its quality advances to level two. The next level requires the data to be in an open, non-proprietary format, such as XML. On level four the data should be published under open standards of W3C, like RDF. Finally, the top level is for the data to be linked with other Open Data.

Over the past few years, a sub-filled of open data the Open Government has surged great interest among researchers. The term open government is based on the notion that citizens should have seamless access to governments publication, which will enable efficient public supervision [2]. More specifically, because of the vital role that the regulatory framework plays in people's everyday lives and the operation of the public and commercial sectors, the major subject of interest is the translation of legal documents into Legal Open Data.

Most nations' legal data is not widely available or machine-readable. In most cases are in PDF format, which is the first level of the Tim-Berners-Lee hierarchy [3]. So, there is a need for a significant effort to be made to retrieve structured information from these papers. Of course, manually encoding data into structured format is a time-consuming process and required a tremendous amount of labor hours to be achieved, since also the past documents need to be transformed. But, using even semi-automatic systems for marking up legal documents may significantly cut down on the time and effort required.

There comes the field of Natural Language Processing (NLP). NLP scientific field has been substantial research in the last decade since it has been proven to be effective in several applications and its significance has been continuously growing [4]. It is heavily based both on the language and the kind of text to be processed. There might be some fundamentals, but the differences in the language (ie. Greek and English) and the structure of the documents (ie. biomedical and legislation documents) can affect steps in the document process [3].

There is a specific sub-field of NLP that focuses on automatically extracting structured important information from sources of textual data, which is known as Information Extraction (IE) [5]. The combination of NLP and consequently IE with Legislation has gained the attention of researchers during the last few years, especially for the Greek language. The most common tasks, researchers are trying to deal with to extract information to enrich the data quality, are the issues of Named Entity Recognition (NER) and Named Entity Linking (NEL). The first one is the task of extracting specific entities (ie. Person, Location) in the text, while the second one is to link those entities to a unique entity [5]. To complete these tasks several essential pre-processing procedures are required, such as stemming [6], which removes a word's suffixes and prefixes to obtain its lemma. In addition, several IE sub-tasks are required, like the Part-Of-Speech Tagging (POS-Tag) that identifies the words-context in a sentence and labels their proper tag [7] and is required information for NEL.

Of course, the researchers do not only focus on the transformation of semantic information extraction to public open data. Several works focus on specific sub-tasks of IE taking place on the way to solve Legal Open Data or even concentrate on different problems, such as Topic classification.

Heading back to open government and legal open data, Greece as a member of the Open Government Partnership should be dedicated to fostering open government data, especially in the legal field, as stated in the National Action Plan on Open Government 2016-2018. But still today, legal documents in Greece are distributed through several online sources in PDF format. The two main sources are the National Printing Office (ET), which is

the public agency responsible for the circulation of Greek legislation, and the Diavgeia Portal, where acts and decisions of public administration bodies are published.

In this research, we aim to extract structured information from Greek legislation documents. More specifically, first, we will perform the task of NER to extract the entities into the documents. Afterward, we will fine-tune a transformers-based model, to identify the relations among these entities, and finally we construct an extensive framework to combine these models and extract structured information from plain text input. The relations extracted could be represented in metalanguages, with the most common one called RDF (Resource Description Framework), where relations are represented as triples in the form of entity-relation-entity, known as a subject-predicate-object expression.

The rest of the research has the following structure: Chapter 2 establishes the context of IE tasks in terms of legislation documents, as well as presents useful machine learning concepts and the structure of ET. Chapter 3 outlines previous research on the endeavor to convert unstructured legislation data into publicly available data in the region of the EU that can be easily accessed by citizens, as well as several works that have been undergone concerning Greek legislation. Chapter 4 presents the procedure followed to extract legislation data from ET, as well as the methodology of how IE tasks will be performed in our dataset. In chapter 5 the implementation and results of the IE tasks are presented. Finally, in chapter 6 we summarize our study findings, identify the value of our work both from theoretical and business perspectives, and finally, we recognize some limitations, and suggest some potential directions for overcoming the limits.

2. Theoretical Background

2.1 Information Extraction

As already stated in the introductory part, the process of automatically extracting structured important information from sources of textual data is known as Information Extraction (IE) [5], comprising one of the most researched fields of Natural Language Processing (NLP), starting its routes back in the 1960s. The structured information extracted could be multiple different things like entities, events, relationships among the entities, and key phrases that describe a sentence. The data extracted can be immediately utilized for user display, kept in a database for integration with the rest system, or even used for Information Retrieval applications [8].

But what makes IE so special and deeply researched field? This has to do with the fact that it extracts information from text, which is considered unstructured data. Given also the vast amount of textual data created every data and the emergence of the field of Big Data it is easy to understand the importance of IE in discovering more effective and efficient ways to analyze these enormous volumes of textual data [9].

Before moving into the next part of describing the main tasks of IE, it would be helpful to distinguish the difference between two different fields that are often confused with IE: Full-Text Understanding and Information Retrieval (IR). The task of Full-Text Understanding necessitates a machine knowing the complete essences of the content, which might be challenging for humans as well, while IE is quite straightforward, as it simply necessitates comprehension of certain text fragments. In simple words, as [10] stated, *"Information Extraction is a more limited task than full-text understanding"*. Concerning IR the difference is more straightforward since IR is the process of locating relevant documents (typically text) among a collection of documents that correspond to the user's query [11] and, as stated previously, in many cases uses data extracted from IE to perform this task.

2.1.1 Information extraction Tasks

One of the basic tasks on IE is the identification of the main keywords or keyphrases that represent it. This task is called keyphrase extraction (KPE) [5]. Moving towards a more advanced task, called Named Entity Recognition (NER). NER's main purpose is to discover the mentions of an entity in the documents and label its type as well [12]. For instance, the label "Elon Musk" as a person. The task of NER will be further analyzed in part 2.3.

But, some entities could be assigned to more than one category. Let's take the example of the word "Apple". This could refer to the company or a fruit. By performing some appropriate sub-tasks, such as Part-Of-Speech tagging, analyzed in part 2.2, Coreference Resolution (part 2.5), and Dependency Parsing (part 2.4), the system should declare in which category it should be assigned. This task is called Named Entity Disambiguation (NED) [5].

Moving on we turn to the process of Relation Extraction (RE), which involves identifying and categorizing semantic relationships between text items [12]. These connections are frequently binary ones, such as child-of. The concept of RE will be further analyzed in part 2.6.

In our project, we will rely on the concept analyses till now. Beyond the aforementioned duties, there are a few additional complex IE tasks. One such is Event Extraction, which has its main purpose in finding the events in that entities take place [12]. Another related task is Temporal Information Extraction, which seeks to extract time and date events [5].

Every one of those duties necessitates varying degrees of language processing. Several rule-based approaches, as well as supervised, unsupervised, and semi-supervised machine learning, are used to tackle the problems. In most cases, a hybrid approach is required to end up with higher results. All the details will be specifically analyzed for each sub-task moving to the next parts.

2.1.2 General Pipeline of Information Extraction

Below a general pipeline is presented, which specifies some of the tasks mentioned above and is presented in detail during the following parts. Through this pipeline, one can have a first understanding of how the tasks are connected and which one of them is a prerequisite to performing some more advanced IE tasks. For instance, one can not extract relations without having first derived the entities in the text and performing the task of Entity Disambiguation. Having the background image, we are now ready to dive deeper into each IE task one by one.

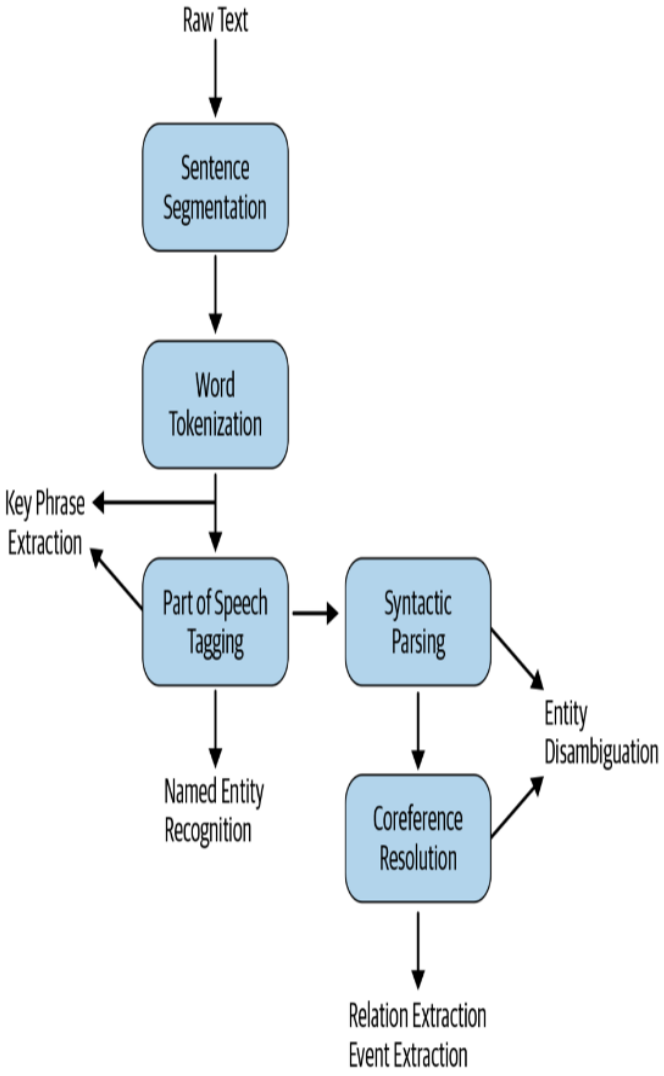

Figure 2.1: Information Extraction Pipeline [5]

2.2 Part Of Speech (POS) Tagging

Part-of-speech tagging (POS tagging) is an essential component of NLP [13]. Most NLP implementations, such as semantic and syntactic analysis, sentiment analysis, and machine translations rely heavily on a comprehensive POS tagging tool [14, 15]. But what exactly is POS tagging? It is the duty of associating each unique word with its proper portion of speech (ie. noun, verb, adverb, etc.). As illustrated in figure 2.2 a system has as an input the tokenized words of a sentence and produces as an output a sequence of tags for each of the input words. This is also a disambiguation task since each word can be assigned to more than one category [16].

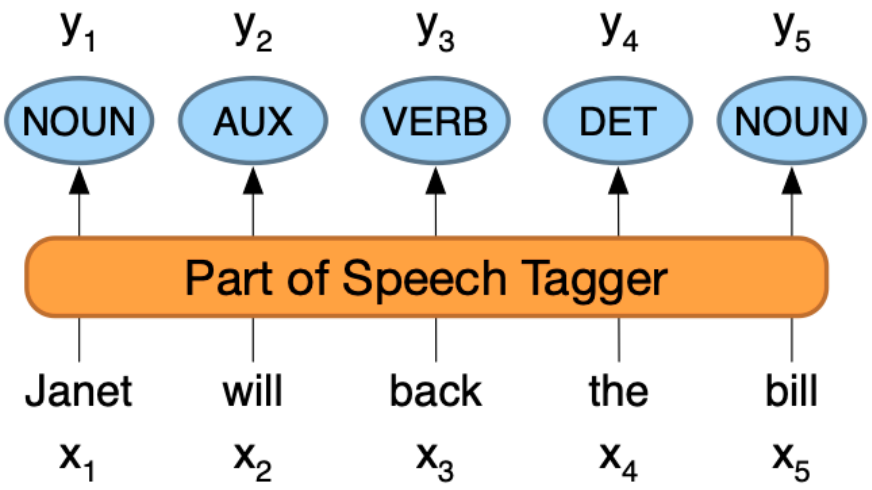

Figure 2.2: Part-of-Speech Tagging Example [12]

Even though POS tagging is a crucial component of NLP, there are NOT many studies that concentrate on the Greek language and only recently started gaining the researcher's attention. In such a try [16], created the first POS-tagged data set of Greeks' social text and also implemented a supervised POS tagger. As suggested by [17], since Greek is a highly inflective language there is also the need for morphological features to increase the efficiency

of the models. In a try to implement such an approach [18] trained a POS tagger that was able to detect the morphology of the tokens as well.

2.2.1 Approaches to POS Tagging

POS tagging is a sequence labeling task since it has to tag each input provided in the system. According to [19] this task can be treated either as a supervised or as an unsupervised task, which can subsequently be divided into three sub-categories; rule-based, stochastic tagging, and hybrid tagging [20].

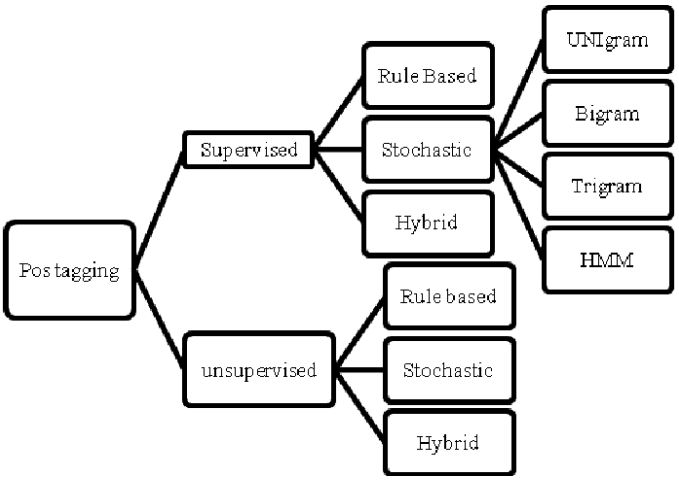

Figure 2.3: Part-of-Speech Tagging Techniques [19]

To train a model using supervised learning, a pre-tagged corpus is necessary, using some features as inputs, such as word embeddings. These models have achieved high accuracy in some cases (97.3 percent) [21]. Unlike supervised learning models, a pre-tagged corpus is not required for unsupervised POS tagging methods. They either compute the required information using stochastic taggers or infer the contextual rules needed through transformation or rule-based systems [22].

Now that we have a clear image of the approaches we use, we should further explore the sub-categories and techniques used to tackle these problems. Hand-written language rules are used in rule-based tagging to give the appropriate tags to the words. They are also known as contact frame rules [20]. This approach may end up in high precision results, but it is quite time-consuming and it requires a deep knowledge of the domain [19].

The stochastic-based approach takes advantage of frequency, probability, and statistics. Each simplest version finds the most commonly used tag for a term in the labeled documents and marks it in the unannotated text. A more advanced approach is the n-gram one, which determines the best tag taking also into account the n previous tags. Some of the most commonly stochastic models used are the Maximum Entropy Markov Model (MEMM), Hidden Markov Model (HMM), and Conditional Random Field (CRF). CRF model is the most commonly used [19], since it is bidirectional, meaning it takes into account the words coming after. The main disadvantage of the stochastic technique is that some sequences of tags may be contradictory to the grammar rules of a language.

Finally, in hybrid models, the two aforementioned techniques are combined. The words first take their tag based on a stochastic model and then linguistic rules are applied. This way the advantages of both techniques are combined [23].

2.3 Named Entity Recognition (NER)

The task of finding and categorizing all items of interest in a document is known as entity recognition. This is a two-step task. Firstly, we identify the spans of texts that refer to an entity and then we classify them to the type they refer to (ie. Person, Organisation, Location, etc.) [24].

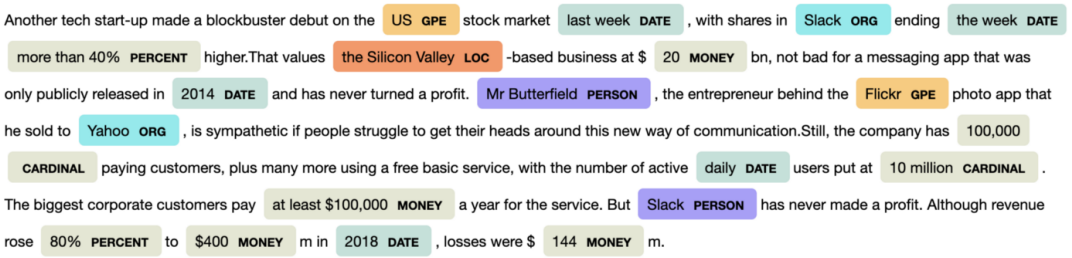

Figure 2.4: Named Entity Recognition Tagging Example [25]

In figure 2.4 a general example of the NER result is presented with some of the most common entity types used. Of course, the entity types may vary according to the domain. For instance, in a Biomedical case, the entities such as disease, drug, gene, and organism are used.

