**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**

**SCHOOL OF SCIENCES**
**DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

**BSc THESIS**

# Extending YAGO4 Knowledge Graph with Geospatial Knowledge

**Maria-Despoina D. Siampou**

**SUPERVISORS:** **Manolis Koubarakis**, Professor NKUA

**ATHENS**

**SEPTEMBER 2020**

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# Επέκταση του γράφου γνώσης YAGO4 με Γεωχωρική Πληροφορία

**Μαρία-Δέσποινα Δ. Σιάμπου**

**ΕΠΙΒΛΕΠΟΝΤΕΣ:** **Μανόλης Κουμπαράκης**, Καθηγητής ΕΚΠΑ

**ΑΘΗΝΑ**

**ΣΕΠΤΕΜΒΡΙΟΣ 2020**

**BSc THESIS**

Extending YAGO4 Knowledge Graph with Geospatial Knowledge

**Maria-Despoina D. Siampou**
**SN:** 1115201600151

**SUPERVISORS:** **Manolis Koubarakis**, Professor NKUA

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Επέκταση του γράφου γνώσης YAGO4 με Γεωχωρική Πληροφορία

**Μαρία-Δέσποινα Δ. Σιάμπου**
**ΑΜ:** 1115201600151

**ΕΠΙΒΛΕΠΟΝΤΕΣ:** **Μανόλης Κουμπαράκης**, Καθηγητής ΕΚΠΑ

# ABSTRACT

YAGO is one of the largest knowledge bases in the Linked Open Data cloud. The latest version of YAGO, YAGO4, reconciles the rigorous typing and constraints of schema.org with the rich instance data of Wikidata. The resulting resource contains 2 billion type-consistent triples for 64 Million entities, and has a consistent ontology that allows semantic reasoning with OWL 2 description logics. In this work we present an extension of YAGO4 with qualitative geospatial information, extracted from Greek Administrative Geography (GAG) dataset. Our main goal is to extend and create preexisting and missing entities respectively, without introducing any duplication knowledge that already exists in the knowledge graph.



**SUBJECT AREA**: Semantic Web

**KEYWORDS**: Linked Open Data, Knowledge Graphs, Knowledge Bases

# ΠΕΡΙΛΗΨΗ

Το YAGO είναι μία από τις μεγαλύτερες βάσεις γνώσης που διαθέτει τα δεδομένα της ως Ανοικτά Διασυνδεδεμένα Δεδομένα. Η τελευταία της έκδοση, YAGO4, συνδιάζει τους λεπτομερείς περιορισμούς με τις πλούσιες οντότητες που παρέχονται από το schema.org και το Wikidata, αντίστοιχα. Με αυτό τον τρόπο, προκύπτουν 2 δισεκατομμύρια "σταθερές" τριάδες που αντιστοιχούν σε 64 εκατομμύρια οντότητες, ενώ ταυτόχρονα δημιουργείται μια σταθερή οντολογία που επιτρέπει τη σημασιολογική συλλογιστική πορεία σύμφωνα με την λογική περιγραφή του OWL 2. Σε αυτή τη δουλειά, επεκτείνουμε αυτό τον γράφο γνώσης με ποιοτική γεωχωρική πληροφορία, η οποία παρέχεται από το σύνολο δεδομένων της Ελληνικής Διοικητικής Γεωγραφίας. Κύριος στόχος μας, είναι η επέκταση των ήδη υπάρχοντων οντοτήτων, καθώς και η δημιουργία εκείνων που λείπουν, χωρίς την εισαγωγή διπλότυπης πληροφορίας.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ**: Σημασιολογικός Ιστός

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**: Ανοικτά Διασυνδεδεμένα Δεδομένα, Γράφοι Γνώσης, Βάσεις Γνώσης

## ACKNOWLEDGEMENTS

I would like to thank my supervisors, Prof. Manolis Koubarakis, Nikolaos Karalis and Georgios Mandilaras for their excellent guidance and assistance during the preparation of this thesis.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

## 1.1 Semantic Web

The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term "Semantic Web" [1] refers to W3C's vision [2] of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL and OWL.

### 1.1.1 Linked Data

The Semantic Web is a Web of data - of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF [3] provides the foundation for publishing and linking these data. It specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. It, nonetheless, extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link. Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations.

### 1.1.2 Query

Query languages go hand-in-hand with databases. SPARQL [4] is the query language for the Semantic Web and can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. It also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

### 1.1.3 Vocabularies

At times it may be important or valuable to organize data. In order to enrich data, OWL was created. The W3C Web Ontology Language (OWL) [5] is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies.

### 1.1.4 Shapes Constraint Language

Shapes Constraint Language (SHACL) [6] is a language for validating RDF graphs against a set of conditions. These conditions are provided as shapes and other constructs expressed in the form of an RDF graph. RDF graphs that are used in this manner are called "shapes graphs" in SHACL and the RDF graphs that are validated against a shapes graph are called "data graphs". As SHACL shape graphs are used to validate that data graphs satisfy a set of conditions they can also be viewed as a description of the data graphs that do satisfy these conditions. Such descriptions may be used for a variety of purposes beside validation, including user interface building, code generation and data integration.

### 1.1.5 Inference

On the Semantic Web, data is modeled as a set of (named) relationships between resources. Using inferences allows us to generate new relationships based on the existing data as well as, some additional information in the form of a vocabulary (e.g. a set of rules) via automatic procedures. Whether the new relationships are explicitly added to the set of data, or are returned at query time, is an implementation issue.

## 1.2 Goal of this work and next chapters

The goal of this work is to extend the knowledge graph of YAGO4 with more geospatial information. The spatial information in YAGO4 is represented with the properties hasLongitute and hasLatitude. Our aim is to extend YAGO4 with geometries from the Greek Administrative Geometry (GAG) dataset, as well as other properties (e.g. population), which are not present in YAGO4. An example of a YAGO4 entity is presented at Fig. 1.1.

```
<https://yago-knowledge.org/resource/yago:Athens>
            rdf:type schema:AdministrativeArea;
            rdf:type schema:City;
            rdf:type schema:Place;
            rdf:type schema:Thing;
            rdf:type yago:Capital_City;
            rdfs:label "ΑΘΗΝΑ"@el;
            schema:geo <geo:37.979444444444745,23.7161111111113> .
```

**Figure 1.1: YAGO4 entity**

The rest of this thesis is structured as follows. Chapter 2 discusses background and related works, while chapter 3 gives detailed information about the design of YAGO4. In chapter 4 we demonstrate the administrative organization of Greece, as well as the inconsistencies found regarding this in YAGO4. Then, in chapter 5, we discuss the methodology that we followed in order to extend YAGO4 with geospatial information. In Chapter 6 we present the extended knowledge graph of YAGO4. Finally, in Chapter 6 we summarize our contributions, present our conclusions and discuss future work.

