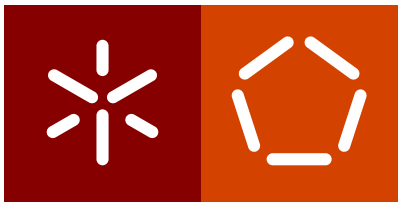
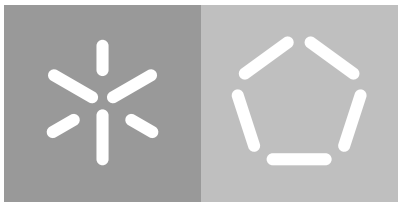


**Universidade do Minho**
Escola de Engenharia
Departamento de Informática

João Costeira Faria Gomes

# Exploration of documents concerning Foundlings in Fafe along XIX Century

September 2022



**Universidade do Minho**
Escola de Engenharia
Departamento de Informática

João Costeira Faria Gomes

# Exploration of documents concerning Foundlings in Fafe along XIX Century

Master dissertation
Integrated Master's in Informatics Engineering

Orientador
**Pedro Rangel Henriques**
Supervisores no local de trabalho
**Cristiana Araújo, Mónica Guimarães**

September 2022

## STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

João Costeira Faria Gomes

_____

# AGRADECIMENTOS

O desenvolvimento deste projeto foi um processo longo, onde muitos obstáculos tiveram de ser enfrentados. De forma a conseguir chegar à meta, o suporte de um conjunto de pessoas foi essencial, tanto no ponto de vista motivacional como numa perspetiva mais técnica. O objetivo desta pequena secção é destacar e agradecer essas pessoas, que diretamente ou indiretamente, tiveram um papel fulcral neste projeto.

Inicialmente, o apoio incondicional da minha família e amigos, que sempre me ajudaram ao longo deste projeto e todo o percurso académico em geral. Estas pessoas foram uma das principais forças motivadoras, a apoiar nos melhores e piores momentos. Este apoio possuiu um papel preponderante na continuação deste projeto, em acreditar que seria capaz de ultrapassar os maiores obstáculos enfrentados, que em alguns momentos, pareciam impossíveis e desistir seria a opção mais fácil de seguir.

O segundo grupo de pessoas que devo destacar são os meus orientadores, Pedro Rangel Henriques e Cristiana Araújo. O seu apoio e acompanhamento semanal, incluindo disponibilidade adicional fora de horas de trabalho, foi essencial num ponto de vista mais técnico, fornecendo aconselhamento sobre as decisões e etapas enfrentadas ao longo do desenvolvimento do projeto.

De forma análoga, tenho de agradecer à Mónica Guimarães, representante da equipa do Arquivo Municipal de Fafe, que estabeleceu a ponte de comunicação com o arquivo. A sua ajuda foi essencial, fornecendo apoio técnico não numa perspetiva informática, mas numa perspetiva de arquivista e de utilizador final da aplicação. O seu conhecimento na área do domínio do projeto, facilitou a compreensão de qualquer dúvida existente e fornecimento da base documental necessária.

Por último, um pequeno destaque ao Jean-Baptiste Lamy, elemento da equipa de desenvolvimento da tecnologia OWLReady. O seu apoio e recomendações técnicas foram cruciais na ligação entre a componente ontológica e o sistema global.

## ABSTRACT

The abandonment of children and newborns is a problem in our society.

In the last few decades, the introduction of contraceptive methods, the development of social programs and family planning were fundamental to control undesirable pregnancies and support families in need. But these developments were not enough to solve the abandonment epidemic.

The anonymous abandonment has a dangerous aspect. In order to preserve the family identity, a child is usually left in a public place at night. Since children and newborns are one of the most vulnerable groups in our society, the time between the abandonment and the assistance of the child is potentially deadly.

The establishment of public institutions in the past, such as the Foundling Wheel, was extremely important as a strategy to save lives. These institutions supported the abandoned children, while simultaneously providing a safer child abandonment process, without compromising the anonymity of the family.

The focus of the Master's Project discussed in this dissertation is the analysis and processing of nineteenth century documents, concerning the Foundling Wheel of Fafe.