2.3.1 Approaches to NER

As of the POS tagging described in part 2.2, NER has usually been described as a sequence classification case, since the entity prediction is also dependent on the context [5] and thus the solutions are mainly treated through stochastic techniques. Additionally, these models rely on a slew of orthographic, morphological, and linguistic characteristics derived from the document. The syntactical structure of the text, certain word capitalization, and the usage of domain-specific n-grams are also extremely useful features to be fed into the models. But, in some domains, these features have not been proven enough, thus Gazetteer and rule-based heuristics have to be added to the process to improve the performance [24]. In general, hybrid models have been proven as the most effective approach [5].

To train a supervised model the data should be presented in a way that makes it possible to model the situation. The most commonly used one is the Inside - Outside (IO) scheme, which annotates the words that belong to an entity as "Inside" and all the others with "Outside", but there were some drawbacks with this scheme, so another approach was introduced called "BIO", where "B" is referred to the tag Before, which is assigned to the first word of the entity [26]. A concrete example of such an annotation is presented in Figure 2.5.

```
Essex   B-ORG
,       O
however O
,       O
look    O
certain O
to      O
regain  O
their   O
top     O
spot    O
after   O
Nasser  B-PER
Hussain I-PER
and     O
Peter   B-PER
Such    I-PER
gave    O
```

Figure 2.5: BIO Annotation Scheme [5]

Concerning Greek texts, several approaches have been tried over the years. Back in the ‘00s, [27] performed a solution using rule-based grammar and gazetteers. But recently the research is following the global trend of approaching a NER problem, which is not other than applying Deep Learning techniques, which allow end-to-end training and involve less feature engineering, having finally higher overall performance [28]. In such a case, [29] constructed a NER dataset focused on Greek, called eINER, and trained state-of-the-art deep learning NER models. Their findings were comparable to those of English researchers.

2.4 Dependency Parsing

The task of building a parse tree from a given text is known as parsing in computational linguistics. A parse tree draws attention to a sentence’s syntactical structure by grammar rules, highlighting this way the connections between words [12]. There are some techniques used on this type that might be conceptually closely related since the end goal is to derive the syntactic information of a sentence, but the results derived from each approach are different. The two main approaches are Constituency Parsing (CP) and Dependency Parsing (DP) and in some cases are used alternatively, but we should declare their main differences.

In the case of Constituency Parsing, which is also known as structure parsing, the sentence is broken down recursively into its parts using a phrase structure tree [30]. An example is presented in Figure 2.6, wherein the terminal node of the words of the sentence exists. On the parent node usually, the POS of each word is contained and the components of the sentence are represented by all other non-terminal nodes.

In the case of the Depency Parsing tree (Figure 2.7), the syntax is described based on word dependencies. So we are moving from the context-free grammar described previously to dependency grammar. Directed arcs from independent to dependent words serve as an illustration of the relationship between the words. Because the labels are chosen from a predetermined list of grammatical connections, this structure is called Typed Dependency Structure [31]. In the next part, we will focus on approaches used in DP.

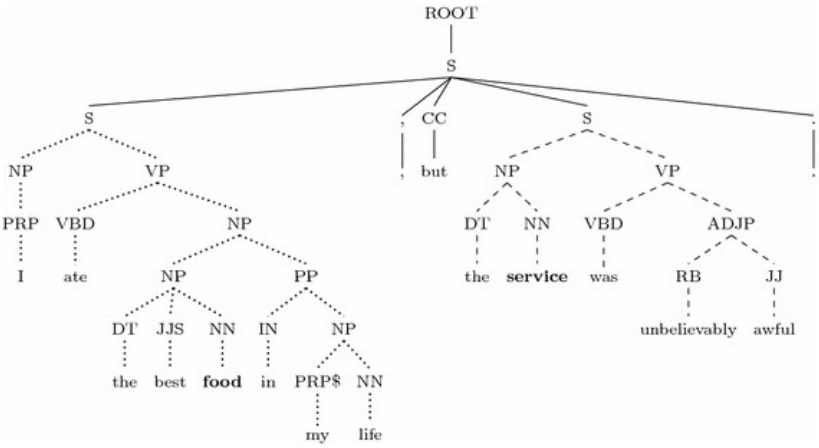

Figure 2.6: Constituency Parsing Tree [31]

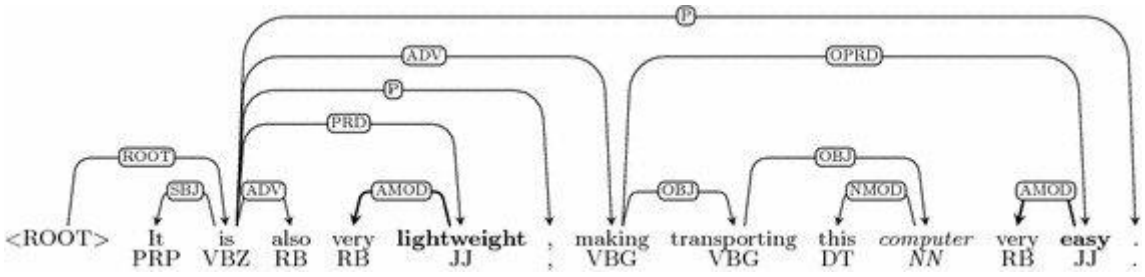

Figure 2.7: Dependency Parsing Tree [31]

2.4.1 Approaches to Dependency Parsing

There are two main ways for someone to approach the problem of DP. We could devote pages to describing each approach, but that is not the purpose of this thesis. At this point, we will try just to present an overview of how the two different approaches work in general.

The first and simplest one is called transition-based dependency parsing. The fundamental concept behind it is to anticipate a sequence of steps that will lead from an initial to a final configuration, creating a target dependency parsing tree [32]. The configuration is comprised of a stack, a collection of word relationships, and a buffer of words that are fed into the stack. In this structure, a transition action predictor is added, which is responsible for

determining the transition action depending on the present configuration. After the prediction of the transition action the configuration shifts based on the predicted action [12]. So, by searching the space of all possible configurations we try to end up in a final configuration, after taking into account all the words, to create a dependency tree. Until the appearance of deep learning methods the problem was solved using conventional machine learning methods which required manually crafted features. But recently, deep learning models diminished the process of feature engineering and yet achieved high-quality results [33].

The second approach is called graph-based dependency parsing. Its main concept lies in the idea that the parser starts by giving a score to all word pairs about the possibility of a valid relation dependency exists. Afterward, the model uses decoders, such as a maximum spanning tree, to construct a parse tree based on the assigned scores [34]. As it is easily understandable a key component of the whole structure is the score function. One of the main advantages of this approach compared to the transition-based one lies in the longer dependencies that it can capture [12]. The transition-based approach mainly achieves high results in shorter dependencies and the performance significantly declines when the dependency increases. Another significant advantage is that it can produce non-projective trees, which might be not a crucial fact for some languages, like English, but it is a problem for many.

But, what does the notion of projectivity means? If a route exists from the head to each term in the phrase that sits between the head and the dependent, then this arc is said to be projective. But, in languages with flexible grammatical structures, such as Greek, some legitimate constructors result in non-projective trees [12].

The lack of Dependency or Constituency Parsing for low-resource languages like Greek is even though it is currently a core and significant tool for several NLP tasks. Therefore, a greater study needs to be done on consistency parsing for Greek because it is a necessary "tool" for many NLP issues, such as Relation Extraction, which will be analyzed in part 2.6.

2.4.2 Evaluation of Dependency Parsing

Labeled and unlabeled attachment accuracy are the most used metrics for assessing dependency parsing. In the case of unlabeled attachment, it only requires the correct assignment of a word to its correct head and it does not take into account the dependency relationship. These metrics are named labeled attachment score (LAS). In the case of unlabeled attachment score (UAS), also the dependency relation is taken into account for the evaluation [12].

2.5 Coreference Resolution

The challenge of locating linguistic statements in natural language and their antecedents that refer to the same discourse entity (co-refer) is known as Coreference Resolution (CR). [35]. We define referring expression as any conversational remark or anaphoric phrase, whereas antecedent is utilized to characterize the entity to which the referring expression refers. The entity mentions that do not have an antecedent are characterized as singletons [36]. To better understand the task of CR a concrete example is presented in Figure 2.8.

Figure 2.8: Coreference Resolution Task [5]

Likewise, there is the concept of Anaphora Resolution, which sometimes is used interchangeably with the concept of CR. These two tasks are closely related, but there are some differences between them. The key distinction is that in the task of anaphora resolution we may aim to identify what is the antecedent of the term, but it can not point to a real-world entity. Thus, this task can only be solved using intra-linguistics, while extra-linguistics

features are also required in the case of CR [37]. As a result, singletons are not included in the scope of anaphora resolution.

2.5.1 Anaphoric Types

A wide range of anaphoric types is covered by the task of Coreference Resolution CR, but most of the algorithms are modeled to target just certain sorts of references. The datasets used are mainly targeted approaches to specific anaphoric types, so in this section, we will briefly try to represent the main anaphoric types that take place [37].

Zero Anaphora: This is one of the most common tasks in CR. A gap is used in a phrase to refer to its antecedent [38].

Pronominal Anaphora: One of the most widespread anaphoric types, which can be divided into four types, indefinite definite, and adjectival pronominal [37]. In the case of one the pronoun, "one" is used to point back to the entity. Indefinite type is the kind of anaphora referring to a generic entity like "Many". While in the case of definite the anaphora is made towards a worldwide unique entity, for instance, "the car". Finally, the adjectival pronominal anaphora has as a referring entity one that is described by an adjective, such as "kind stranger".

Demonstratives: In this anaphoric type a comparison takes with something that was presented earlier. In most cases, words used to refer to the antecedent are "this" and "that" [39].

Presuppositions: The anaphora is made towards ambiguous pronouns, such as "anyone".

Split Anaphora: In this case, anaphora refers to more than one antecedent [40].

Cataphora: This is the opposite of anaphora and it points to entities that are referred to afterward [41].

Bridging Anaphora: It is an anaphoric type that points back to a phrase that consequently points to an antecedent mentioned at an earlier point of the document [37].

Non-referential terms: Finally, a very important task is to find out the pronouns that do not point back to any antecedent. The pronoun “it” is one of the most commonly referred to terms.

2.5.2 Anaphoric Constraints

To find the suitable antecedent(s) for each anaphoric phrase, it is necessary to analyze certain syntactic and semantic characteristics that are designed to restrict the alternatives. The main constraints used in the models, mainly English ones, are presented below, based on the analysis of Sukthanker et al. [37].

Gender Agreement: The co-references must agree on gender.

Number Agreement: Two mentions can co-refer if and only if they agree on the singularity or plurality.

Person Agreement: The connection between two mentions can be made only in the case they agree on one of the three-person categories, which are the first, the second, and the third.

Verb Constraints: Some connections are not able to be done due to the verbs used in the sentence. In simple words, some candidate antecedents are eliminated.

Recency: Ones introduced lately have more significance than entities presented previously in the conversation.

Discourse structure: Structural characteristics can also limit the preference of one antecedent. Such a characteristic is considered parallelism.

World Knowledge: This is the most difficult constraint to involve in a system and to the best of our knowledge none of the systems achieved incorporated it completely so far. The difficulty lies in the fact that the terms refer to real-world entities which are not stable through time.

2.5.3 Approaches to Coreference Resolution

The Coreference Resolution CR models constructed through the years can be categorized into the following five distinct types: Mention-Pair models, Mention-Ranking, Entity-Based, Latent-Structure, and the most recent one Language-Modeling [42].

Mention-Pair models are the most simple ones tried to tackle the problem of CR. The model evaluates a pair of mentions at a time, taking into account the characteristics of each reference, and arriving at a binary result [43]. This approach is no longer not used due to the limited abilities it offers.

Mention-Ranking models were created to overcome the main disadvantage of mention-pair models which was the binary outcome it provides that could not inform the user of how good an antecedent is. This model gives a probabilistic value to each antecedent and simultaneously ranks them and makes a connection only with the highest-ranked antecedent [44]. This approach was further improved by introducing deep learning techniques in the models constructed.

An expansion of mention-ranking models is the Entity-Based models, which are capable of determining when entities should or should not be combined. This capability takes into account the transitivity which is important in CR and without it, pairing-mention clusters would include many mistakes. In the first implementations using deep learning techniques, the models built clusters instead of entities, shifting the task to cluster ranking [45].

Following the previous models Latent-Structure models have continued to map the entities into clusters, but in a tree-like structure [46]. Several deep learning techniques have been implemented and introduced significant improvements, like reinforcement learning, entity equalization, and adversarial training.

Finally, a great breakthrough has been made by incorporating Language-Modelling to enhance CR. By changing the models' main objective and introducing syntactic and linguistic features CR results were significantly improved [47].

2.5.4 Evaluation of Coreference Resolution

There are three widely used metrics to evaluate the performance of the CR task. The metrics are MUC, B-cubed, and CEAF. The process is to compare a set of clusters generated by the model to the gold clusters from a human annotation [12]. Their difference lies in how they calculate Precision and Recall.

The MUC metric is the first one ever used to evaluate CR and it is a link-based metric [48]. For precision, it counts the common links generated by model K(d) and from the human annotation S(d) divided by the number of links from the human annotation. While for the recall the numerator is divided by the number of clusters from the human labeling.

$$Precision = \frac{c(K(d), S(d))}{s(S(d))} \quad (2.1) \qquad Recall = \frac{c(K(d), S(d))}{k(K(d))} \quad (2.2)$$

B-cubed is a mention-based approach [49]. We compute the metrics for each reference separately and we then add them using a weighted sum to receive the overall metrics.

$$Precision = \frac{wc(Cji)}{wk(Ki)} \quad (2.3) \qquad Recall = \frac{wc(Cji)}{ws(Sj)} \quad (2.4)$$

Finally, the Constrained Entity Alightment F-measure (CEAF) is an entity-based metric, which seeks to discover a one-to-one mapping (g) among the links generated by the model K(d) and from the human annotation S(d), in addition to a similarity metric to gauge how similar the entities are [50].

$$Precision = \frac{\Phi(g*)}{\sum_{i=1}^{|K(d)|}(\varphi(Ki, Ki))} \quad (2.5) \qquad Precision = \frac{\Phi(g*)}{\sum_{i=1}^{|K(d)|}(\varphi(Ki, Ki))} \quad (2.6)$$

2.6 Relation Extraction

Assume we have extracted the document's entities through the task of NER described in part 2.3. The next step is to detect the relationships that exist among them [51], which task is called relation extraction (RE). For instance, we might find in a text relation of child-of (X is the child-of Y). Sometimes these relations are represented in metalanguages, with the most common one called RDF (Resource Description Framework). More specifically, a triple in RDF is a tuple of entity-relation-entity, known as a subject-predicate-object expression [12].

This is for sure the most advanced language processing task described so far since except for finding and disambiguating the entities, there is the need for a model to extract the relationships among them by taking into account the words connecting them. As one can understand this task is of particular importance not only to create rich knowledge bases, but also the relations can be used to create a question-answering model, index documents in search engines, and assist in the textual comparative analysis [52].

2.6.1 Approaches to Relation Extraction

Four approaches have been used through the years to tackle the task of RE. These approaches use or sometimes combine some of the following algorithms: handwritten heuristics, supervised, semi-supervised, and unsupervised machine learning [12]. The most commonly used approach is to treat the task as a supervised classification problem with two steps. The first one is to explore if two entities are related to each other (binary classification) and the second step is to explore the relationship between them (multiclass classification).

Now that we have a first picture of the approaches used, it is time to explore deeper each one of the algorithms. The earliest and still widely used is the pattern-based approach, using lexico-syntactic patterns, which was introduced by Hearst back in 1992 [53]. The main advantage of hand-crafted patterns is the high precision they achieve especially when they are focused on a single domain, but, on the other hand, there is the cost of a low recall.