# 2. BACKGROUND AND RELATED WORK

## 2.1 Schema

The term ”schema” refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). It is mainly, a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. This describes how real-world entities are modeled in the database.

### 2.1.1 Schema.org

Schema.org [7] is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond. Schema.org vocabulary can be used with many different encodings, including RDFa, Microdata and JSON-LD. These vocabularies cover entities, relationships between entities and actions, and can easily be extended through a well-documented extension model. Over 10 million sites use Schema.org to markup their web pages and email messages.

## 2.2 Knowledge Base

A knowledge base (KB) is a machine-readable collection of knowledge about the real world, which contains entities (such as organizations, movies, people, and locations) and relations between them (such as birthPlace, director, etc.). KBs have wide applications in search engines, question answering, fact checking, chatbots, and many other NLP and AI tasks. Numerous projects have constructed KBs automatically or by help of a community. Notable KBs include YAGO [8], DBpedia [9], BabelNet [10], NELL [11], and Wikidata [12].

### 2.2.1 WikiData

Wikidata [12] is a free, open knowledge base that can be read and edited by both humans and machines, which are able to control both its data and schema. Wikidata acts as central storage for the structured data of its Wikimedia sister projects such as, Wikipedia, Wiktionary, Wikisource etc. It is a multilingual knowledge base, and unlike Wikipedia which has different versions for every language, the information of the entities of Wikidata is translated to multiple languages. Wikidata is a part of the Linked Data Cloud, since its data is available in RDF.

### 2.2.2 Freebase

Freebase [13] was a large collaborative knowledge base consisting of data composed mainly by its community members. It was an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions. Freebase aimed to create a global resource that allowed people (and machines) to access common information more effectively. In 2016, Freebase was shut down, having its data moved to Wikidata.

### 2.2.3 YAGO

YAGO [8] was one of the first academic projects to build a knowledge base automatically. The main idea of YAGO was to harvest information about entities from the infoboxes and categories of Wikipedia, and to combine this data with an ontological backbone derived from classes in WordNet. Since Wikipedia is an excellent repository of entities, and WordNet is a widely used lexical resource, the combination proved useful.

YAGO sent each fact through a pipeline of filtering, constraint checking, and de-duplication steps. This procedure scrutinized noisy input and boosted the quality of the final KB, to a manually verified accuracy of 95%. This precision was possible thanks to the tight control that the YAGO creators had over the extraction process, the filtering process, the ontological type system, the choice of the relations, and the semantic constraints.

YAGO2 [14], the second version of YAGO, was released in 2011. YAGO2 introduces spatial and temporal information to the YAGO knowledge graph. Wikipedia is not the only source from which YAGO2 extracts spatial information. The YAGO2 knowledge base is extended with information from a new source, GeoNames.

For this purpose a new class yagoGeoEntity that groups together all geoentities, i.e. all entities with a permanent physical location on Earth, is introduced in YAGO2. The subclasses of yagoGeoEntity are: location, body of water, geological formation, real property, facility, excavation, structure, track, way, and land.

Geographical coordinates, consisting of latitude and longitude, are used to describe the position of a geoentity. A special data type is also introduced to store geographical coordinates, yagoGeoCoordinates. An instance of yagoGeoCoordinates is a pair of a latitude and a longitude value. Each instance of yagoGeoEntity was directly connected to its geographical coordinates by the hasGeoCoordinates relation. YAGO2 only knows about coordinates, not polygons, so even locations that have a physical extent are represented by a single geocoordinate pair. As we extract these coordinates from Wikipedia, the assignment of coordinates to larger geoentities follows the rules given there: for a settlement like a city, it represents the center, for military and industrial.

Temporal information was added mainly to entities that represent people, groups, artifacts or events. YAGO3 [15], came out in 2015. YAGO3 combines information from Wikipedias in multiple languages.

Despite the new versions YAGO2 and YAGO3 with substantial jumps in scope and size, the focus on Wikipedia infoboxes meant that YAGO has not arrived at the same scale as Freebase or Wikidata. YAGO4 [16] is the latest version of the YAGO knowledge base. It is based on Wikidata - the largest public general-purpose knowledge base.

YAGO4 refines the data as follows:

- All entity identifiers and property identifiers are human-readable.
- The top-level classes come from schema.org - a standard repertoire of classes and properties maintained by Google and others, combined with bioschemas.org. The lower level classes are a selection of Wikidata classes.
- The properties come from schema.org.
- YAGO 4 contains semantic constraints in the form of SHACL. These constraints keep the data clean and consistent, and allow for logical reasoning on YAGO.

YAGO contains more than 50 million entities and 2 billion facts.

### 2.3 Yago2Geo

Yago2Geo [17] is an extension of YAGO2 with geospatial information represented by geometries (e.g., lines, polygons, multipolygons, etc.) encoded by Open Geospatial Consortium standards.

The authors of YAGO2Geo used geographical administrative data provided by official sources of Greece, the United Kingdom and the Republic of Ireland. They also extracted geospatial information about the administrative units of every country from the GADM dataset as well as for other types of features, such as lakes, from OpenStreetMap. Apart from the geometries, each data source provides additional information (e.g., population for cities) that we include in YAGO2Geo.

The resulting knowledge graph is currently the richest in terms of geospatial information publicly available, open source, knowledge graph.

Yago2Geo is available publicly at `http://yago2geo.di.uoa.gr`. The free and open dataset there includes the extended KG encoded in RDF. The geospatial information follows the standards of the Open Geospatial Consortium, hence Yago2Geo can be queried using GeoSPARQL.

## 2.4 Summary

In this chapter we discussed previous and related works. In next chapter, we are presenting YAGO4 design choices.

# 3. UPPERCASE DESIGN

In this section, we demonstrate the design choices made for the construction of YAGO4 knowledge graph by the original authors.

## 3.1 Taxonomy

YAGO4 takes avantage of the stable class hierarchy of schema.org, as well as, the fine-grained classes that Wikidata offers. As a result, the top-level classes in YAGO4 come from schema.org - a standard repertoire of classes and properties maintained by Google and others, combined with bioschemas.org which is an extension of schema.org - while the leaves of its taxonomy correspond to WIkidata classes.