The supervised machine learning approach is the straightforward one. There is a training corpus with hand-annotated relations between the entities and by using handcrafted and contextual features, as well as syntactic structure, a model is trained to be able to annotate a test set. In the last few years after the appearance of more advanced deep learning techniques such as the transformers, which will be further analyzed in part 2.7.3, the researchers try to take advantage of the transfer learning and use a pre-trained transformer encoder algorithm like BERT by adding a linear classifier on top of the model. In figure 2.9 such a model is presented, which was constructed by Joshi et al. in 2017 [54].

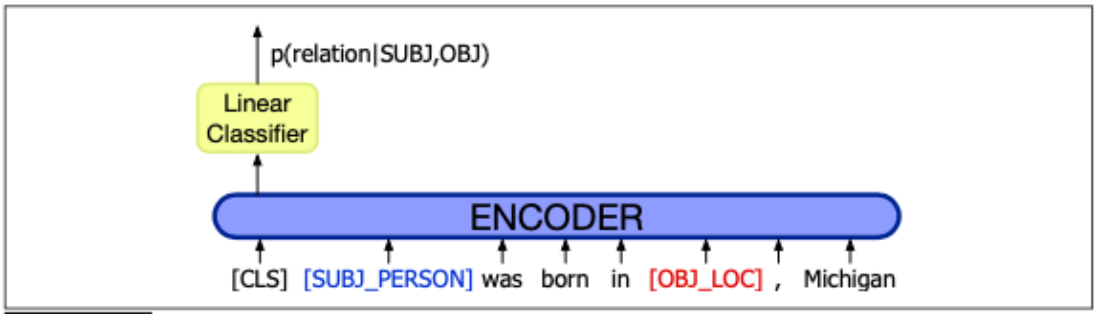

Figure 2.9: Coreference Resolution Task [54]

The problem with the supervised machine learning approach is that it requires labeled data, which is a costly procedure. There comes the semi-supervised approach which takes advantage of two techniques, Bootstrapping and Distant Learning [12]. Both concepts are closely related but there are few distinctions. In the case of Bootstrapping, supposing we have few relations (either hand-crafted or hand-annotated), the algorithm proceeds by searching for words (on the web or in the dataset we are using) that contain both of the items in the relation, returning this way new relation tuples [55].

Distant supervision is an extension of bootstrapping, since not only uses some hand-crafted seeds but also uses patterns from large databases to extract more relational examples. These examples might create large noisy pattern features but after combining them with a supervised classifier a powerful model is created [55]. To better understand the concept of distant learning an example will be provided. Let's assume we could like to extract the relationship born-year. We might have only a few hand-annotated examples, but we could take advantage

of DBpedia, which might provide thousands of such relation examples [5]. Worth noting is that this technique can take place if and only if there is such a large database. Snorkel is an application that can be used of taking advantage of the distant learning technique.

Unsupervised algorithms are an option if we are unable to get training data for supervised approaches. Unsupervised RE, commonly referred to as "open IE," seeks to identify relationships on the web without using any training data or relational database. The extracted relations have the form of tuples with the elements verb, argument1, and argument2 [56]. A verb may occasionally have extra arguments. With unsupervised algorithms, many relationships can be extracted, but the challenge is to map the extracted relationships in a standardized way.

For low-resource languages like Greek, there is a lack of corpora to perform RE. In such a try Christou and Tsoumakas [52] have created a dataset of Greek literary fiction from the 19th century and constructed a model that recognizes six relations. In an alternative implementation, some researchers preferred to translate the text from Greek to English perform triple extraction on the English text using pre-trained models, and afterward translate it back into Greek [57].

2.6.2 Evaluation of RE

In the scenario of supervised relation extraction, using accuracy, recall, and F-measure makes things straightforward. But in the case of unsupervised and even the case of semi-supervised things are not so simple. The approach used in this case is to get a sample of relations extracted and let a human check the accuracy of these results. In this way, we do not take into account how many relations should be extracted, but we only check if the extracted relations were correct [12]. So, the evaluation is just an approximate precision and nothing more.

2.7 Useful Machine Learning Concepts

In this section, we will present some machine learning concepts that are widely used in the IE tasks described before. These concepts will help us further understand the methodology we will follow in dealing with the task we have.

2.7.1 The Transformer Architecture

The Transformer architecture lets models analyze content in a bidirectional way, meaning that the text is processed from start to finish and vise-versa. This is the core breakthrough compared to previous state-of-art models, which were capable only of the one-way process of the text (from start to finish) [58]. The core of the system is attention combined with linear projections. Attention works not that different from humans on how we process information. We pay attention only to things that matter. In the same way, machine learning models construct differential weights indicating the words in a phrase that needs to be processed further [12].

As shown in figure 2.10, the architecture is constructed utilizing stacked Transformer blocks called encoders [58]. If there is a need a decoder is also added for predicting the target output, but in some cases like BERT, decoders are not utilized. Worth noting is that no recurrent mechanisms are used, making the model faster in training, since it enables parallelization, and is also less vulnerable to the vanishing gradient problem.

Due to their modular nature, transformers have several applications in NLP, the most significant of which is Language Modeling, which serves as the foundation for current NLP applications. Encoder-Decoder models, in addition to Language Modeling, have been demonstrated to be quite helpful for Machine Translation applications [59]. Similarly, by adding task-specific layers after the LM layers, Pre-Trained LMs may be utilized for different classification tasks [60].

Despite the wide usage of Transformers they still suffer from disadvantages with the main one being the quadratic computational complexity. Thus, several approaches are researched to overcome the drawbacks.

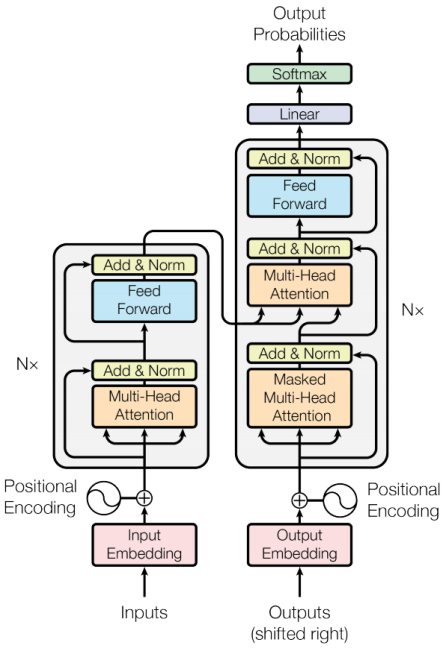

Figure 2.10: The Transformer Architecture [58]

2.7.2 The concept of Weak Supervision Learning

As already stated in previous parts one of the key drawbacks of less popular languages, like Greek, is the shortage of annotated datasets since it is a highly costly procedure. Thus some techniques, like the distant learning presented in part 2.6.1 are applied. Distant learning is an extension of Weak Supervision and in some cases is used interchangeably. In other words, it is an approach that leverages high-level and often noisy resources to create much larger training sets far more quickly than manual supervision could [61]. In some cases, weak supervision does not combine external sources but the labels are generated by using some noisy heuristics from the corpora that we already have [62]. Summing up the use of several weak supervision sources not only provides a scalable framework for constructing huge labeled datasets but may also be considered as a vehicle for incorporating high-level, conceptual input into the data labeling process.

2.7.3 The concept of Active Learning

Active learning is an alternative way to face the problem of obtaining labeled data. It is a framework that helps the system to select by itself which data the user should annotate, after learning from a small initial dataset [63]. To make it more clear, in active learning, a model after being trained in a limited collection of data uses an acquisition function to determine which documents should the user label. This decision is frequently dependent on the model's uncertainty. The documents or points are selected from the pool of unlabelled data we have. After the user annotates these points, these are added to the training dataset and the process is repeated.

Such approaches have the benefit of drastically reducing the amount of labeling necessary to train a model, saving consequently cost and time. Of course, this does not come with a cost, since there is a lack of scalability to high-dimensional data. So, one should use models that can learn from a small amount of data, restricting this way the models to be used [64]. This is a problem that researchers are trying to confront to combine active learning with deep learning models.

Worth noting is the existence of an application, called Prodigy, which takes advantage of an active learning framework and provides users with an interactive interface for data annotation.

2.8 Background on National Printing House (ET)

The National Printing Office (ET) is the public agency responsible for the circulation of Greek legislation and is under the Ministry of Administrative Reform and E-Governance. It is a technical production unit whose competence consists in the publication and circulation of the Gazette of the Government of the Hellenic Republic. The "Government Gazette" (F.E.K) is published following Law 3469 (issue A' 131/ 28.6.2006) "National Printing Office, Government Gazette and other provisions".

The F.E.K are published and circulated simultaneously in two formats, printed and electronic. The electronic circulation is carried out with the registration of the F.E.K. on the

website of the ET. The electronic form of the F.E.K. is in portable document format (pdf). Worth noting is that since 2006, the National Printing Office has a recognized digital certificate and digitally signs the electronic form of F.E.K automatically upon their production.

2.8.1 Structure of National Printing House

The publications are categorized into 12 distinct types. For simplicity reasons, we will only mention the most relevant categories and the key topics in each one of these.

In CATEGORY ONE (A) are published:

A. The Constitution and the legislation enacted and passed in its aftermath, as well as the country's international conventions or agreements with other governments or organizations, and any announcements relating to these conventions or agreements.
B. Presidential decrees with regulatory substance and presidential decrees issued in accordance with paragraphs 37, 38, 40, 41, and 43 par. 1 of the Constitution, as well as four additional sub-topics.

In CATEGORY TWO (B) are published:

A. The decrees approved the establishment of public benefit institutions and amended their organizations.
B. The regulatory decisions of the Prime Minister, ministers, deputy ministers, and other bodies of the Administration, as long as the present legislation does not allow for public disclosure by any other means and this law does not provide differently.
C. Ship registration decisions, and four more sub-topics.

Employees of Special Positions and Administrative Bodies of Public and Broader Public Sector Bodies (Y.O.D.D.) are published in the CATEGORY:

A. Appointments, acceptances of resignations, and removals of the President's Secretary General, the Parliament's Secretary General, the Government's Secretary General, the

general secretaries of ministries, the general secretaries who head the general secretaries of ministries, and the special secretaries who head unified administrative sectors of ministries.

B. Decrees and acts for appointing, renewing, accepting the resignation, or terminating members of autonomous administrative agencies.
C. The summaries of the decisions appointing presidents, directors, and members appointed under the first Chapter of Law 3429/2005 (F.E.K 314 A) and four more subtopics of Law 3429/2005.

As already stated, on the portal of ET there is the possibility of finding a F.E.K in a pdf format, and it is also possible to display only the first page of the pdf (summary). To do so, the user has first chosen one category (like A, B, etc.) and then the results can be sorted by issue date, publication date, number of F.E.K, and also the number of pages.

There is also the option of advanced search by keyword, after selecting the related category. The results returned presently the number of F.E.K, a provider of the text where the keyword exists, as well as details of the F.E.K.

It is important to note that there are additional types of searches, which are not of interest in this particular work.

3. Related Work

A primary priority for many governments throughout the world has been to modernize how individuals access legislative data. This chapter will outline previous research on the endeavor to convert unstructured legislation data into publicly available data in the region of the EU that can be easily accessed by citizens. In some cases, the research has gone one step ahead, by trying also to extract valuable knowledge and metadata from the documents, to assist in the transparency of the public sector. In addition, we will present several works that have undergone Greek legislation, a subject of particular research interest during the last five years.

3.1 Open Data Initiatives in European Union

The amount of size of rules and regulations published in the region of the European Union is growing. This is not only affecting the way citizens can find and process the data, but also the legislative and executive entities of governments since the environment is becoming more complex. Here comes, Information and Communications Technology (ICT), which can help both people and public institutions in handling this increasing amount of data. To do so, the legislative data are required to be transformed into a structured format and afterward create systems where this information will be kept including some enriched metadata.

In such a trial, the OpenLaws EU project [65] has as a target to collect legal materials from the EU members and link them together, providing this way an innovative service via an open platform. EUcases [66], which used Akoma Ntoso, a standardized legal XML vocabulary to represent legislative documents in a structured way, as the format for expressing law and case law, is another closely related initiative. The project's goal was to turn multilingual Legislative Open Data into Linked Open Data after performing structural and semantic analysis.

Another standardized legal XML syntax is the MetaLex, which was developed by the CEN workshop, and thus in most cases is met as CEN MetaLex. Researchers took advantage

of this standardized way of representation and published the Dutch national regulations into MetaLex Document Server (MDS), after converting texts from legacy XML into CEN Metalex, in a try to enhance access to legal sources and transparency [67]. The MDS project not only defines the conversion mechanism stated above, but also the documents converted to RDF and Pajek network files. It also reveals data over HTTP protocol and provides a SPARQL endpoint for advanced queries into the RDF database. The project was the first step to opening the Dutch regulation for semantic annotation as well as advanced analysis.

The European Council's most recent framework announcement was the European Law Identifier(ELI) [68], a new standard framework that should be applied by the national legal publishing systems to synchronize and connect national law with European legislation. ELI proposes a URI pattern as a starting point for locating legal materials online and provides an OWL ontology for characterizing the metadata of legal documents and proceedings.

Of course, ELI, as well as the other frameworks analyzed before, are not an end-to-end solution for all the member nations and they should be expanded to account for the unique characteristics of many countries' legal systems. They have been the keystones for further development. For instance, the MetaLex was further extended under the EU project Estrella, which explains the structure, substance, and events of the legislative process via modeling legal and legislative ideas using an ontology. This project is named Legal Knowledge Interchange Format (LKIF) [69]. Also, a related effort for conceptualizing case law is the European Case Law Identifier (ECLI), a sibling project of ELI, which was introduced not too long ago [70].

Lastly, worth mentioning is the CELLAR project, created by the Publications Office of the EU with the objective of storing official EU institution publications and bibliographic resources. This information is publicly available through the EUR-lex service [71].

3.2 Information extraction of Greek Legal Documents

The subject of IE from Greek legal documents has gained a lot of attention during the past few years, as already mentioned. One of the key works, which is based on the concepts

analyzed previously, is the OWL ontology, called Nomothesia, developed by a group of Greek researchers. Through Nomothesia, users can browse, search and query Greek legislation [72]. The OWL ontology extends the capabilities of MetaLex and ELI and it also links data with DBpedia and other Greek geospatial datasets. To do so, NER is performed to extract persons, organizations, geo-political entities, etc., to be linked with the external sources and provide the user the ability to perform complex SPARQL queries, such as how many legislations were signed by the Ministry of Education.

Before this work, a significant semantic approach was performed by taking into account the Akoma Ntoso XML schema. In their first step, they identified the structure, as well as the type of document [73]. After retrieving this information, the researchers broke down the legal blocks to discover the required metadata to develop a model for each document with associated URI markups. To do so, they used context-free language, which nevertheless was domain specific.

Another research team has proposed a solution for the consolidation of Greek legislation [74]. To find and update new legislation, regular expressions and pattern matching are used. In general, they state that in legislation documents a high structure language is in use, thus the NLP tasks would be beneficial, but there are key drawbacks, such as the lack of API to access official laws, the fact that in the best case the documents are in PDF form and thus many difficulties exist like the encoding of Greek characters.

A few years later, in 2018, the same research team got funded for a project entitled "Automated Analysis and Processing of Legal Texts for their Transformation into Legal Open Data", a try to transform unstructured legislation documents into Legal Open Data, by taking advantage of the Akoma Ntoso standard. In figure 3.1 the seven-step methodology that researchers follow is presented. Several publications were made during the months of implementation to present the progress of their work [75, 76, 77].

Of course, the works do not only focus on semantic information extraction. Several works deal with sub-tasks of Legal Open Data or even concentrate on a different task. For

instance, in the first case, the project of Nomothesia described above, included the linking of entities from legislation documents to external sources. To achieve this, the research team had to deal with two tasks NER and NEL presented in the Theoretical Background part. For the NER part, around 250 datasets were trained with word embeddings, and they tested several Long-Short-Term-Memory (LSTM) DNN. In the case of NEL, they converted the documents to RDF and connected them using heuristics [78].

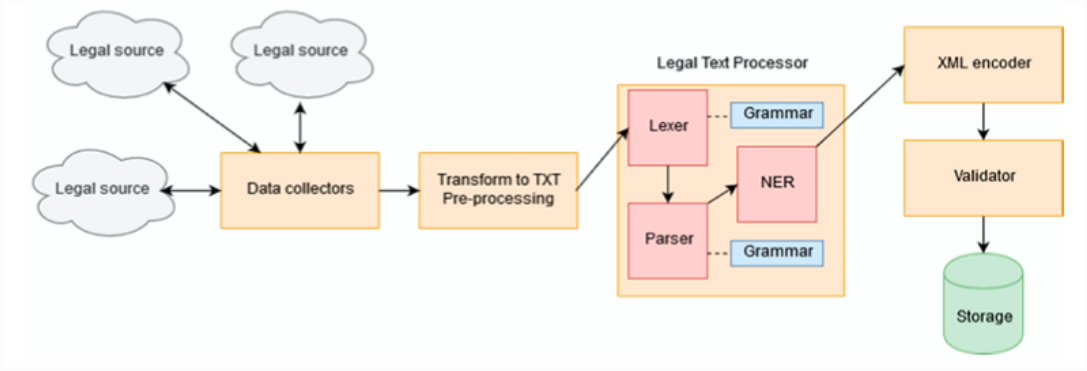

Figure 3.1: Implementation Framework [75]

An example of the second case where researchers try to deal with different research problems, members of the research team of Nomothesia tries to tackle the problem of Contract Element Extraction [79]. They perform several techniques combining manually constructed features, word embeddings, and POS tag embeddings, trained with linear classifiers, concluding that the most effective approach is the hybrid method. They also try to achieve higher results by using DNN models, such as the biLSTM-CRF combination [80].

Worth mentioning are some works of this research group on the legal domain, which might be not in Greek legislation, but have a high impact. Multi-Label Text Classification, which is an extension of Topic Classification, is the subject of one such project, where they create an addition to the EURLEX database that contains data on annotated EU law [81]. The most influential though is the first implementation of BERT in the legal domain [82], named LEGAL-BERT, which achieved cutting-edge yield in three tasks. They also released a three-times smaller version LEGAL-BERT-SMALL.

Finally, we will present some works deserving of attention. One example is the Greek government's efforts to improve access to the law and produce unified, re-usable legal materials. In their solution they include among the others ICT tools for semantic text analysis, the use of ELI, Akoma Ntoso, etc., targeting the following solutions: a) making content available in formats that are both machine-and human-readable; b) allowing for the visualization and access of legal documents over time; c) connecting Greek law with that of the European Union and the other Member States, and finally d) using a thematic legal index to enhance the efficiency of legal searches across multiple languages [83].

Another one is the work of Loutsaris et al. [84]. In their work, after performing a literature review of the techniques and infrastructures used in legislation mining, they suggest a framework for an open and automated legal system that can provide legislation data for any EU member state, based on the current ontologies. In the same spirit, Stylianou et al. [85] have proposed a framework to convert legislation documents from the DIAVGEIA portal to knowledge graphs, after performing information extraction tasks, like NER, CR, DP, and RE and extracting triples to exploit domain-specific ontology standards.

Last but not least, we should state that NLP tasks in Greek legislation have also gained research interest for undergraduate and graduate theses. Students at the National and Kapodistrian University of Athens have created a Greek Version of Legal-Bert and performed several NLP tasks [86, 87].

Summing up, we can see that NLP tasks on Greek legislation have gained great attention during the last few years. Researchers work on them either as a sub-task required for a more general concept of Linked Open Data or focus on specific problems like NER and NEL.

4. Basic Principles and Implementation Framework

4.1 Methodology and Implementation Framework

The primary goal of the research is to extract structured information from legal papers, especially the layout of public organizations as defined on a public authority document. As noted in section 2.1 this is an information extraction task since we need to discover the groups within the text and determine whether or not there is a link between them.

More precisely, our design is influenced by the IE pipeline depicted in Figure 2.1, but it is not entirely consistent with it. The distinction is that the models we trained for named entity recognition and relation extraction did not take into consideration variables such as part of speech and word dependence.

So far, the majority of solutions developed have been based on this feature engineering methodology. However, thanks to the transformers' architecture we were now able to obtain and train cutting-edge pre-trained models using the Hugging Face package. At this point, it is crucial to reiterate that there are four main approaches to address the problem of relation extraction and information extraction in general, which are the following: handwritten heuristics, supervised, semi-supervised, and unsupervised machine learning techniques. We approach the challenges as supervised classification issues in our study.

Summing up, we take advantage of the Greek version of the Legal-Bert language model constructed by students at the National and Kapodistrian University of Athens and we focus on our two tasks. Regarding named entity recognition, we annotate some data, create a dataset in a BIO format, and treat the task as a binary token classification problem, in which the model determines whether a word or a sequence of words is an organization since the model is bidirectional and has this capability, is an organization. Section 5.1 contains further information on the work. In the example of relation extraction, we produce a dataset indicating the relationships and non-relationships between organizations and trained the model as a binary text classification issue. Section 5.2 has further information about the work.

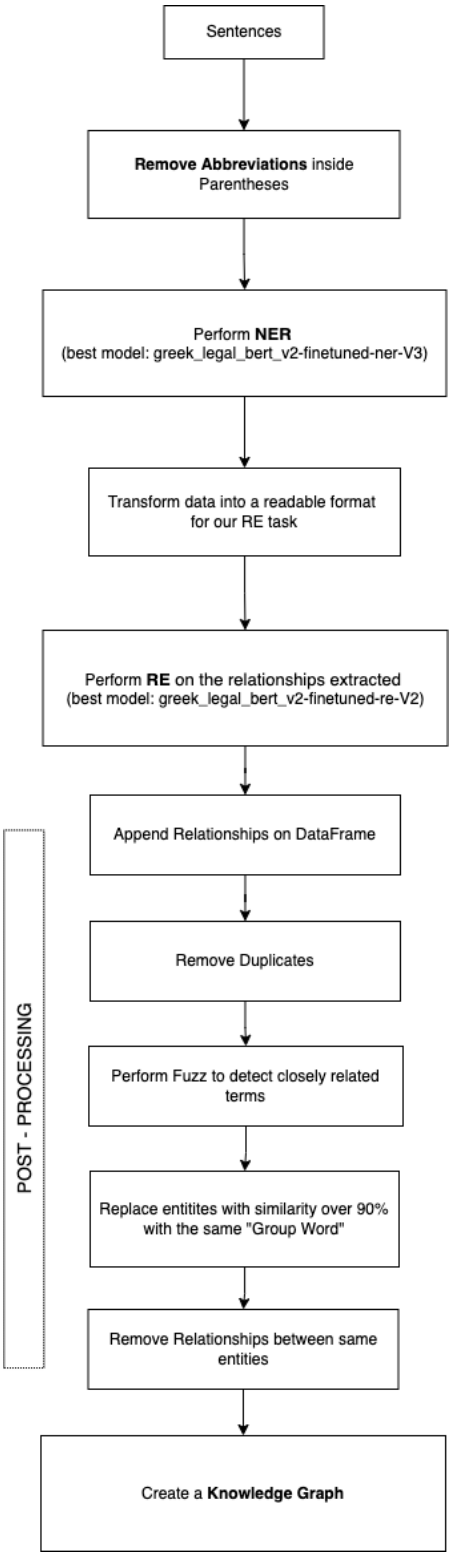

Figure 4.1: Implementation Framework

After we figure out how to recognize organizations in a text and determine whether or not there is a link between them, we have to merge them all into a pipeline. The fundamental objective is to create a program that receives text inputs and produces a knowledge graph of the organizations' relationships. The following are the primary actions that should be followed to produce this graph:

a) The incoming data is preprocessed by eliminating abbreviations into parentheses.
b) Named Entity Recognition is carried out.
c) The data is converted into a format our Relation Extraction model can comprehend.
d) Relation Extraction is performed.
e) A post-process is performed, where we append data into a data frame, remove duplicate relationships, and perform a similarity measure to check whether two entities are similar to a 90% degree and keep a common name for these entities.
f) To ensure that there is not a relationship between the same entity, a check is performed and in the case, a relationship exists, the information is removed.
g) Finally, a knowledge graph is created.

This procedure is seen in Figure 4.1 and will be detailed in full in Section 5.3.

4.2 Datasets

Two different datasets are created to complete the training of our models. Starting with the task of named entity recognition, we pick and retrieve only 17 files that are particularly referred to the structure of organizations, from a dataset generated by the DORG-IHU team. This dataset contains many legal documents extracted from the Kodiko website and converted to CSV file format. We convert the CSV file to a simple TXT format after retaining only the text and removing unnecessary information to submit it to the Prodigy application, our annotation tool. The data to be annotated vary from phrases to sentences and large paragraphs.

The annotation interface and a relative example are shown in figure 4.2, which appears in the browser as a local host page. Prodigy provides multiple workflows for annotation. We

proceed with the manual annotation using the model's recommendations (the "el core news sm"). It provides us with some valuable, yet incomplete suggestions.

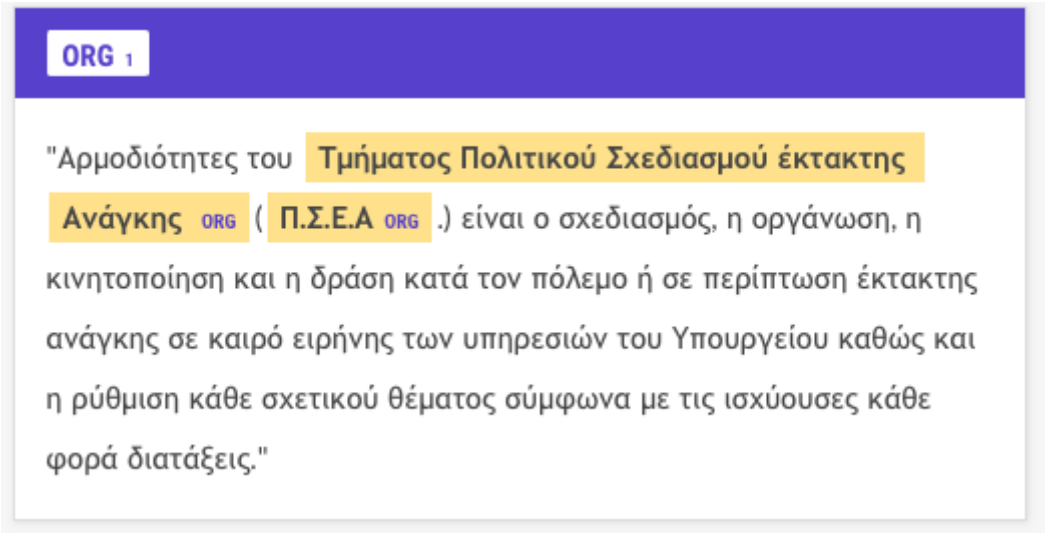

Figure 4.2: Annotation of Entities on Prodigy

After we finish with the annotation of 10 documents, we export the results into a JSON format, to be used for fine-tuning our NER model. Section 5.1.1 details the next stages.

The situation is a little bit more difficult when it comes to the dataset that has to be produced to fine-tune the RE model. We subtract one of the 17 relevant legal documents and we manually perform the subsequent actions described in this paragraph. First, we use Excel to copy and paste the document and delete extraneous information like “αρμοδιότητες”. We break up lengthy paragraphs into several smaller ones while maintaining the first row as it includes the primary organizational unit. Moving forward, in the circumstances where the major organization is specified too broadly, such as “Διεύθυνση” or “Φορέας”, we manually update the information.

After we have completed this process for 8 documents, we gather and upload them into a TXT format in Prodigy. We begin with the NER annotation, as before, but this time we use suggestions from the patterns we created before to help us with the annotation. One

significant distinction we utilize is that when there is an abbreviation inside parentheses adjacent to or within our organization, we highlight all the phrases as a single organization.

After we conclude the NER annotation, we use the results to annotate dependencies and relations. Except for the "SubOrg" relation, which indicates that one organization is a sub-organization of another and so has a relationship with it (figure 4.3 depicts a relative case), we also utilize the "CoRef" connection when one labeled entity is identical to another.

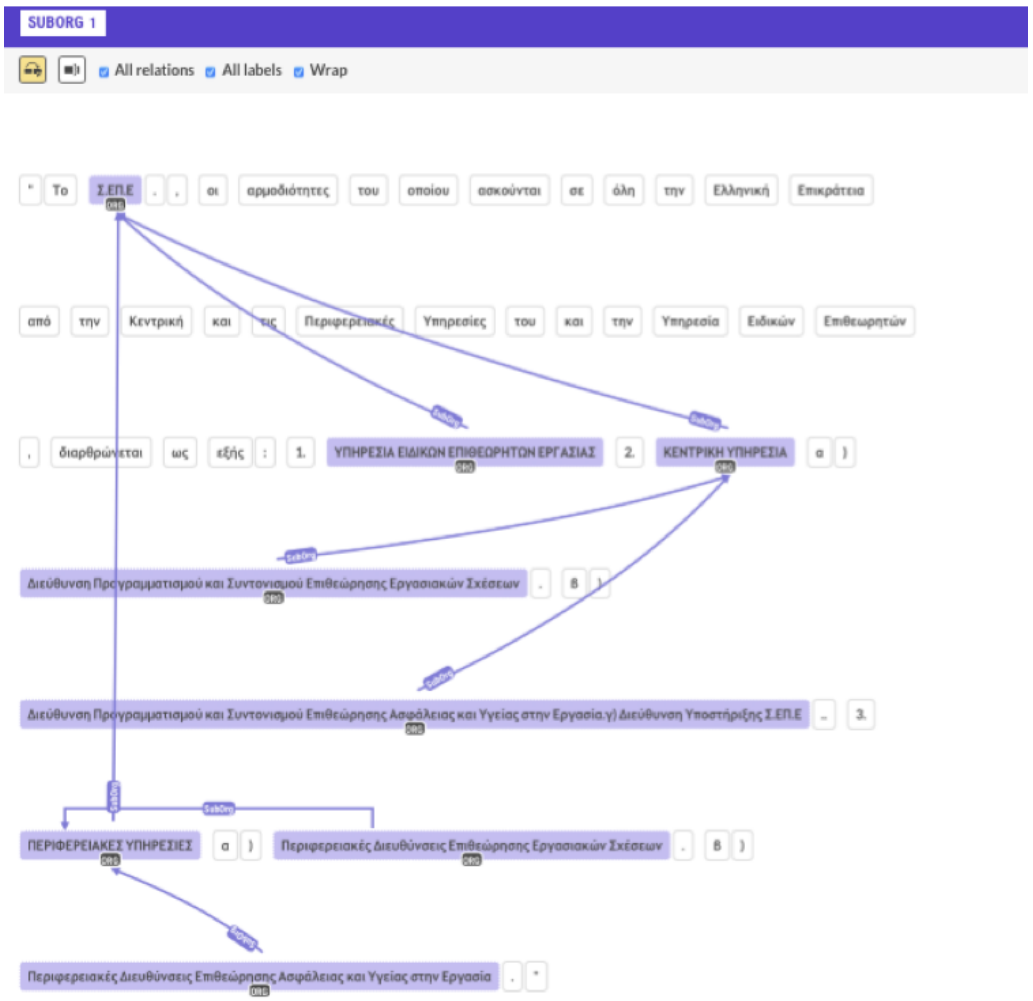

Figure 4.3: Annotation of Relations on Prodigy

After completing the annotation, we export the file from Prodigy in JSONL format. Then, we execute a script that removes superfluous information from the produced document and

renames the key values to make them more meaningful. Three key values are kept. The first one is called “sentText” and includes the paragraph under consideration. The second one is “entityMentions”, which contains all the organizations from this paragraph, and the last one is called “relationMention” and includes the entity pairs along with their kind of relationship.

After that, we proceed to the most essential step: we build the “NoRel” relation, for all entities that do not have a linkage, append these pairs to the “relationMention” section, and export the file in JSON format.

Finally, the file is separated into train-valid-test files and their precise worth of information is presented in section 5.2.1, along with the following methods for developing a model that can determine whether or not there is a link between the entities.