With these inputs, the YAGO4 taxonomy is then constructed as follows:

- For each instance in Wikidata, each possible path in the Wikidata taxonomy to the root node is taken into consideration. If the first class on the path has a Wikipedia article, it is then included in YAGO4.
- The path to the root in the Wikidata taxonomy is continued, discarding all classes on the way, until a class that has been mapped to schema.org is found. The path to the root in the schema.org taxonomy is then continued, adding all classes on the way to YAGO4.
- If a class that has been mapped to schema.org is not found, the entire path is discarded.
- All Wikidata classes that have less than 10 direct instances are also discarded.

As a result of the constraints mentioned, only 10k classes from the 2.4M original Wikidata classes were kept, shrinking the taxonomy by 99.6%.

In detail, 235 classes of schema.org were manually mapped to Wikidata classes, yet, those that could not be mapped, mostly shopping-related or social-media classes such as schema:LikeAction, were discarded.

For our purposes, we focus on the administrative organization of Greece, whose structure in the taxonomy in YAGO4 can be seen in Figure 3.1.

## 3.2 Entities

YAGO 4 is stored in the RDF data format. All entities consist of human-readable URIs, in order to make the KB more accessible for interactive use. If an entity has a Wikipedia page, then the entity name is its Wikipedia title. Otherwise, the entity name is consisted of the concatenation of the English label of the entity with its Wikidata identifier (e.g., Crete_Region_Q1267522). Studies like [1] suggest that the Wikidata labels are fairly stable, leading to fairly stable YAGO URIs. If the entity has no English label, then only the Wikidata identifier is used. This gives the vast majority of entities human-readable names, without introducing duplicates or ambiguity.

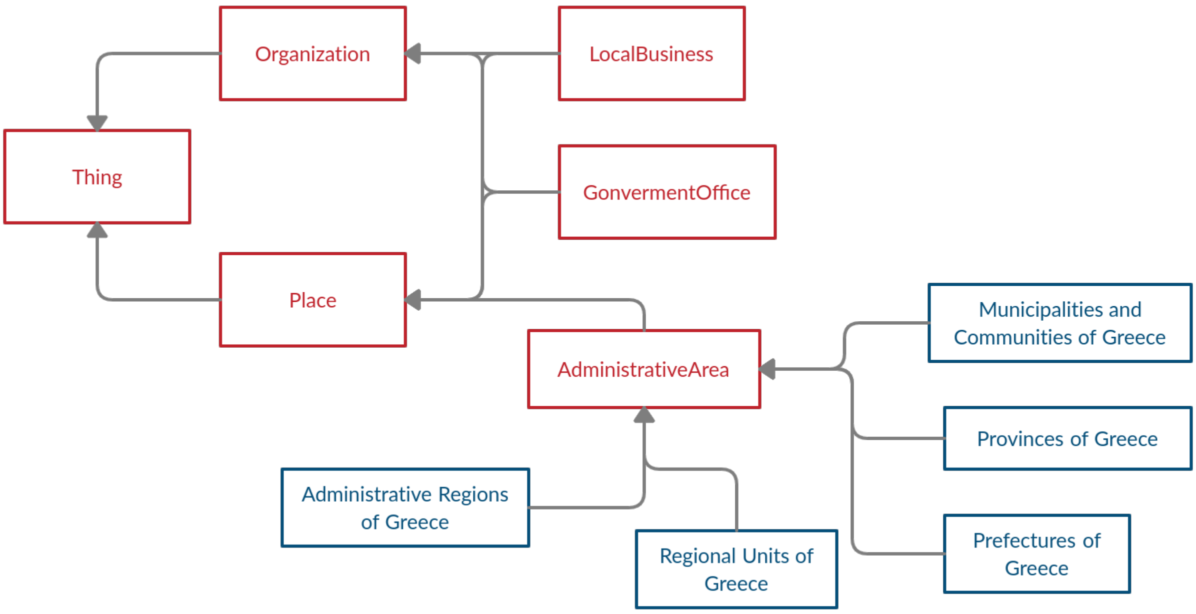


**Figure 3.1: YAGO4 Taxonomy**

*Red color indicates classes that were taken from shcema.org, whereas blue color indicates Wikidata classes.*

*GonvermentOffice class leads to -among others- the instances of the 7 Decentralized Administrations of Greece.*

## 3.3 Well-typed Values

YAGO4 has not just well-typed entities, but also well-typed literals. For this purpose, the data values of Wikidata are translated to RDF terms.

Additionally, globe coordinates are mapped to schema:GeoCoordinates resources, expressing location as a longitude-latitude pair. In section 4.4 , we introduce our methodology to express the geographic coordinates of each location with more complex geometries, such as polygons, lines and multipolygons.

```
<http://yago-knowledge.org/resource/Andros>
    <http://schema.org/geo> <geo:37.8333333333409,24.9333333333383>.
```

**Figure 3.2: Example of geoCoordinates in YAGO4**

## 3.4 Relations and Constraints

The relations in YAGO4 come from schema.org and are mapped to Wikidata predicates. Moreover, YAGO4 has hand-crafted semantic constraints that not just keep the data clean, but also allow logical reasoning on the data. These constraints are modeled in the W3C standards SHACL and OWL. In general, YAGO4 assumes that no other properties are allowed for each class, thereby interpreting the SHACL constraints under a “closed world assumption”. These constraints are automatically enforced during the construction of the KB, hence the data of YAGO4 satisfy all constraints. Lastly, the generated YAGO4 ontology uses the OWL2 axioms DisjointClasses, ObjectPropertyDomain, DataPropertyDomain, ObjectPropertyRange, DataPropertyRange, ObjectUnionOf, FunctionalDataProperty, FunctionalObjectProperty, and falls into the OWL DL flavor. For our extension purposes, we added several extra properties to YAGO4, presented in section 4.2.

```
schema:Person sh:property yago:birthPlaceProperty
yago:birthPlaceProperty sh:path schema:birthPlace
yago:birthPlaceProperty sh:node schema:Place
```

**Figure 3.3: Example of SHACL Constraints in YAGO4**

## 3.5 Summary

In this chapter, detailed information regarding YAGO4 design were given. In the next chapter, we present several inconsistencies that were spotted regarding the administrative organization of Greece.

# 4. THE ADMINISTRATIVE ORGANIZATION OF GREECE

In this section, we describe the current administrative organization of Greece. We, also, point out the inconsistencies in YAGO4 design that are related to this administrative organization.

## 4.1 The Kallikratis and Kleisthenis Plan

The administrative organization of Greece was formed by The Kallikratis Plan and and its recent reform, Kleisthenis Plan, which came into effect in 2019.

According to it, the first level of government consists of decentralized administrations, which are consisted of a group of regions.

Moreover, the second level is composed by regions, which are divided into regional units. A group of municipalities composes a regional unit.

The third level of government is constituted by municipalities which have resulted from merging several former municipalities and communities (themselves the subject of a previous reform with the 1997 Kapodistrias plan). The municipalities are further subdivided into municipal units which consist of municipal communities and local communities.

Subject of a previous reform with the 1997 Kapodistrias plan, were also provinces and prefectures. In Kallikratis Plan, provinces are totally abolished and prefectures are replaced by regions.

There are 7 decentralized administrations, 13 regions, 74 regional units and 325 municipalities in Greece. Regarding the last ones, Kleisthenis Plan, introduced the creation of 12 new municipalities, increasing their total number from 325 to 332.

## 4.2 Administrative divisions of Greece instances in YAGO4

Each of the classes related to the administrative divisions of Greece in YAGO4 (also presented in figure 3.1) leads to the following number instances:

- yago:Decentralized_Administrations: 7
- yago:Administrative_Regions_of_Greece: 13
- yago:Regional_units_of_Greece: 77
- yago:Municipalities_and_communities_of_Greece: 340
- yago:Provinces_of_Greece: 70

- yago:Prefectures_of_Greece: 55

Evaluating the quality of those instances, we noticed that regional units are scattered between prefectures and regional units, while some of them are instances of both classes. For example, *Pella_(regional_unit)* is both a type of *Regional_units_of_Greece* and *Prefectures_of_Greece*. Additionally, the class of *Provinces_of_Greece*, leads to 70 instances of past-existing provinces of Greece, which are possibly still existing in WikiData, such as *Imathia_Province*.

## 4.3 Consistency in YAGO4

Taking this description into consideration we can observe that YAGO4 taxonomy is not fully consistent with the administrative organization of Greece. In particular, we can observe that there are three classes representing provinces, prefectures and municipalities & communities of Greece in YAGO4, while classes representing municipal units, municipal communities and local communities do not exist at all.

Nevertheless, contrasting the number of instances of each YAGO4 class with the number of instances of their respective GAG class, we can observe that the classes of YAGO4 contain more entities than their corresponding official administrative divisions, mostly due to duplication (e.g. Regional Units & Prefectures) as well as outdated information (e.g. Provinces). However, YAGO4 contains the information regarding the new municipalities created by the Kleisthenis reform. Lastly, the number of instances of municipal units, local and municipal communities are non-existent due to the taxonomy deficiencies that were mentioned.

**Table 4.1: Number of instances in GAG and YAGO4**

| Administrative Unit | Number of instances in GAG | Number of instances in YAGO4 |
|---|---|---|
| Decentralized Adm. | 7 | 7 |
| Regions | 13 | 13 |
| Regional Units | 74 | 77 |
| Prefectures | 0 | 55 |
| Provinces | 0 | 70 |
| Municipalities | 325 | 340 |
| Municipal Units | 1034 | 0 |
| Municipal Communities | 3 | 0 |

## 4.4 Summary

In this chapter, we presented the Kallikratis Plan, the most recent administrative organization of Greece. In the next chapter, we propose several strategies to address the deficiencies and inconsistencies that were mentioned previously.

# 5. EXTEDNING YAGO4

In this chapter, we introduce the methodology followed in order to extend YAGO4 with geospatial information, as well as the results obtained.

## 5.1 Taxonomy Extension

As mentioned in section 3.1. the part of the taxonomy of YAGO4 pertaining the administrative divisions of Greece is not fully consistent with the administrative organization of Greece. We addressed this problem by introducing some extra classes into the leaf classes of YAGO4.

With respect to the pre-existing classes the taxonomy has been modified as follows:

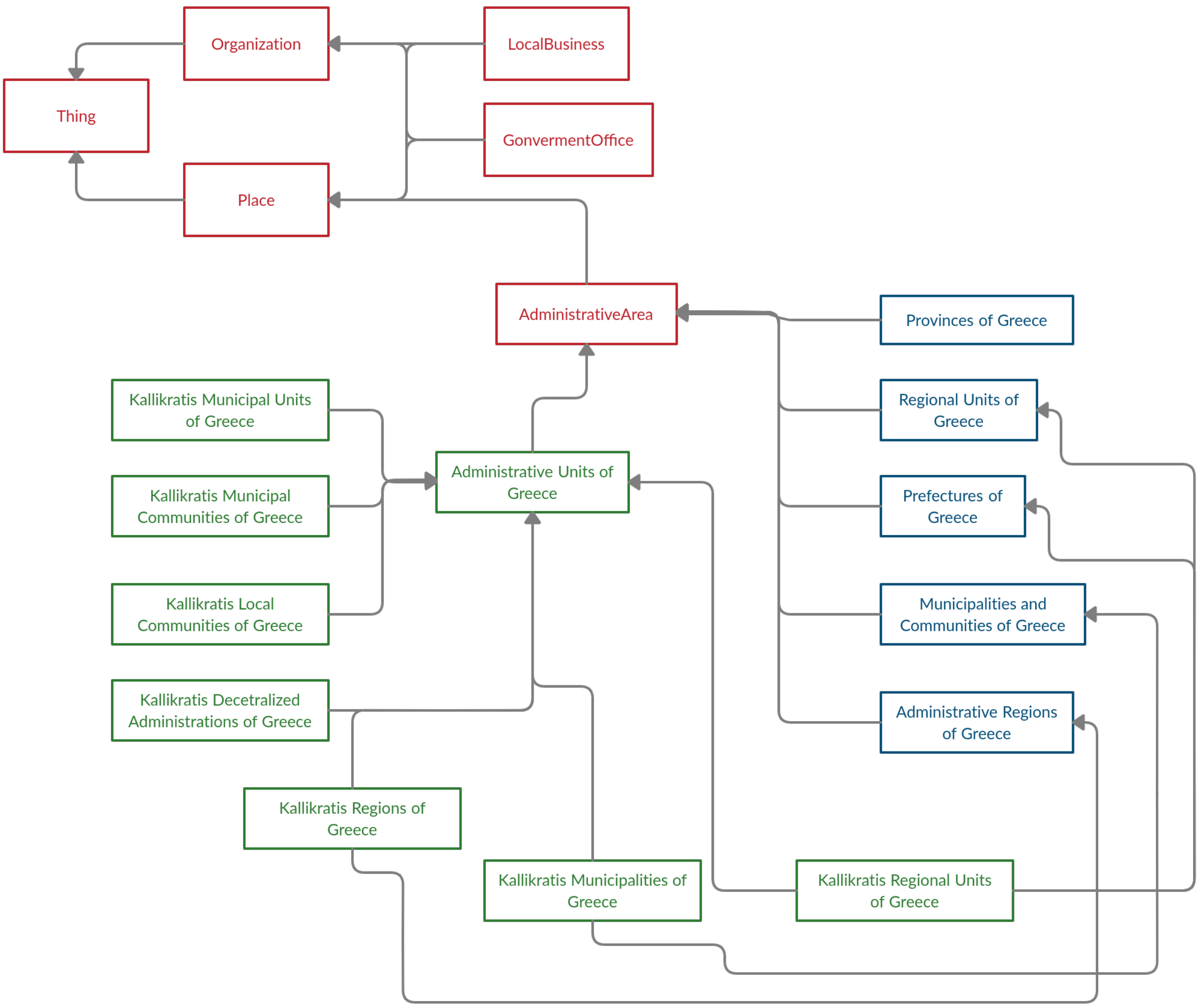


**Figure 5.1: Extended YAGO4 Taxonomy**

*Green color indicates the newly inserted classes.*

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix yagor: <http://yago-knowledge.org/resource/> .
@prefix exto: <http://kr.di.uoa.gr/yago4-extension/ontology#> .
@base <http://kr.di.uoa.gr/yago4-extension/ontology> .

<http://kr.di.uoa.gr/yago4-extension/ontology> a owl:Ontology ;
owl:versionIRI <http://kr.di.uoa.gr/yago4-extension/ontology/1.1.0> .

exto:Administrative_Units_of_Greece a owl:Class ;
rdfs:subClassOf <http://schema.org/AdministrativeArea> ;
rdfs:label "Administrative_Units_of_Greece"@en.

exto:Kallikratis_Municipalities_of_Greece a owl:Class ;
rdfs:subClassOf exto:Administrative_Units_of_Greece ,
yagor:Municipalities_and_communities_of_Greece ;
rdfs:label "Kallikratis_Municipalities_of_Greece"@en.
```

**Figure 5.2: Gist of Taxonomy Extension**

As presented above, we propose the insertion of a super class that represents the Administrative Units of Greece and is a subclass of AdministrativeArea. We further inserted 7 additional classes that represent each of those Administrative Units separately. Each class is a subclass of the Administrative Units of Greece class, as well as a subclass of its respective YAGO4 class -assuming it exists. For example, *Kallikratis Regions of Greece* is both a subclass of *Administrative Units of Greece* and *Administrative Regions of Greece*, whereas *Kallikratis Municipal Units of Greece* is only a subclass of *Administrative Units of Greece*.

This extension, allows us to extend YAGO4 with missing entities without violating its taxonomy and constraints.

## 5.2 Addition of Properties

The available properties of YAGO4 do not allow us to represent useful information of administrative units. Hence, we extend the taxonomy of YAGO4 with such properties, without violating the strict constraints that are in place.

1. Object Properties
   - has seat, which connects a municipality with a municipal or local community
2. Data Properties
   - has population
   - has code

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix yagor: <http://yago-knowledge.org/resource/> .
@prefix exto: <http://kr.di.uoa.gr/yago4-extension/ontology#> .
@base <http://kr.di.uoa.gr/yago4-extension/ontology> .

<http://kr.di.uoa.gr/yago4-extension/ontology> a owl:Ontology ;
owl:versionIRI <http://kr.di.uoa.gr/yago4-extension/ontology/1.1.0> .

exto:has_code a owl:DatatypeProperty ;
rdfs:domain exto:Administrative_Units_of_Greece ;
rdfs:range xsd:integer ;
rdfs:label "has_code"@en.
```

**Figure 5.3: Gist of Properties Addition**

## 5.3 Matching Phase

In order to extend each pre-existing YAGO4 entity with qualitative geospatial information, we tried to match each entity with one of our GAG dataset. For example, the YAGO4 entity *Athens_Municipality_Q1224979* and the GAG entity with identifier *9186*, represent the same location, therefore should be matched. For this purpose, the following two filters were implemented;

### 5.3.1 Label Similarity Filter

Label Similarity filter matches the entities of the two datasets that share similar labels, using Levenshtein distance between the two strings with a similarity threshold of 0.8. As GAG dataset's labels are in Greek, we only kept the labels written in Greek and English language from YAGO4 dataset. We further converted each Greek label to utf-8 format, in order to achieve a successful matching. The exact algorithm is presented bellow:

**Algorithm 1** Label Similarity Filter

```
INPUT: Yago, GAG
OUTPUT: Matched_Entities
Matched_Entities ← ∅
for Yago_Entity in Yago do
    MaxR ← 0
    MaxL ← " "
    Yago_Entity_Label ← Yago[Yago_Entity]
    for GAG_Entity in GAG do
        GAG_Entity_Label ← GAG[GAG_Entity]
        Dist ← Levenshtein(YAGO_Entity_Label, GAG_Entry_Label)
        if Dist > MaxR then
            MaxR ← Dist
            MaxL ← GAG_Entity_Label
        end
    end
    if MaxR ≥ threshold then
        Matched_Entities[MaxL] ← Yago_Entity
    end
end
```

### 5.3.2 Geometry Distance

Geometry Distance filter examines the entities matched at the previous step, discarding the matches that have an Euclidean distance greater than 25km. The Euclidean distance is calculated between polygons' centroid and point. In order, to compute the distances all coordinates where converted to the WGS:84 coordinate system. The methodology of the algorithm is presented bellow:

**Algorithm 2** Geometry Distance Filter

```
INPUT: Matched_Entities, Yago, GAG
OUTPUT: New_Matched_Entities
New_Matched_Entities ← ∅
for Entry in Matched_Entities do
    GAG_Entity ← Entry
    GAG_Entity_Geometry ← GAG[GAG_Entity]
    for Yago_Entity in Yago_Entities do
        Yago_Entity ← Matched_Entities[Entry]
        Yago_Entity_Geometry ← Yago[Yago_Entity]
        Dist ← Euclidean_Distance(Yago_Entity_Geometry, GAG_Entity_Geometry)
        if Dist < 25 then
            New_Matched_Entities[GAG_Entity] ← Yago_Entity
        end
    end
end
```

### 5.3.3 Results

Our results are presented at the table bellow:

**Table 5.1: Matching Phase Results**

| GAG | YAGO4 | GAG Matches | YAGO4 Matches |
|---|---|---|---|
| Decentralized Administrations | Government Offices | 7/7 | 7/7 |
| Regions | Regions of Greece | 13/13 | 13/13 |
| Regional Units | Regional Units and Prefectures of Greece | 67/74 | 78/92 |
| Municipalities | Municipalities and Communities of Greece | 322/325 | 331/340 |

Our methodology enabled us to match most of the entities existing in GAG dataset. In fact, many GAG entities were matched with more than one entities, yielding the existence of duplicate entities in YAGO4. For that reason, the number of YAGO4 matches in table 5.1 is greater than the number of GAG matches.