5. Implementation and Results

In this section, we will go through the approaches and tactics we use to achieve our final goal of extracting organizational structure from the text. The problem will be broken into three major sections. In particular, in the first section, we will focus on the approaches we utilize to extract the organizations from the text. Following that, we will discuss the methods used to assess whether or not there is a relationship between them, and finally, we will illustrate the entire pipeline, but in greater depth, as seen in figure 4.1.

5.1 Named Entity Recognition

5.1.1 Main process

As described in section 4.2, we have annotated around 10 FEK in prodigy and generate a dataset of approximately 15.000 inputs. These inputs might be phrases, sentences, or even entire paragraphs. Some inputs may have no organizations, while others may contain several. The first step, as described in section 2.3.1, is to convert the inputs into a BIO format. Our input phrase might look like this:

```
['"', '2.', 'Το', 'Υπουργείο', 'Ανάπτυξης', 'και', 'Επενδύσεων',
'μεριμνά', ',', 'στο', 'πλαίσιο', 'της', 'αποστολής', 'του', ',',
'κυρίως', ',', 'για', 'την', 'εγχώρια', 'παραγωγή', 'και', 'διάθεση',
'διεθνώς', 'εμπορεύσιμων', 'προϊόντων', ',', 'μέσα', 'από', 'την',
'εφαρμογή', 'της', 'κυβερνητικής', 'πολιτικής', 'για', ':']
```

```
['O', 'O', 'O', 'B-ORG', 'I-ORG', 'I-ORG', 'I-ORG', 'O', 'O', 'O',
'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'].
```

At this stage, we only retain inputs with a maximum length of 400 words. There are just a few inputs that surpass this level, and they are thus discarded. The reason for this filtering is the 512-word barrier that most language models have.

Moving on, we partition the dataset into 80% train and 20% validation and then encode them with padding such that all of the inputs are in a machine-readable format and of the same length.

We can now fine-tune our model so that our data is in the appropriate format. The language model used is Greek-Legal-Bert-V2, which is publicly available on Hugging Face. The training arguments are depicted in Figure 5.1. The total number of training steps to be taken is 7.490, and the assessment training duration takes approximately 1 hour and 20 minutes. The model achieves an F1 score of 82,19%.

```
model_name = model_checkpoint.split("/")[-1]
args = TrainingArguments(
    f"{model_name}-finetuned-{task}",
    evaluation_strategy = "steps",
    learning_rate=2e-5,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    eval_steps=250,
    num_train_epochs=10,
    weight_decay=0.01,
    push_to_hub=True,
    metric_for_best_model='eval_f1',
    load_best_model_at_end = True,)
```

Figure 5.1: Training Arguments

The model is then uploaded to the Hugging Face hub, which also includes a live testing interface. In one of them, we use a term that, as shown in figure 5.2, identifies the structure. It is worth noting that the acronym in parenthesis denotes a separate organization.

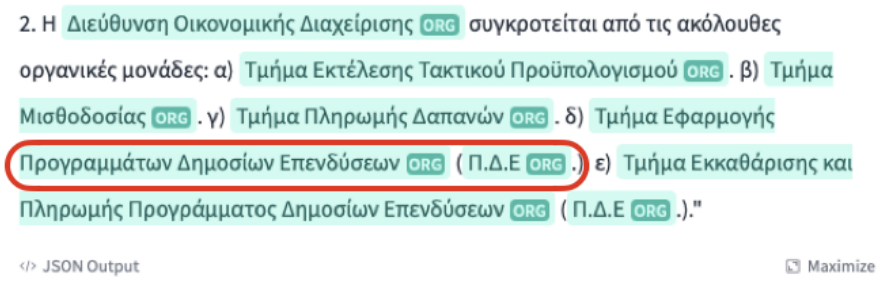

Figure 5.2: Hugging Face Live Test

Finally, Figure 5.3 displays the whole procedure discussed in this section. We will explore enhancements in sub-parts of the process in the next subsections of the NER task, but the overall structure will remain unaltered.

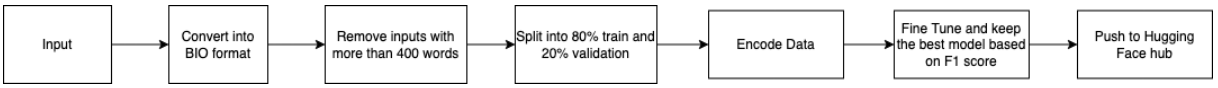

Figure 5.3: Implementation Framework

5.1.2 Enhance the Results

The model appeared to perform well for our objective, however, the metrics appeared to be poor and could benefit from some improvements. To increase the efficiency of the model we eliminate all inputs that do not contain an organization and all inputs that have more than 50 tokens but only contain one organization. This way the dataset decreases from 15.000 to 7.750 inputs. The F1 score rises to 85,3% when the model is trained under the same conditions. Less than 25 minutes were needed to finish the training face with 3.880 steps.

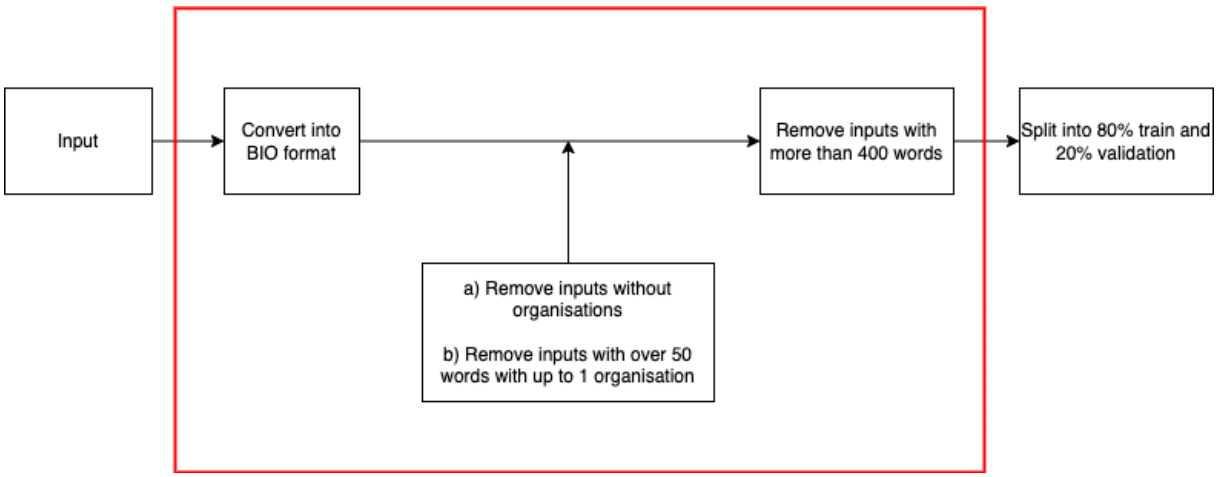

Figure 5.4: Enhanced Region in our Process

5.1.3 Compare the Language Models

To ensure that we have made the best selection, we should compare our chosen language model to some of our other possibilities. Greek-Bert, a global Greek language model, is one option. The second is a multilingual language model, which incorporates Greek among other languages. We anticipate that both of them will perform worse than the domain-specific Greek-Legal-Bert, which is trained particularly for Greek legislation, and that Greek Bert will

outperform Roberta since it is trained for the Greek language. Our beliefs are validated after training the models under identical conditions, and Greek-Bert has a little superior performance with an F1 score of 85,3%, whereas Greek-Bert comes in second with 85%.

	Greek Legal Bert	Greek Bert	Roberta (Multilingual)
F1	**85,3%**	85%	84,7%
Precision	84,2%	84,4%	83%
Recall	86,4%	85,6%	86,6%

Table 5.1: Language Model Comparison

5.1.4 Fine-Tune the Fine-Tuned Model

Our model treats the abbreviations inside parentheses as independent organizations, as seen in figure 5.2. This may not appear to be a problem at first, but as we go and attempt to compare all of the organizations to determine if there is a link between them, the acronyms will be taken into consideration and we risk making mistakes. Of course, removing acronyms before starting the NER work, which will be discussed in section 5.3.1, is a significant improvement. But we should also make adjustments to our model just in case getting rid of the acronyms does not work.

To achieve this, we will refine the model that is currently the best-tuned version we have created thus far. The data we'll be using comes from the second dataset mentioned in section 4.2 and was prepared specifically for the relation extraction job. However, as previously noted, before annotating the links between the organizations, we had to perform an annotation for the organizations again. This time the organizations, including their parenthesis, are unified. The model is fed with 410 paragraphs, and around 2.780 examples of organizations annotated, and the data are split into 80% for training and 20% for testing. The steps required are 210 and as presented in Figure 5.5 the model quickly achieves a great F1 score of 92,15%.

Step	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
25	No log	0.124631	0.826401	0.818996	0.822682	0.968003
50	No log	0.072902	0.871622	0.924731	0.897391	0.980932
75	No log	0.055345	0.897785	0.944444	0.920524	0.985112
100	No log	0.059109	0.898973	0.940860	0.919440	0.985242
125	No log	0.059840	0.901724	0.937276	0.919156	0.984850
150	No log	0.062843	0.906412	0.937276	0.921586	0.984197
175	No log	0.063564	0.903114	0.935484	0.919014	0.984328
200	No log	0.065888	0.903282	0.937276	0.919965	0.984459

Figure 5.5: Fine-tuning of best-fine-tuned model

Finally, we submit the model to the hugging face hub and run the same test as before. The issue at hand appears to be resolved (Figure 5.6).

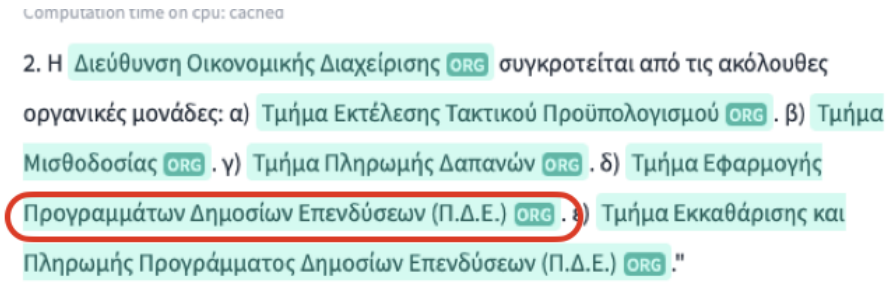

Figure 5.6: Hugging Face Live Test

5.2 Relation Extraction

5.2.1 Main process

In this part we will focus on the Relation Extraction task. As a result, we should train a model that can determine whether or not there is a link between the organizations. To do this, we manually annotate an 8 FEK dataset, as detailed in section 4.2. The outcome is three files (train, valid, and test) in JSON format, which have 2227, 289, and 263 organizations respectively. The training dataset has 6222 examples of no relation, 1528 of relation, and 210 examples of co-reference. The validation dataset has 784 examples of no relationship, 196 of relationship, and 24 of co-reference, while the examples in the test dataset are 869, 202, and 10 respectively. Table 5.2 contains this information.

	train	valid	test
NoRel	6222	784	869
SubOrg	1528	196	202
CoRef	210	24	10

Table 5.2: Distribution of classes within dataset files

The data are in JSON format and the structure is clearly described in section 4.2. As a result, we must convert data into a proper format that can be fed into the model We follow a straightforward procedure. We obtain the entire paragraph from the "sentText" key value, and then we choose a relationship from the "relationMentions" key value. Then we split the sentence into tokens and the <S:ORG> and </S:ORG> tokens are added before and after the subject entity. In the same manner that we treat object entities, but in this case, the tokens added are <O:ORG> and </O:ORG>. The following is the final format of the input paragraph:

{"tokens": ["\"2.", "<S:ORG>", "Περιφερειακές", "Διευθύνσεις", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "</S:ORG>:", "α.", "<O:ORG>", "Περιφερειακή", "Διεύθυνση", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Αθηνών", "</O:ORG>,", "με", "έδρα", "το", "Δήμο", "Αθηναίων.", "Η", "Περιφερειακή", "Διεύθυνση", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Αθηνών", "συγκροτείται", "από", "τα", "ακόλουθα", "Τμήματα:", "αα)", "Τμήμα", "Συντονισμού", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Αθηνών", "με", "έδρα", "την", "Αθήνα.", "ββ) Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Ανατολικού", "Τομέα", "Αθηνών,", "με", "έδρα", "την", "Αθήνα.", "γγ)", "Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Δυτικού", "Τομέα", "Αθηνών,", "με", "έδρα", "την", "Αθήνα.", "δδ)", "Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Γλυφάδας-Δάφνης,", "με", "έδρα", "την", "Ηλιούπολη.", "εε)", "Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Καλλιθέας,", "με", "έδρα", "την", "Καλλιθέα.", "στστ)", "Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Νέας", "Ιωνίας,", "με", "έδρα", "τη", "Νέα", "Ιωνία.",

"ζζ)", "Τμήμα", "Επιθεώρησης", "Εργασιακών", "Σχέσεων", "Αγίας", "Παρασκευής,", "με", "έδρα", "την", "Αγία", "Παρασκευή.\""], "label": 2}

One last detail required is the label of this relationship. In other words, whether there is one or not a linkage. This information is extracted through the "relationMentions" and is appended to our transformed data. Worth noting is that the CoRef relationships are not taken into account. Of course, this whole process is executed through a loop since more than one relationships exist in a single input.

The next step is to include the new tokens (S:ORG>, /S:ORG>, O:ORG>, /O:ORG>) into our tokenizer and then transform the data into a machine-readable format, as we did throughout the NER process by padding them to the same length.

Now that we have our data in the right format, we will fine-tune the Greek-Legal-Bert-V2 language model. Figure 5.7 depicts the training arguments used and the best model to be kept is based on the F1 score. The problem is seen as a text classification task, and the AutoModelForSequenceClassification argument is utilized to solve it. The total number of training steps needed is 2.425, and the training time takes around 34 minutes.

```
[ ]  model_name = BASE_MODEL_NAME.split("/")[-1]
     args = TrainingArguments(
         f"{model_name}-finetuned-re-V2",
         save_strategy="epoch",
         learning_rate=2e-5,
         num_train_epochs=5,
         weight_decay=0.01,
         per_device_train_batch_size=BATCH_SIZE,
         per_device_eval_batch_size=BATCH_SIZE,
         push_to_hub=True,
         save_total_limit=1,
         metric_for_best_model='f1',
         greater_is_better=True,)
```

Figure 5.7: Training Arguments of Relation Extraction Model

After performing the tests, the model earns an F1 score of 97,01%, and we finally push the model to the hugging face hub, where we can even execute some live testing. Figures 5.8 and 5.9 show two of them, the first of which has a connection while the other does not.

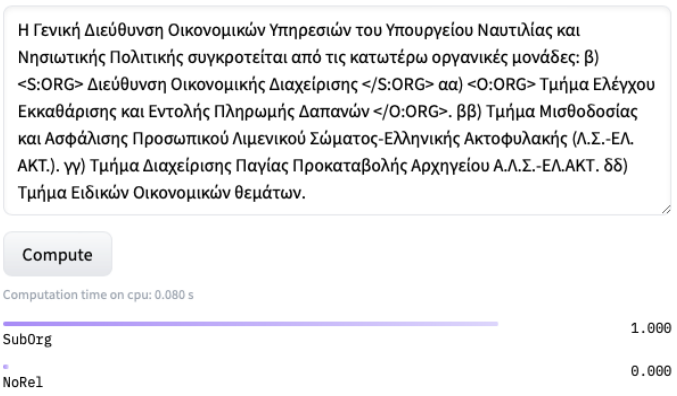

Figure 5.8: Hugging Face test of presenting a relationship

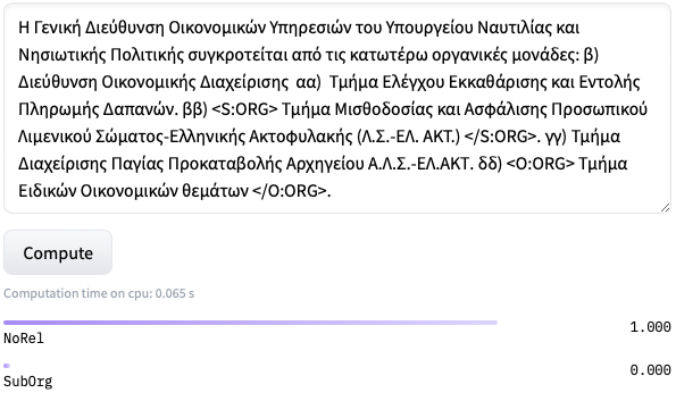

Figure 5.9: Hugging Face test of presenting no relationship

Finally, Figure 5.10 displays the whole procedure discussed in this section. We will investigate improvements in the next sub-part of the RE task process, but the general structure will remain unaltered.