More precisely, all decentralized administration and region entities have been matched with exactly one YAGO4 entity. In addition, 67 regional units and 325 municipalities were matched with a YAGO4 entity, while 11 regional unit and 9 municipality duplicate instances were spotted. Finally, due to the absence of the municipal units, local and municipal communities classes, the corresponding GAG entities failed to be matched.

For our extension purposes, we discard duplicate entities, as we do not wish to introduce duplicate information in the extended knowledge graph.

### 5.3.4 Comparison with previous work

In our previous work we have extended YAGO2 knowledge graph with qualitative and quantitative geospatial information. For this purpose we used multiple sources, such as GADM, OSI and GAG dataset etc.

Regarding GAG dataset, in this work we managed to achieve - in average - a greater number of matches than in the previous one. More precisely, in the extension of YAGO2 knowledge graph most of the decentralized administration entities and regions were matched, but not all of them, as well as 21 regional units. However, the number of municipalities, municipal units and communities matched, beat our current matches.

The different results obtained by both cases, are mostly due to the different class hierarchies between the two versions of YAGO. In YAGO2 the information related with the administrative divisions of Greece lie upon the following classes:

- geoclass_administrative_division
- geoclass_first-order_administrative_division
- geoclass_second-order_administrative_division
- geoclass_third-order_administrative_division
- geoclass_fourth-order_administrative_division
- geoclass_fifth-order_administrative_division
- populated_place & locality

As we can observe, the taxonomy of YAGO4 -presented in section 3.1- is more compatible with the current administrative organization of Greece, in contrast with the taxonomy of YAGO2, a fact that renders the detection of the desired entities easier.

In addition, YAGO2 is built automatically from Wikipedia, GeoNames, and WordNet, resulting a KB that contains 447 million facts about 9.8 million entities. Yet, YAGO4, which is entirely built of the rich instance data of Wikidata, contains 2 billion type-consistent triples for 64 million entities.

According to the design choices made for the two versions of YAGO, as well as the results from our works, we came to the following conclusions:

1. The high number of unmatched regional units entities in YAGO2 is due to their absence at the knowledge base.
2. YAGO4 contains many duplicate entities, especially regional units and municipalities. For example <*https://yago-knowledge.org/resource/Andros*> and <*https://yago-knowledge.org/resource/Andros_(town)*> are both referring to Andos Municipality.
3. There is a lack of information regarding municipal units & communities in YAGO4, resulting from the absence of the relating Wikidata instances.

The number of matches resulted in both versions of YAGO- YAGO2 and YAGO4- are presented in figure 5.4.

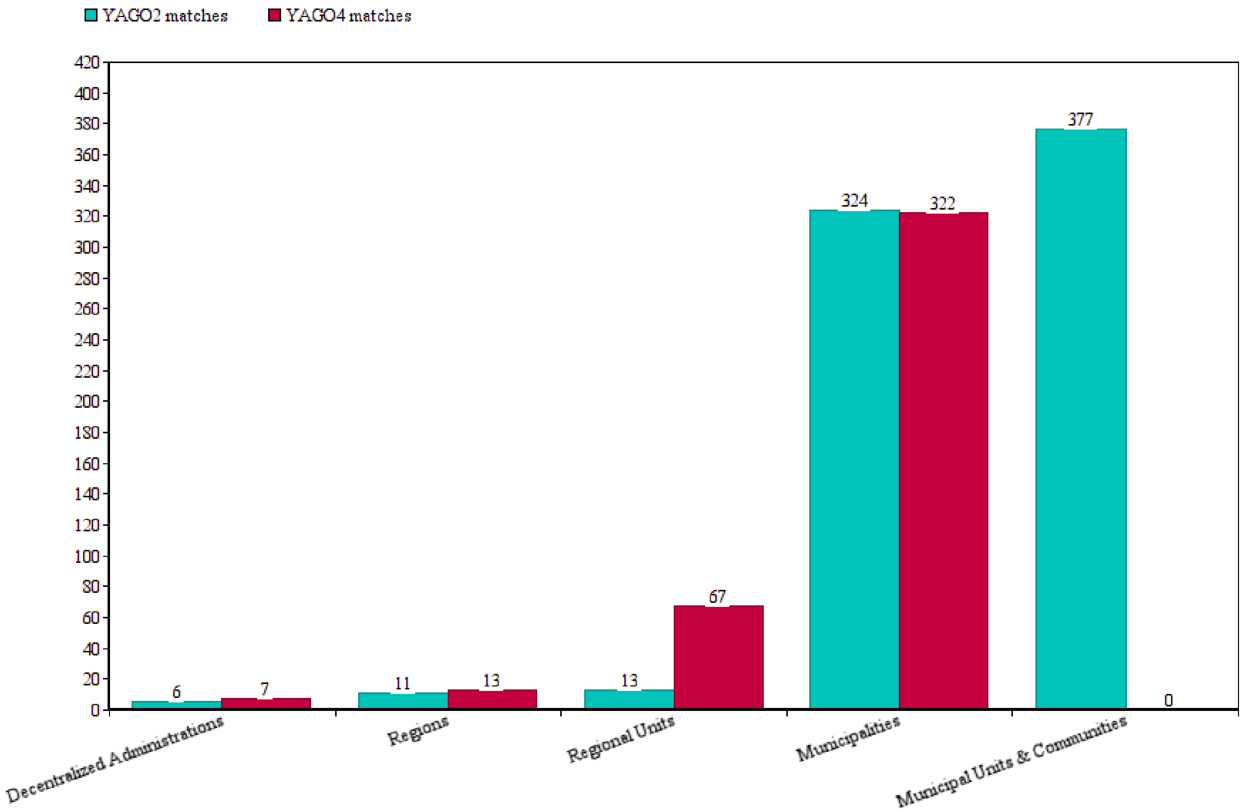


**Figure 5.4: YAGO2 vs YAGO4 matches**

## 5.4 Location Coordinates

Schema.org's design came from Geography Markup Language (GML) [18]. Therefore only the Point, Polygon and LineString geometry types are supported. Furthermore, the geographic coordinates of a place are expressed with the use of schema:geo property, whose values are expected to be either of type GeoCoordinates or GeoShape.

YAGO4 maps coordinates to schema:GeoCoordinates resources. As previously mentioned, schema: GeoCoordinates expresses coordinates as a latitude, longitude pair. On the other hand, schema:GeoShape demonstrates the geographic shape of a place, having properties such as schema:polygon, schema:line, schema:circle etc.

In our GAG dataset, geographic information is expressed in WKT format, using polygons and multipolygons to indicate the area of each location. However, YAGO4 does not support WKT format. For that reason, we tried to make our geographic information compatible to schema.org's design.

More precisely, we converted each WKT Polygon to schema:polygon. Nevertheless, as WKT Multipolygon is not supported in schema.org, we chose to split the geometry to a series of polygons. Finally, the coordinate system used is EPSG:4326.

Here lie some examples of the conversion:

```
POLYGON((
21.71163060727338 40.85096274759539,
21.71163660172085 40.85091782978087,
... ,
21.71163060727338 40.85096274759539))
```

**Figure 5.5: WKT Polygon**

```
"21.71163060727338,40.85096274759539
 21.71163660172085,40.85091782978087
 ...
 21.71163060727338,40.85096274759539"
```

**Figure 5.6: Schema Polygon**

```
MULTIPOLYGON(((
22.17531135346773 37.7954940387722,
... ,
21.89528756628183 21.89528756628183)
( ... ) , ( ... )), ((
23.0612515256197 37.35904845671249
... ,
23.0612515256197 37.35904845671249)))
```

**Figure 5.7: WKT MultiPolygon**

```
"22.17531135346773,37.7954940387722 ...
 21.89528756628183,36.67458427881572 ...
 21.89528756628183,36.67458427881572 ...
 22.17531135346773,37.7954940387722"
" ... "
"23.0612515256197,37.35904845671249 ...
 23.0612515256197,37.35904845671249"
```

**Figure 5.8: Schema MultiPolygon**

The expression of these two geometries using the vocabulary provided by schema.org, allows us to encode their list of coordinates using pre-existing parsers. However, it is not, yet, possible to perform spatial queries using this approach, as GeoSPARQL uses WKT in order to encode geometries.

## 5.5 Summary

In this chapter we handed over our methodology followed in order to extend YAGO4 knowledge graph. In each section an approach to address each deficiency was presented, as well as related examples.

# 6. DEMONSTRATION OF YAGO4 EXTENSION

In this chapter we use GraphDB -an enterprise ready Semantic Graph Database- in order to perform queries and demonstrate our results. GraphDB, also, supports a W3C standard Shapes Constraint Language (SHACL) validation tool for efficient data consistency checking, which confirms that our extension does not violate the constraints of YAGO4.

## 6.1 YAGO4 Extended Entities

During extension phase we tried to match as many as possible unique YAGO4 entities with one of GAG dataset. Those entities were latter extended with the properties presented at section 5.2. GAG entities that were not matched, were newly inserted into YAGO4 knowledge graph.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix schema: <http://schema.org/> .
@prefix exto: <http://kr.di.uoa.gr/yago4-extension/ontology#> .
@prefix extr: <http://kr.di.uoa.gr/yago4-extension/resource/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://yago-knowledge.org/resource/Tinos_Regional_Unit_Q12875777>
rdf:type exto:Kallikratis_Regional_Units_of_Greece ;
schema:alternateName "ΠΕΡΙΦΕΡΕΙΑΚΗ ΕΝΟΤΗΤΑ ΤΗΝΟΥ" ;
exto:has_code "1213"^^<http://www.w3.org/2001/XMLSchema#integer> ;
exto:has_population "8574"^^<http://www.w3.org/2001/XMLSchema#integer> ;
schema:containedIn <http://yago-knowledge.org/resource/South_Aegean> ;
schema:geo extr:Geometry_0.
extr:Geometry_0 schema:polygon "..." .
extr:Geometry_0 schema:polygon "..." .
...
extr:Geometry_0 schema:polygon "..." .
```

**Figure 6.1: Extension of Matched Entity**

```
extr:kallikratis_entity_304
rdf:type exto:Kallikratis_Municipal_Units_of_Greece ;
schema:alternateName "ΔΗΜΟΤΙΚΗ ΕΝΟΤΗΤΑ ΜΕΣΣΑΠΙΩΝ" ;
exto:has_code "914802"^^<http://www.w3.org/2001/XMLSchema#integer> ;
exto:has_population "13756"^^<http://www.w3.org/2001/XMLSchema#integer> ;
schema:containedIn <http://yago-knowledge.org/resource/Dirfys-Messapia> ;
schema:geo extr:Geometry_1.
extr:Geometry_1 schema:polygon "..." .
extr:Geometry_1 schema:polygon "..." .
...
extr:Geometry_1 schema:polygon "..." .
```

**Figure 6.2: Insertion of Unmatched Entity**

One the one hand, figure 6.1 displays the extension of a matched YAGO4 entity. In particular, the YAGO4 entity *Tinos_Regional_Unit_Q12875777* was matched with *1213* GAG entity. Exploiting the information given from this GAG entity in regards with the properties presented in section 5.2 we extended the YAGO4 property with more precise information regarding its population, alternate name, geometry, containment relation and code in GAG dataset.

On the other hand, figure 6.2 displays an unmatched GAG entity that is inserted in YAGO4 knowledge graph. As this entity was not matched, we can assume that it is unknown to YAGO4, therefore the knowledge graph does not hold any information about it. For that reason, -in order to further extend the knowledge graph's content- the entity's properties are appended to YAGO4. The URIs used, are newly constructed in order to indicate their reference to GAG dataset.

## 6.2 SPARQL Queries

We now present the some query results before and after the extension of YAGO4 knowledge graph:

```
PREFIX gag_ontology: <http://geo.linkedopendata.gr/gag/ontology/>
PREFIX schema: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX exto: <http://kr.di.uoa.gr/yago4-extension/ontology#>
select ?pred ?obj where {
    <http://yago-knowledge.org/resource/Corinthia> ?pred ?obj .
}
```

**Figure 6.3: SPARQL Query**

**Table 6.1: Gist of query results before extension**

| ?pred | ?obj |
|---|---|
| rdfs:label | ”Korintia”@eu |
| schema:containedInPlace | http://yago-knowledge.org/resource/ Peloponnese_(region) |
| schema:alternateName | ”Νομός Κορινθίας”@el |
| rdf:type | schema:http://yago-knowledge.org/resource/ Regional_units_of_Greece |
| rdf:type | schema:AdministrativeArea |
| rdf:type | schema:Place |
| rdf:type | schema:Thing |

**Table 6.2: Gist of query results after extension**

| ?pred | ?obj |
|---|---|
| rdfs:label | ”Korintia”@eu |
| schema:containedInPlace | http://yago-knowledge.org/resource/ Peloponnese_(region) |
| schema:alternateName | ”Νομός Κορινθίας”@el |
| schema:alternateName | ”ΠΕΡΙΦΕΡΕΙΑΚΗ ΕΝΟΤΗΤΑ ΚΟΡΙΝΘΙΑΣ” |
| rdf:type | schema:http://yago-knowledge.org/resource/ Regional_units_of_Greece |
| rdf:type | schema:AdministrativeArea |
| rdf:type | exto:Administrative_Units_of_Greece |
| rdf:type | exto:Kallikratis_Regional_Units_of_Greece |
| rdf:type | schema:Place |
| rdf:type | schema:Thing |
| exto:has_code | ”1003”^^$xsd : integer$ |
| exto:has_population | ”143221”^^$xsd : integer$ |
| schema:geo | extr:Geometry_8 |

We clearly distinguish the information augmentation in table 6.2, in contrast with 6.1. In this specific example, *http://yago-knowledge.org/resource/Corinthia*, is now also a type of *Kallikratis_Regional_Units_of_Greece* and *Administrative_Units_of_Greece*. We further notice the extra information about population, as well as alternate names.

## 6.3 Summary

In this chapter we demonstrated the extension of YAGO4 with qualitative geospatial information, extracted from GAG dataset. We used GraphDB, in order to both validate our constraints and perform queries, whose results were also presented. We, also, contrasted our current results with the ones obtained from our previous related work.

# 7. CONCLUSIONS AND FUTURE WORK

In this thesis we presented how we extended the YAGO4 knowledge graph with geospatial information. In order to achieve that we used the Greek Administrative Geography dataset.

Our future work will focus on extending YAGO4 using multiple data sources including well known projects such as GADM and OpenStreetMap, as well as datasets from official sources that contain detailed information about the administrative divisions of other countries. For example, during the extension of YAGO2, Ordnance Survey (OS) and Ordnance Survey Northern Ireland (OSNI) were used in order to obtain information about the administrative division of the United Kingdom.

We will, also, focus on extending YAGO4 with temporal information, as administrative divisions and units change over time. In order to capture these changes we plan to add temporal information about the time an administrative unit was created and seized to be valid. The first use case is going to be Greece, since we are more experienced with its administrative divisions, but afterwards we plan to add such information about other countries as well.

Last but not least, we will dive into finding ways to integrate WKT representation of geometries in YAGO4, in order to be able to express more complex geometries -such as GeometryCollection- and perform spatial queries.

# ABBREVIATIONS - ACRONYMS

| | |
|---|---|
| W3C | World Wide Web Consortium |
| URI | Uniform Resource Identifier |
| RDF | Resource Description Framework |
| RDFs | RDF Schema |
| SPARQL | SPARQL Protocol and RDF Query Language |
| OWL | Web Ontology Language |
| YAGO | Yet Another Great Ontology |
| GAG | Greek Administrative Geography |
| WKT | Well Known Text |
| GML | Geography Markup Language |

# BIBLIOGRAPHY

[1] James Hendler Tim Berners-Lee and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[2] James Hendler Tim Berners-Lee and Ora Lassila. The World Wide Web. *Computer Network and ISDN Systems*, 25:454–459, 1991.

[3] Resource Description Framework (RDF). `https://www.w3.org/RDF/`, 2014.

[4] SPARQL. `https://www.w3.org/TR/rdf-sparql-query/`.

[5] Web Ontology Language (OWL). `https://www.w3.org/OWL/`, 2012.

[6] Shapes Constraint Language (SHACL). `https://www.w3.org/TR/2017/REC-shacl-20170720/`, W3C Recommendation 20 July 2017.

[7] Shema.org. `https://schema.org`, April 2015.

[8] YAGO: A High-Quality Knowledge Base. `https://yago-knowledge.org`.

[9] Georgi Kobilarov Jens Lehmann Richard Cyganiak Sören Auer, Christian Bizer and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors. *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, pages 722–735, Springer, 2007.

[10] R. Navigli and S. Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, (193):217–250, Elsevier, 2012.

[11] Bryan Kisiel1 Burr Settles Estevam R. Hruschka Jr. Andrew Carlson, Justin Betteridge and Tom M. Mitchell. Toward an Architecture for Never-Ending Language Learning. *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA*, Elsevier, 2010.

[12] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 10(57):78–85, September 2014.

[13] Praveen Paritosh Tim Sturge Kurt Bollacker, Colin Evans and JamieTaylor. Freebase:Acollaboratively created graph database for structuring human knowledge. *In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08*, pages 1247–1250, New York, NY, USA, 2008. ACM.

[14] Klaus Berberich Johannes Hoffart, Fabian M. Suchanek and Gerhard Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence(2013) and World Wide Web (WWW 2011)*, 2011.

[15] Johannes Hoffart Joanna "Asia" Biega Erdal Kuzey Gerhard Weikum Thomas Rebele, Fabian M. Suchanek. YAGO3: YAGO: a multilingual knowledge base from Wikipedia Wordnet and Geonames. *International Semantic Web Conference (ISWC 2016), Conference on Innovative Data Systems Research (CIDR 2015) and Asia Pacific Web Conference (APWEB 2014)*, 2015.

[16] Gerhard Weikum Thomas Pellissier-Tanon and Fabian Suchanek. YAGO 4: A Reason-able Knowledge Base. *Extended Semantic Web Conference (ESWC)*, 2020.

[17] Georgios Mandilaras Nikolaos Karalis and Manolis Koubarakis. Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge. *International Semantic Web Conference*, 2019.

[18] Raimundo F. Dos Santos and Chang-Tien Lu. Geography Markup Language. Springer, 2017.