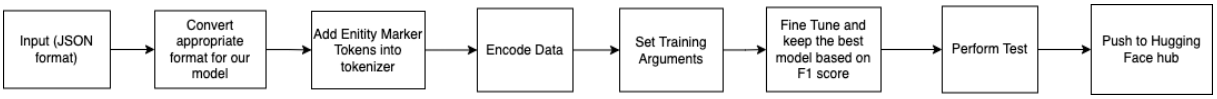

Figure 5.10: Relation Extraction Live Test

5.2.2 Enhance the Results

One common error produced by the tests is that the same organizations are recognized as having a link in certain situations. For example, consider the following sentence: "*2.β.* ***Γενική***

Γραμματεία Κοινωνικής Αλληλεγγύης *Στη* ***Γενική Γραμματεία Κοινωνικής Αλληλεγγύης*** *υφίσταται θέση Τομεακού Γραμματέα. Ο Τομεακός Γραμματέας είναι επικεφαλής της διοικητικής ιεραρχίας της* ***Γενικής Γραμματείας Κοινωνικής Αλληλεγγύης*** *και είναι υπεύθυνος για την εκτέλεση και τη διοικητική εφαρμογή της πολιτικής των υπηρεσιών των οποίων προΐσταται, όπως αυτή καθορίζεται από την Κυβέρνηση και τα όργανά της."* The words in bold highlight the same organization. When asked if there is a relationship between them, sometimes the model generates that there is. So we must deal with this situation.

The problem lies in the fact that the "CoRef" label was not taken into account during training. So, we simply change this name to "NoRel", as shown in figure 5.11, and all these entity combinations are regarded to have no relationship. Following that, we repeat the processes shown in Figure 5.10, and the results are somewhat poorer, with a 95.93% F1 score, but the problem of detecting the identical entities as having a relationship is no longer present.

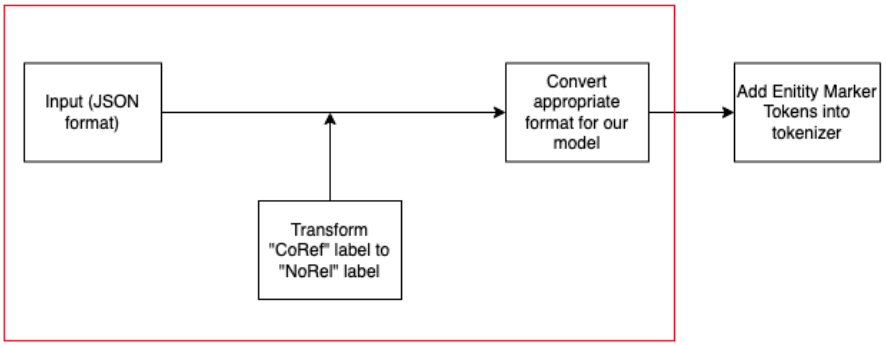

Figure 5.11: Enhanced Region in our RE Process

5.3 Pipeline

As shown in section 4.1 and figure 4.1 there is a pipeline we follow after we construct the tools required to detect the organizations and then assess their connection status. We investigate its piece of the process further in the following sub-parts to have a better grasp of the entire procedure. For representation reasons, suppose we have the following phrases from which we want to extract the organizational structure.

A. *"2. Η Γενική Διεύθυνση Διοικητικών Υπηρεσιών και Ηλεκτρονικής Διακυβέρνησης συγκροτείται από τις ακόλουθες οργανικές μονάδες: α) Διεύθυνσης Υποστήριξης Ανθρώπινου Δυναμικού και Υπηρεσιών. β) Διεύθυνση Προμηθειών και Μέριμνας. γ) Διεύθυνση Ηλεκτρονικής Διακυβέρνησης και Εξυπηρέτησης του Πολίτη."*

B. *"2. Η Διεύθυνση Υποστήριξης Ανθρώπινου Δυναμικού και Υπηρεσιών συγκροτείται από τις ακόλουθες οργανικές μονάδες: α) Τμήμα Διοίκησης Ανθρώπινου Δυναμικού Υπουργείου Εργασίας Κοινωνικής Ασφάλισης και Κοινωνικής Αλληλεγγύης (ΥΠ.Ε.Κ.Α.Α.) και Σώματος Επιθεώρησης Εργασίας (Σ.ΕΠ.Ε.). β) Τμήμα Διοίκησης Ανθρώπινου Δυναμικού Εποπτευομένων Φορέων. γ) Τμήμα Οργάνωσης και Απλούστευσης Διαδικασιών Υπουργείου Εργασίας Κοινωνικής Ασφάλισης και Κοινωνικής Αλληλεγγύης (ΥΠ.Ε.Κ.Α.Α.) και Σώματος Επιθεώρησης Εργασίας (Σ.ΕΠ.Ε.) δ) Τμήμα Οργάνωσης και Απλούστευσης Διαδικασιών Εποπτευομένων Φορέων. ε) Τμήμα Ποιότητας, Αποδοτικότητας και Επιμόρφωσης Ανθρώπινου Δυναμικού. στ) Τμήμα Γραμματείας."*

5.3.1 Remove Abbreviations inside Parenthesis

The significance of this phase has previously been discussed in section 5.1.4, where we modify our model to include the organizations' abbreviations inside the parentheses. However, to be more certain of the findings, we do an additional step before inputting data for entity identification and eliminate abbreviations using the following regex script::

```
    for string in row: #remove Abbreviations in parentheses
          row = re.sub("\([A-Ωα-ω]*.[A-Ωα-ω.]*\)","",string)
          row = re.sub(' +', ' ',row)
```

In our instances, running this script keeps the first paragraph intact while removing the parentheses in the second paragraph and transforming it as follows:

"2. Η Διεύθυνση Υποστήριξης Ανθρώπινου Δυναμικού και Υπηρεσιών συγκροτείται από τις ακόλουθες οργανικές μονάδες: α) Τμήμα Διοίκησης Ανθρώπινου Δυναμικού Υπουργείου

Εργασίας Κοινωνικής Ασφάλισης και Κοινωνικής Αλληλεγγύης και Σώματος Επιθεώρησης Εργασίας. β) Τμήμα Διοίκησης Ανθρώπινου Δυναμικού Εποπτευομένων Φορέων. γ) Τμήμα Οργάνωσης και Απλούστευσης Διαδικασιών Υπουργείου Εργασίας Κοινωνικής Ασφάλισης και Κοινωνικής Αλληλεγγύης και Σώματος Επιθεώρησης Εργασίας δ) Τμήμα Οργάνωσης και Απλούστευσης Διαδικασιών Εποπτευομένων Φορέων. ε) Τμήμα Ποιότητας, Αποδοτικότητας και Επιμόρφωσης Ανθρώπινου Δυναμικού. στ) Τμήμα Γραμματείας."

It is worth noting that abbreviations outside parentheses and normal text within parenthesis remain unaffected. The sentence stays the same, as shown in the example below.

Initial: "*Περιφερειακές Υπηρεσίες είναι οι Περιφερειακές Διοικήσεις του Λ.Σ.-ΕΛ.ΑΚΤ, οι υπαγόμενες σε αυτές Λιμενικές Αρχές (Κεντρικά Λιμεναρχεία, Λιμεναρχεία, Λιμενικοί Σταθμοί και Λιμενικά Τμήματα)*"
Converted: "*Περιφερειακές Υπηρεσίες είναι οι Περιφερειακές Διοικήσεις του Λ.Σ.-ΕΛ.ΑΚΤ, οι υπαγόμενες σε αυτές Λιμενικές Αρχές (Κεντρικά Λιμεναρχεία, Λιμεναρχεία, Λιμενικοί Σταθμοί και Λιμενικά Τμήματα)*"

5.3.2 Perform NER on the inputs

Now that out sentences are properly formatted, we have first to identify the entities. To accomplish so, we use the hugging face, which provides a pre-built pipeline for doing the token classification work. We input the best model we have built, together with its tokenizer, as seen below, and we are ready to execute the process.

```
    ner_model=AutoModelForTokenClassification.from_pretrained("amichailidis/greek_legal_bert_v2-finetuned-ner-V3")
    ner_tokenizer=AutoTokenizer.from_pretrained("amichailidis/greek_legal_bert_v2-finetuned-ner-V3")
    ner_pip = pipeline("ner", model=ner_model, tokenizer=ner_tokenizer, grouped_entities=True)
```

5.3.3 Transform Data into a readable format for RE

To execute, the relation extraction requires a specific structure of inputs, which is explicitly stated in the second paragraph of section 5.2.1. So, the next step is to change the data in this manner. Using the previously recognized entities, we build pairs of entities to check their connection, recognize their location inside the paragraph, and add the <S:{:ORG}>, </S:{:ORG}>, <O:{:ORG}>, </O:{:ORG}> accordingly. The text in our first example will be converted as follows:

2. "Η ***<S:{:ORG}> Γενική Διεύθυνση Διοικητικών Υπηρεσιών και Ηλεκτρονικής Διακυβέρνησης </S:{:ORG}>*** *συγκροτείται από τις ακόλουθες οργανικές μονάδες: α)* ***<O:{:ORG}> Διεύθυνσης Υποστήριξης Ανθρώπινου Δυναμικού και Υπηρεσιών. </O:{:ORG}>*** *β) Διεύθυνση Προμηθειών και Μέριμνας. γ) Διεύθυνση Ηλεκτρονικής Διακυβέρνησης και Εξυπηρέτησης του Πολίτη."*

5.3.4 Perform RE on the transformed data

The only difference between this step and the one mentioned in part 5.3.2, is that we utilize the class of "AutoModelForSequenceClassification" and insert the best model built for relation extraction.

```
re_model=AutoModelForSequenceClassification.from_pretrained("amichailidis/greek_legal_bert_v2-finetuned-re-V2")
re_tokenizer=AutoTokenizer.from_pretrained("amichailidis/greek_legal_bert_v2-finetuned-re-V2")
re_pip=pipeline("text-classification",model=re_model,tokenizer=re_tokenizer)
```

5.3.5 Post-Processing of Relationships extracted

We now know about the relationships between the entities after we have completed all of the preceding steps. So, in our first example, we'll get the following results:

1. *γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης--διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων:****SubOrg***

2. *γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης--διευθυνση προμηθειων και μεριμνας:**SubOrg***
3. *γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης--διευθυνση ηλεκτρονικης διακυβερνησης και εξυπηρετησης του πολιτη:**SubOrg***
4. *διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων--διευθυνση προμηθειων και μεριμνας:NoRel*
5. *διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων--διευθυνση ηλεκτρονικης διακυβερνησης και εξυπηρετησης του πολιτη:NoRel*
6. *διευθυνση προμηθειων και μεριμνας--διευθυνση ηλεκτρονικης διακυβερνησης και εξυπηρετησης του πολιτη:NoRel*

The label *"SubOrg"* represents that there is a relationship, but we have labeled it this way because having a relationship between the organizations means that the object organization is also a sub-organization of the other.

We only maintain the occurrences where our model identified a *"SubOrg"* link between the entities and append them to a data frame. We eliminate redundant relationships that may have been extracted for optimization purposes in the following steps. In our cases, the data frame is as follows:

	subject	object	relation
0	γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης	διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	SubOrg
1	γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης	διευθυνση προμηθειων και μεριμνας	SubOrg
2	γενικη διευθυνση διοικητικων υπηρεσιων και ηλεκτρονικης διακυβερνησης	διευθυνση ηλεκτρονικης διακυβερνησης και εξυπηρετησης του πολιτη	SubOrg
3	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα διοικησης ανθρωπινου δυναμικου υπουργειου εργασιας κοινωνικης ασφαλισης και κοινωνικης αλληλεγγυης και σωματος επιθεωρησης εργασιας	SubOrg
4	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα διοικησης ανθρωπινου δυναμικου εποπτευομενων φορεων	SubOrg
5	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα οργανωσης και απλουστευσης διαδικασιων υπουργειου εργασιας κοινωνικης ασφαλισης και κοινωνικης αλληλεγγυης και σωματος επιθεωρησης εργασιας	SubOrg
6	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα οργανωσης και απλουστευσης διαδικασιων εποπτευομενων φορεων	SubOrg
7	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα ποιοτητας, αποδοτικοτητας και επιμορφωσης ανθρωπινου δυναμικου	SubOrg
8	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	τμημα γραμματειας	SubOrg

We are now only one step away from completing our knowledge graph. Only the fact that the organizations recognized are not in the same declension is absent. As a result, we must detect these situations and replace one of them with a common declension so that our model can match them and generate the graph. To deal with this problem we use the Fuzz package in python. To solve this problem, we use Python's Fuzz module. A robust package that also

allows us to conduct text similarity using transformers, maintains only examples where the resemblance is more than 90%. After that, we simply remove the proposed common term from our model in all of the organizations that should be updated in our data frame. In our example, the model produced the following results:

	index	From	To	Similarity	Group
0	3	διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων	0.933	διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων

So, we just replace the organization "*διευθυνση υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων*" with the "*διευθυνσης υποστηριξης ανθρωπινου δυναμικου και υπηρεσιων*" in our data frame.

After we have validated that the entities are of the same declension, we do a check to guarantee that no relationship between the same entity exists. Even though we attempted to handle this issue during the training of the relations extraction model, it may emerge in some circumstances. So we just check to see if the two cells in the dataframe are identical, and if they are, we eliminate the row.

5.3.6 Create a Knowledge Graph

The next stage, and the heart of our pipeline thus far is to represent the information gathered in a knowledge graph. More precisely, to construct an organization chart from the input examples we have. To do so, we use the NetworkX package and by feeding the data frame as an input we create our graph, which is presented in figure 5.12. There are plenty of formats for depicting relationships, but the one shown here was deemed the best for portraying an organization's structure.

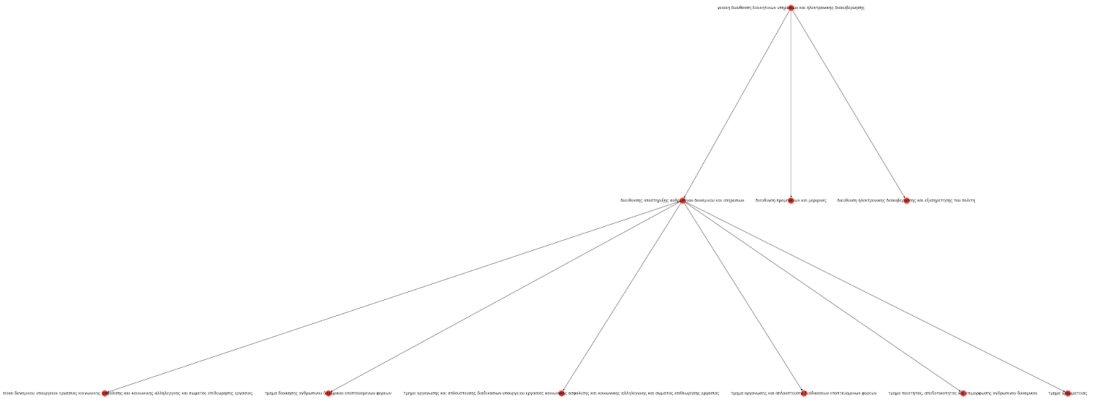

Figure 5.12: Organization Chart

To summarize, this section describes the entire process of deriving organizational structure from the text. The aim is to enter text into a model and then, after running the models we built for NER and RE, and with a few extra processing steps in between, generate a knowledge graph similar to the one shown in Figure 5.12.

6. Experimentation on Entire Document

In this part, we will attempt to implement the pipeline described in the previous section with two input paragraphs into a whole legislative text and analyze the results of our research. To do so, there are few limitations on the transformers-based models that we have fine-tuned.

Starting, the models may initially take a maximum of 512 tokens. In other words, the maximum number of tokens for an input paragraph is 512. If not, the analysis would crash or the paragraphs would be truncated. In our case, many paragraphs to be fed into our model are longer than 512, therefore we would lose a considerable deal of information.

The performance of the models is another disadvantage. To analyze the full FEK, we would need a substantial amount of processing power and plenty of time. Therefore, we should find a technique to eliminate some superfluous content from the text while retaining the paragraphs relating to the organizational framework.

Lastly, we should consider that our pipeline to be set into action should be fed with text inputs. However, Greek legislative documents are in PDF format, so we must extract the text from them, which is not always a straightforward task.

6.1 Process Performed

To overcome all these hurdles, we utilize some packages developed by the IHU Dorg team, which are freely accessible on GitHub under the repository name gg-extraction-2022. In particular, they have developed a model for extracting articles from PDF documents and preserving them in a simple text format by deleting articles referring neither to organizational structure nor the responsibilities inside it. And by initiating a second model, these articles are broken down into paragraphs, and candidates that explain the organizational structure are retained. Worth noting is that these models are mostly based on handcrafted patterns of both specific keywords, and also words and phrases formats. So, we need initially clone the repository locally and add the required modules as follows:

```
from rbner.rbNER import rbNER
rbner = rbNER()
```

```
from rbner.respas import respas
from src.fek_parser import PreParser, FekParser
from rbner.structure import structure

import re
from collections import OrderedDict


from src.rb_respas_tool import respas
```

Then, we choose a legislative document from those already uploaded to the repository, obtain its path, and execute the process described before. Following this, we execute a series of basic, yet essential actions to prepare our dataset. First, we append the paragraphs to a list, then we execute a script to guarantee that any empty components are eliminated, and finally, we flatten the list, as nested lists may include more than two items-paragraphs.

After performing these simple actions, we add the list to a data frame, remove paragraph characters, and create a second cell, which will count the number of words, to gain a more comprehensive view of our data. Figure 6.1 depicts such a paragraph, which illustrates the difficulties we must overcome. The paragraph has more than 512 tokens, but the most significant aspect is that the first phrase may represent an organizational structure, while the remaining information is superfluous. The same applies to a few other examples. By correcting this issue, the length matter will also be resolved. So, we just generate a list of the terms that are most pertinent to responsibilities, and we execute a for loop that examines each row of the data frame, and if any of these terms are detected, the text after them is erased. Finally, these rows are replaced with the newly formed ones. Following lines of code are used for this task:

```
words=["αρμοδιότητες", "αρμόδιο", "αρμόδια", "αρμόδιες", "υπάγεται"]
for par in range(len(paragraphs['paragraph'])):
    sentence = paragraphs['paragraph'].iloc[par]
    regex = r"\b(?:{})\b".format("|".join(words))
    new_sentence = re.split(regex,sentence)
    paragraphs['paragraph'].iloc[par] = new_sentence[0]
```

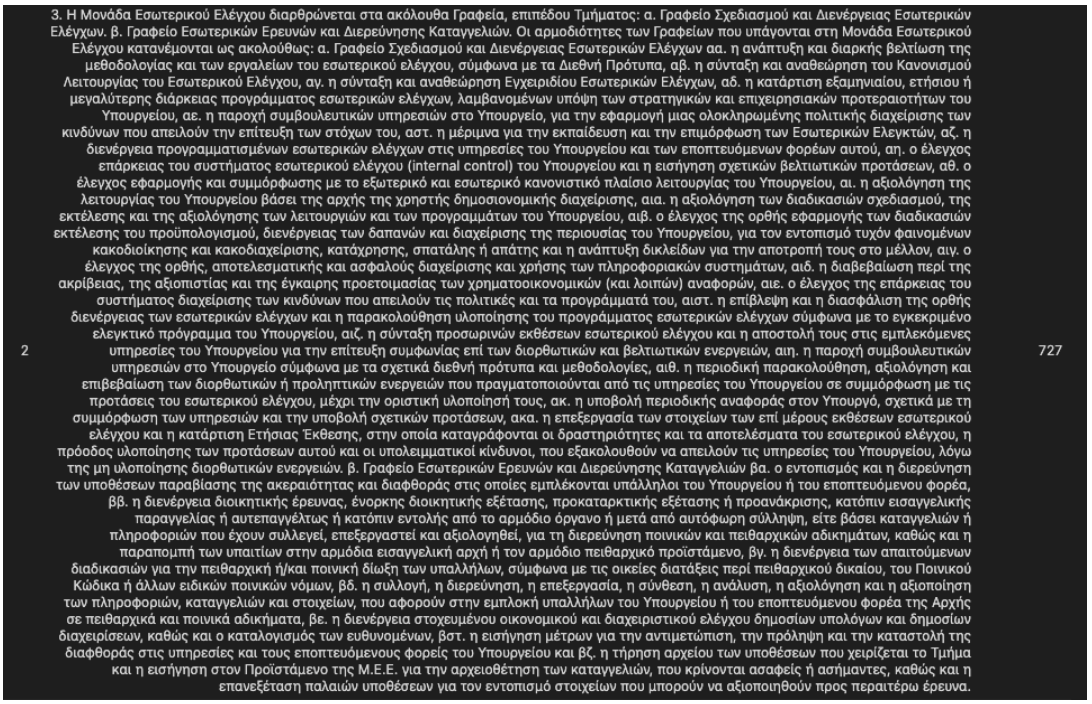

Figure 6.1: Indicative Paragraph Extracted

At the end, we simply save the data frame as a CSV file, which will be incorporated into our pipeline for additional analysis.

6.2 Outcome of Analysis

Now that we have the input file, the only remaining step is to feed the data into our pipeline described in section 5.3. We feed them in and the graph presented in Figure 6.2 is derived. There are a few positives and a few cons in the graph. The primary advantage is that a structure is given clearly. This indicates that our pipeline identified some entities and their relationships. However, some entities are not related to one another. This is an issue with how some entities are mentioned in the text. For instance, in certain cases, the Ministry of Migration Policy (Υπουργείο Μεταναστευτικής πολιτικής), could be mentioned just as Ministry (Υπουργείο). This would result in model mismatches, and as a result, the structure could not be represented in its entirety.

Another problem arouse was due to the following paragraph that should be examined:

2. α. Στον Υπηρεσιακό Γραμματέα του άρθρου 36 του ν. 4622/2019 υπάγονται οι κάτωθι υπηρεσίες: αα. Γενική Διεύθυνση Διοικητικών και Οικονομικών Υπηρεσιών. αβ. Γενική Διεύθυνση Πληροφορικής και Επικοινωνιών. αγ. Υπηρεσία Συντονισμού. αδ. Τεχνική Υπηρεσία. αε. Γραφείο Υπηρεσιακού Γραμματέα. αστ. Ιατρείο. β. Η Γενική Γραμματεία Μεταναστευτικής Πολιτικής, διαρθρώνεται ως ακολούθως: βα. Γενική Διεύθυνση Μεταναστευτικής Πολιτικής. ββ. Γενική Διεύθυνση Συντονισμού και Διαχείρισης Προγραμμάτων Ταμείου Ασύλου, Μετανάστευσης και Ένταξης και Ταμείου Εσωτερικής Ασφάλειας και άλλων πόρων. βγ. Υπηρεσία Ασύλου. βδ. Αρχή Προσφυγών. βε. Ιδιαίτερο Γραφείο Γενικού Γραμματέα Μεταναστευτικής Πολιτικής. Στη Γενική Γραμματεία Μεταναστευτικής Πολιτικής προΐσταται ο ανωτέρω Γενικός Γραμματέας. γ. Η Γενική Γραμματεία Υποδοχής Αιτούντων Άσυλο διαρθρώνεται ως ακολούθως: γα. Υπηρεσία Υποδοχής και Ταυτοποίησης (Υ.Π.Υ.Τ). γβ. Ιδιαίτερο Γραφείο Γενικού Γραμματέα Υποδοχής Αιτούντων Άσυλο. Στη Γενική Γραμματεία Υποδοχής Αιτούντων Άσυλο προΐσταται ο ανωτέρω Γενικός Γραμματέας. δ. Η Ειδική Γραμματεία Προστασίας Ασυνόδευτων Ανηλίκων διαρθρώνεται ως ακολούθως: δα. Μονάδα Εποπτείας και Αξιολόγησης Κέντρων Φιλοξενίας. δβ. Μονάδα Διαχείρισης Αιτημάτων Στέγασης και Μετεγκαταστάσεων. δγ. Μονάδα Ένταξης και Υποστήριξης Ασυνόδευτων Ανηλίκων. δδ. Μονάδα Θεσμικής Προστασίας. δε. Ιδιαίτερο Γραφείο Ειδικού Γραμματέα Προστασίας Ασυνόδευτων Ανηλίκων. Στην Ειδική Γραμματεία Προστασίας Ασυνόδευτων Ανηλίκων προΐσταται ο ανωτέρω Ειδικός Γραμματέας. ε. Η Ειδική Γραμματεία Συντονισμού Εμπλεκομένων Φορέων διαρθρώνεται ως ακολούθως: εα. Μονάδα Μητρώου Μη Κυβερνητικών Οργανώσεων. εβ. Μονάδα Συνεργασίας με Μη Κυβερνητικές Οργανώσεις, Ευρωπαϊκούς και Διεθνείς Οργανισμούς και Υπηρεσίες. εγ. Μονάδα Ανταποδοτικών Προγραμμάτων. εδ. Μονάδα Συνεργασίας με φορείς και Ο.Τ.Α. εε. Ιδιαίτερο Γραφείο Ειδικού Γραμματέα Συντονισμού Εμπλεκομένων Φορέων.

The paragraph might be quite complicated and not well-formated, yet our model did not fully meet the requirements, since it captured unnecessary relations. In the remaining simpler paragraphs, however, the model produced pretty excellent results.

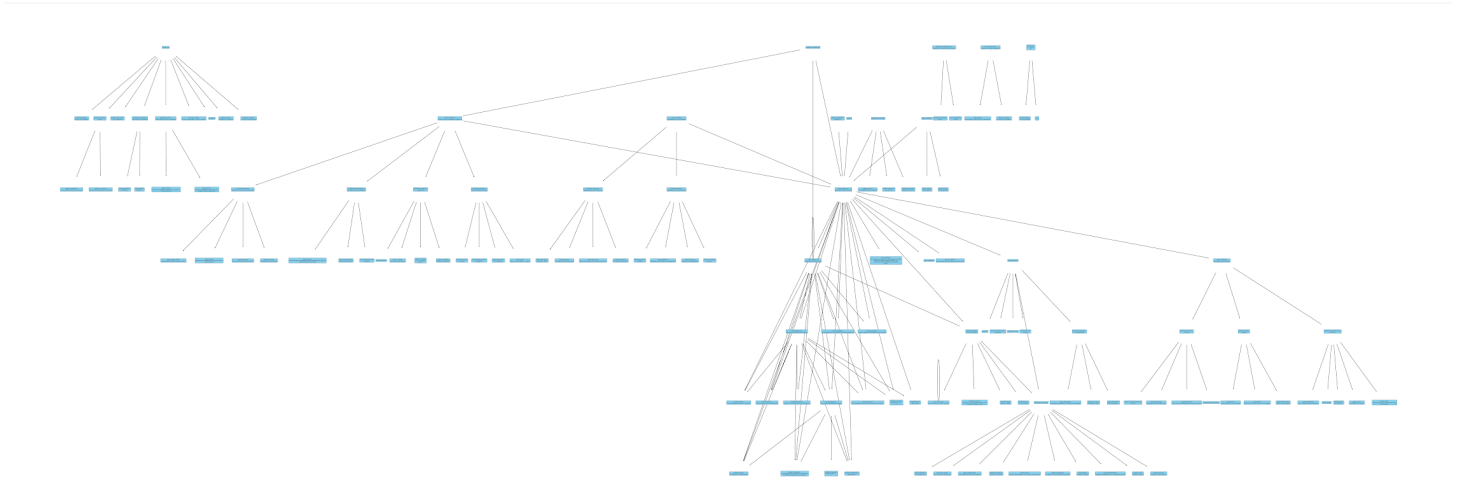

Figure 6.2: Organization Chart of Hellenic Ministry of Migration Policy

Summing up, we can see that our pipeline has some potential, but deep learning models alone are insufficient to solve the problem of extracting the structure of organizations from text, and it should be combined with some pre-process and post-process steps of hand-written patterns to compensate for their limitations. In other words, we should follow a hybrid method.

7. Conclusions & Future Work

In this section, we compare the key results reached in prior studies and discuss the current study's conclusion. We also talk about the implication, limits, and future research prospects.

7.1 Conclusions

The current thesis is concerned with the creation of a Knowledge Graph using Greek legal documents. The primary goal is to portray the structure of government organizations. This information may be found on legal papers published in PDF format by the National Printing Office (ET) in the Government Gazette (F.E.K). The DORG-IHU team retrieved the papers from "Kodiko," a publicly accessible website representing F.E.K through HTML, and transformed the material into CSV, a more organized format.

After identifying the 17 F.E.K that are most relevant to our situation, implying that the structure of Greek ministries is represented inside, we needed to first discover a means to extract the organizations from plain text, i.e. execute a named entity identification operation. Consequently, we had to evaluate the relationship between these organizations, which is a relation extraction task. And these two primary principles should be linked by a comprehensive pipeline that will take just paragraph inputs, conduct the relevant stages, and build a knowledge graph that will reflect the organizational structure indicated in the input paragraphs.

In the context of the current study, we manually annotated 10 F.E.K for the named entity task and 8 F.E.K, after discarding irrelevant information for our needs, both for the named entity and relation extraction task. To recognize the organizations and their relationships inside the plain text, transformers-based named entity recognition and relation extraction models were trained using the annotated data. After training the models, we built an exhaustive pipeline that, when fed input text, removes abbreviations inside the parenthesis, extracts organizations using the model we trained, transforms data into a valid format for the relation extraction task, runs the task, removes duplicate relationships, eliminated different declensions of the same organization, and finally generates an organizational structure graph.

To the best of our knowledge, the suggested solution is the first attempt to extract relationships from Greek legal documents. Furthermore, this is the first study to handle an information extraction job in this way. Most previous work that used supervised learning on Named Entity Recognition and Relation Extraction problems required to include processes for feature engineering to be fed for model training. However, because of the transformers architecture, which allows models to analyze content in a bidirectional manner and fine-tune powerful pre-trained models through transfer learning, we were able to avoid the step of feature engineering, such as knowing the part of speech of a word.

7.1.1 Findings

Deep learning models based on transformers have shifted the way we approach information extraction tasks. Three easy steps are required to derive the structure of public organizations from legal documents: a) Create a NER model, b) Create a RE model, and connect them into a single pipeline by incorporating some pre- and post-processing stages.

We achieved pretty satisfactory results by utilizing the pre-trained language model for Greek Legislation and fine-tuning it for our NER assignment. When we fine-tuned the model we constructed to remove the existence of a problem, the findings bounced off.

The same holds for the RE model we built, which is the first transformers-based relation extraction model established for the Greek language. This model was refined and performed well on a binary classification challenge, but it could also be trained on a multi-classification task, requiring, though, a larger annotated training dataset

Finally, the most significant result is that we can obtain essential structured information from text by fine-tuning two deep learning models and placing them in an exhaustive pipeline with certain pre and post-processing stages.

7.2 Implications

7.2.1 Theoretical Implications

Existing research has been worked on the information extraction task. However, when attempting to accomplish this utilizing the most recent transformers-based deep learning approaches, there appears to be a gap in the literature. From a theoretical standpoint, this work advances our understanding of transformer power and how it may be used to increase our research skills in the information extraction task and the NLP sector in general.

Furthermore, our research indicated that we could extract useful information from Greek manuscripts. This technique, in particular, expands the capabilities of the public sector since we can extract structured information from the documents, which will improve the open data policy that the public sector should follow and bring us one step closer to the ultimate stage of Open Government.

7.2.2 Management Implications

This thesis also has some practical ramifications for companies or brands. To begin, businesses could take advantage of transformers-based deep learning models, which provide excellent results while requiring only a little amount of CPU capacity.

Furthermore, they may benefit from the entire process we went through. They could extract structured information from their data by tailoring the procedure to their needs, which could then be further analyzed or enhanced using publicly available information under the open data concept.

Finally, if the public sector provided data in a more structured format and adhered more closely to the concept of Open Government, the private sector would profit from a significant amount of data that it could access and analyze, which would improve the business decisions of companies and accelerate their development.

7.3 Limitation and Suggestion for Future Research

Our findings should be viewed in light of their limitations. Our goal was to build a pipeline that would take some inputs and return an organizational structure. As a result, we did not dive into the training efficiency of our NER and RE models. There is a probability that the models can be further optimized by doing numerous tests on the training arguments combination. The same applies to the training data. We annotated a quite satisfying number of legal papers, but additional data is needed for our model to become more robust.

A notable drawback of the findings reported during the analysis of the entire FEK is that few entities are referred to as a whole solely in the article's title. Therefore, we need to run a pre-process and replace the entities in the text with their complete forms through the title.

Furthermore, the process could be hybridized since we could utilize heuristics to increase the performance of our model. Some regex patterns, particularly in the case of NER, might assist us in recognizing some organizations that are not discovered by our model or even filter out incorrect predictions of the model. In the same direction, pre-processing of the text with hand-written patterns could help us break complex paragraphs in simpler ones so that our model can process them more efficiently.

Moreover, we should bear in mind that certain paragraphs may include more than 512 tokens and that these excess tokens should not be eliminated even after the process of checking and removing sentences if a term from a list appears. In this scenario, the paragraph should be divided into almost equal-sized batches. However, we should keep in mind that in most cases the main entity lies at the start of the paragraph, thus the first line should be repeated at the beginning of each batch.

Additionally, the relationships extracted could also be represented using a metalanguage, the most prevalent of which being RDF (Resource Description Framework), in which relations are represented as triples in the form of entity-relation-entity, also known as a subject-predicate-object expression. This form of representation allows representing, sharing

and reusing data in a structured and interoperable way constituting 4-start rating according to Tim Burners Lee.

Worth mentioning is that this process, or each sub-part, could be expanded into and trained for alternative tasks. For instance, one could adjust this process to recognize the purpose or responsibilities of each organization. In this case, the model should not only be able to extract the organizations, but also the section where the objective or duties are indicated.

Finally, we should note that there is a great need for such research and applications utilizing state-of-the-art deep learning models, such as Bert, in low-resource languages, like Greek in our case, becasue most studies focus on English. So

References

[1] Colpaert, P., Joye, S., Mechant, P., Mannens, E., & Van de Walle, R. (2013). The 5 stars of open data portals. In *Proceedings of the 7th International Conference on Methodologies, Technologies and Tools Enabling E-Government (MeTTeG13), University of Vigo, Spain* (pp. 61-67).

[2] Lathrop, D., & Ruma, L. (2010). *Open government: Collaboration, transparency, and participation in practice*. " O'Reilly Media, Inc.".

[3] Krasadakis, P., Sakkopoulos, E., & Verykios, V. S. (2021, November). A Natural Language Processing Survey on Legislative and Greek Documents. In *25th Pan-Hellenic Conference on Informatics* (pp. 407-412).

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

[5] Vajjala, S., Majumder, B., Gupta, A., & Surana, H. (2020). *Practical natural language processing: a comprehensive guide to building real-world NLP systems*. O'Reilly Media.

[6] Jain, D., Borah, M. D., & Biswas, A. (2020, December). Fine-tuning text rank for legal document summarization: A Bayesian optimization-based approach. In *Forum for Information Retrieval Evaluation* (pp. 41-48).

[7] Papageorgiou, H., Prokopidis, P., Giouli, V., & Piperidis, S. (2000, May). A Unified POS Tagging Architecture and its Application to Greek. In *LREC*.

[8] Xiao, L., Wissmann, D., Brown, M., & Jablonski, S. (2004). Information extraction from the web: System and techniques. *Applied Intelligence*, *21*(2), 195-224.

[9] Small, S. G., & Medsker, L. (2014). Review of information extraction technologies and applications. *Neural computing and applications*, *25*(3), 533-548.

[10] Ralph Grishman. Information extraction: Techniques and challenges.
Information Extraction A Multidisciplinary Approach to an Emerging Information Technology, (2):10–27, 1997.

[11] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press., 2008.

[12] Jurafsky, D., & Manning, C. (2012). Natural language processing. *Instructor*, *212*(998), 3482.

[13] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2010). Part-of-speech tagging for Twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

[14] Bach, N. X., Linh, N. D., and Phuong, T. M. (2018). An empirical study on pos tagging for Vietnamese social media text. *Computer Speech & Language*, 50:1–15.

[15] Liu, F., Weng, F., and Jiang, X. (2012). A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1035–1044. Association for Computational Linguistics.

[16] Nikiforos, M. N., & Kermanidis, K. L. (2020). A Supervised Part-Of-Speech Tagger for the Greek Language of the Social Web. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 3871-3877).

[17] Prokopis Prokopidis, Byron Georgantopoulos, and Haris Papageorgiou. 2011. A SUITE OF NATURAL LANGUAGE PROCESSING TOOLS FOR GREEK. (2011), 9

[18] Partalidou, E., Spyromitros-Xioufis, E., Doropoulos, S., Vologiannidis, S., & Diamantaras, K. I. (2019). Design and implementation of an open-source Greek POS Tagger and Entity Recognizer using spaCy. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 337-341). IEEE.

[19] Gamit, M. V., Joshi, R., & Patel, E. (2019). A Review on Part-Of-Speech Tagging on the Gujarati Language. *International Research Journal of Engineering and Technology (IRJET)*.

[20] Bhirud, N., Tataale, S., Randive, S., & Nahar, S. (2019). A literature review on chatbots in the healthcare domain. *International journal of scientific & technology research*, *8*(7), 225-231.

[21] Manning, Christopher D. (2011). “Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?” In: Computational Linguistics and Intelligent Text Processing. Ed. by Alexander F. Gelbukh. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 171–189.

[22] Hasan, F. M. (2006). *Comparison of different POS tagging techniques for some South Asian languages* (Doctoral dissertation, BRAC University).

[23] Mehta, D. N., & Desai, N. (2011). A survey on part-of-speech tagging of Indian languages. In *1st International Conference on Computing, Communication, Electrical, Electronics, Devices and Signal Processing* (Vol. 34).

[24] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[25] Hongdou, Li. (2019). *A Brief Introduction to Named Entity Recognition.* https://medium.com/@hli111/a-brief-introduction-to-named-entity-recognition-59320b08dfe2

[26] Hwang, W., Kim, S., Seo, M., Yim, J., Park, S., Park, S., ... & Lee, H. (2019, September). Post-OCR parsing: building a simple and robust parser via BIO tagging. In *Workshop on Document Intelligence at NeurIPS 2019*.

[27] Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000). Rule-based named entity recognition for Greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)* (pp. 75-78).

[28] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. (2020a). On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

[29] Nikos Bartziokas, Thanassis Mavropoulos, and Constantine Kotropoulos. 2020. Datasets and Performance Metrics for Greek Named Entity Recognition. In 11th Hellenic Conference on Artificial Intelligence. 160–167.

[30] Alonso, M. A., Gómez-Rodríguez, C., & Vilares, J. (2021). On the use of parsing for named entity recognition. *Applied sciences*, *11*(3), 1090.

[31] Hangya, V., & Farkas, R. (2017). A comparative empirical study on social media sentiment analysis over various genres and languages. *Artificial Intelligence Review*, *47*(4), 485-505.

[32] Na, S. H., Li, J., Shin, J. H., & Kim, K. (2018). Transition-Based Korean Dependency Parsing Using Hybrid Word Representations of Syllables and Morphemes with LSTMs. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *18*(2), 1-20.

[33] Yoshua Bengio. 2009. Learning deep architectures for AI. Found. Trends Mach. Learn. 2, 1 (2009), 1–127.

[34] Ji, T., Wu, Y., & Lan, M. (2019). Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*(pp. 2475-2485).

[35] Massimo Poesio, Roland Stuckardt, and Yannick Versley. 2016. *Anaphora resolution*. Springer.

[36] Kees van Deemter and Rodger Kibble. 2000. On co-referring: Coreference in muc and related annotation schemes. *Computational Linguistics*, 26:629–637.

[37] Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. *Information Fusion*, *59*, 139-162.

[38] Fillmore, C. J. (1987). Pragmatically controlled zero anaphora. In *Annual Meeting of the Berkeley Linguistics Society*(Vol. 12, pp. 95-107).

[39] Dixon, R. M. (2003). Demonstratives: A cross-linguistic typology. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, *27*(1), 61-112.

[40] Nissim, M., Dingare, S., Carletta, J., & Steedman, M. (2004, May). An Annotation Scheme for Information Status in Dialogue. In *LREC*.

[41] Mitkov, R., Evans, R., & Orasan, C. (2002, February). A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 168-187). Springer, Berlin, Heidelberg.

[42] Stylianou, N., & Vlahavas, I. (2021). Transformed: End-to-End transformers for evidence-based medicine and argument mining in the medical literature. *Journal of Biomedical Informatics*, *117*, 103767.

[43] Denis, P., & Baldridge, J. (2007, April). Joint determination of anaphoric and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*(pp. 236-243).

[44] Yang, X., Zhou, G., Su, J., & Tan, C. L. (2003, July). Coreference resolution using competition learning approach. In *Proceedings of the 41st annual meeting of the association for computational linguistics* (pp. 176-183).

[45] Plu, J., Prokofyev, R., Tonon, A., Cudré-Mauroux, P., Difallah, D. E., Troncy, R., & Rizzo, G. (2018, May). Sanaphor++: Combining deep neural networks with semantics for coreference resolution. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[46] Fernandes, E., dos Santos, C., & Milidiú, R. L. (2012, July). Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task* (pp. 41-48).

[47] Joshi, M., Levy, O., Weld, D. S., & Zettlemoyer, L. (2019). BERT for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

[48] Vilain, M., Burger, J. D., Aberdeen, J., Connolly, D., & Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

[49] Moosavi, N. S., & Strube, M. (2016, August). Which coreference evaluation metric do you trust? a proposal for a link-based entity-aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 632-642).

[50] Luo, X. (2005, October). On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (pp. 25-32).

[51] Shi, P., & Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

[52] Christou, D., & Tsoumakas, G. (2021). Extracting Semantic Relationships in Greek Literary Texts. *Sustainability*, *13*(16), 9391.

[53] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

[54] Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017). Triviaqa: A large-scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

[55] Lin, H., Yan, J., Qu, M., & Ren, X. (2019, May). Learning dual retrieval module for semi-supervised relation extraction. In *The World Wide Web Conference* (pp. 1073-1083).

[56] Fader, A., Soderland, S., & Etzioni, O. (2011, July). Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1535-1545).

[57] Papadopoulos, D., Papadakis, N., & Matsatsinis, N. (2021). PENELOPIE: Enabling open information extraction for the greek language through machine translation. *arXiv preprint arXiv:2103.15075*.

[58] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

[59] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, *21*(140), 1-67.

[60] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

[61] Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., & Weld, D. S. (2011, June). Knowledge-based weak supervision for information extraction of overlapping

relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (pp. 541-550).

[62] Boecking, B., Neiswanger, W., Xing, E., & Dubrawski, A. (2020). Interactive weak supervision: Learning useful heuristics for data labeling. *arXiv preprint arXiv:2012.06046*.

[63] Gal, Y., Islam, R., & Ghahramani, Z. (2017, July). Deep Bayesian active learning with image data. In *International Conference on Machine Learning* (pp. 1183-1192). PMLR.

[64] Tong, S. (2001). *Active learning: theory and applications*. Stanford University.

[65] Lampoltshammer, T. J., & Wass, C. (2015, May). EU Project open laws. EU. In *Conference for E-Democracy and Open Government* (p. 305).

[66] Boella, G., Di Caro, L., Graziadei, M., Cupi, L., Salaroglio, C. E., Humphreys, L., & Stroetmann, V. (2015, June). Linking legal open data: breaking the accessibility and language barrier in European legislation and case law. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law* (pp. 171-175).

[67] Hoekstra, R. (2011, October). The MetaLex document server. In *International Semantic Web Conference* (pp. 128-143). Springer, Berlin, Heidelberg.

[68] Force, E. T. (2016). ELI: a technical implementation guide.

[69] Alexander, B. O. E. R. (2009). LKIF core: Principled ontology development for the legal domain. *Law, ontologies and the semantic web: channeling the legal information flood*, *188*, 21.

[70] Van Opijnen, M. (2011). European Case Law Identifier: an indispensable asset for legal information retrieval.

[71] Francesconi, E., Küster, M. W., Gratz, P., & Thelen, S. (2015, August). The ontology-based approach of the publications office of the EU for document accessibility and open data services. In *International Conference on Electronic Government and the Information Systems Perspective* (pp. 29-39). Springer, Cham.

[72] Angelidis, I., Chalkidis, I., Nikolaou, C., Soursos, P., & Koubarakis, M. (2018). Nomothesia: a linked data platform for Greek legislation.

[73] Koniaris, M., Papastefanatos, G., & Vassiliou, Y. (2016, November). Towards automatic structuring and semantic indexing of legal documents. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics* (pp. 1-6).

[74] Garofalakis, J., Plessas, K., & Plessas, A. (2016, November). A semi-automatic system for the consolidation of Greek legislative texts. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics* (pp. 1-6).

[75] Garofalakis, J., Plessas, K., Plessas, A., & Spiliopoulou, P. (2018, November). A project for the transformation of Greek legal documents into legal open data. In *Proceedings of the 22nd Pan-Hellenic Conference on Informatics* (pp. 144-149).

[76] Garofalakis, J., Plessas, K., Plessas, A., & Spiliopoulou, P. (2019, June). Modeling Legal Documents for Their Exploitation as Open Data. In *International Conference on Business Information Systems* (pp. 30-44). Springer, Cham.

[77] Garofalakis, J., Plessas, K., Plessas, A., & Spiliopoulou, P. (2019). Application of an Ecosystem Methodology Based on Legal Language Processing for the Transformation of Court Decisions and Legal Opinions into Open Data. *Information*, *11*(1), 10.

[78] Angelidis, I., Chalkidis, I., & Koubarakis, M. (2018, September). Named Entity Recognition, Linking, and Generation for Greek Legislation. In *JURIX* (pp. 1-10).

[79] Chalkidis, I., Androutsopoulos, I., & Michos, A. (2017, June). Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*(pp. 19-28).

[80] Chalkidis, I., & Androutsopoulos, I. (2017, December). A Deep Learning Approach to Contract Element Extraction. In *JURIX*(pp. 155-164).

[81] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2019). Extreme multi-label legal text classification: A case study in EU legislation. *arXiv preprint arXiv:1905.10892*.

[82] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

[83] Stasis, A., Dalakou, V., Karakatsanis, I., Demiri, L., Valatsou, G., & Sarantis, D. (2020, September). Better access to law by codification and consolidation of legal acts: the case of the Hellenic Law Codification Portal. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance* (pp. 696-704).

[84] Avgerinos Loutsaris, M., Lachana, Z., Alexopoulos, C., & Charalabidis, Y. (2021, June). Legal Text Processing: Combing two legal ontological approaches through text mining. In *DG. O2021: The 22nd Annual International Conference on Digital Government Research* (pp. 522-532).

[85] Stylianou, N., Vlachava, D., Konstantinidis, I., Bassiliades, N., & Peristeras, V. (2022). Doc2KG: Transforming Document Repositories to Knowledge Graphs. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *18*(1), 1-20.

[86] Athinaios, K., Chalkidis, I., Pantazi, D. A., & Papaloukas, C. (2020). *Named Entity Recognition using a Novel Linguistic Model for Greek Legal Corpora based on BERT model* (Doctoral dissertation, BS Thesis, School of Science, Department of Informatics and Telecommunications).

[87] Apostolopoulou, A. G., Briakos, S. A., & Pantazi, D. A. (2021). NLP Tasks with GreekLegalBERT v2.