The analysis over the sample documents is the initial step in the development of an ontology. The ontology has a fundamental role in the organization and structure of the information contained in these historical documents. The identification of concepts and the relationships between them, culminates in a structured knowledge repository. Other important component is the development of a digital platform, where users are able to access the content stored in the knowledge repository and explore the digital archive, which incorporates the digitized version of documents and books from these historical institutions.

The development of this project is important for some reasons. Directly, the implementation of a knowledge repository and a digital platform preserves information. These documents are mostly unique records and due to their age and advanced state of degradation, the substitution of the physical by digital access reduces the wear and tear associated to each consultation. Additionally, the digital archive facilitates the dissemination of valuable information. Research groups or the general public are able to use the platform as a tool to discover the past, by performing biographic, cultural or socio-economic studies over documents dated to the ninetieth century.



**Keywords:** Foundling Wheel, Ontology, Knowledge Repository, Digital Platform

# RESUMO

O abandono de crianças e de recém-nascidos é um flagelo da sociedade.

Nas últimas décadas, a introdução de métodos contraceptivos e de programas sociais foram essenciais para o desenvolvimento do planeamento familiar. Apesar destes avanços, estes programas não solucionaram a problemática do abandono de crianças e recém-nascidos. Problemas socioeconómicos são o principal factor que explica o abandono.

O processo de abandono de crianças possui uma agravante perigosa. De forma a proteger a identidade da família, este processo ocorre normalmente em locais públicos e durante a noite. Como crianças e recém-nascidos constituem um dos grupos mais vulneráveis da sociedade, o tempo entre o abandono da criança e seu salvamento, pode ser demasiado longo e fatal.

A casa da roda foi uma instituição introduzida de forma a tornar o processo de abandono anónimo mais seguro.

O foco do Projeto de Mestrado discutido nesta dissertação é a análise e tratamento de documentos do século XIX, relativos à Casa da Roda de Fafe preservados no Arquivo Municipal de Fafe.

A análise documental representa o ponto de partida do processo de desenvolvimento de uma ontologia. A ontologia possui um papel fundamental na organização e estruturação da informação contida nos documentos históricos. O processo de desenvolvimento de uma base de conhecimento consiste na identificação de conceitos e relações existentes nos documentos.

Outra componente fundamental deste projecto é o desenvolvimento de uma plataforma digital, que permite utilizadores acederem à base de conhecimento desenvolvida. Os utilizadores podem pesquisar, explorar e adicionar informação à base de conhecimento.

O desenvolvimento deste projecto possui importância. De forma imediata, a implementação de uma plataforma digital permite salvaguardar e preservar informação contida nos documentos. Estes documentos são os únicos registos existentes com esse conteúdo e muitos encontram-se num estado avançado de degradação. A substituição de acessos físicos por acessos digitais reduz o desgaste associado a cada consulta.

O desenvolvimento da plataforma digital permite disseminar a informação contida na base documental. Investigadores ou o público em geral podem utilizar esta ferramenta com o objectivo de realizar estudos biográficos, culturais e sociais sobre este arquivo histórico.

**Palavras-Chave:** Casa da Roda, Exposto, Ontologia, Base de Conhecimento,Plataforma Digital

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF LISTINGS

# ACRONYMS

**C**

**CRUD** Create Read Update Delete.

**CSV** Comma-Separated Values.

**E**

**ERD** Entity Relation Diagram.

**F**

**FOAF** Friend Of A Friend.

**J**

**JSON** JavaScript Object Notation.

**JSONL** JSON Lines.

**M**

**MS** Microsoft.

**MVC** Model View Controller.

**N**

**NER** Named Entity Recognition.

**NLP** Natural Language Processing.

**O**

**OWL** Ontology Web Language.

**P**

**POS** Part Of Speech.

**R**

**RDF** Resource Description Framework.

**RDFS** Resource Description Framework Schema.

**REL** Relation Extraction.

**S**

**SPARQL** SPARQL Protocol and RDF Query Language.

**SQL** Structured Query Language.

**U**

**URI** Uniform Resource Identifiers.

**URL** Uniform Resource Locator.

**W**

**W3C** World Wide Web Consortium.

**X**

**XML** Extensible Markup Language.

# 1

# INTRODUCTION

This chapter introduces the motivation and the main objectives to be accomplished with the project development.

Towards the end of this chapter, the document structure and the research hypothesis are presented.

## 1.1 MOTIVATION

The abandonment of children and newborns is a social problematic that occurs all over the world, across cultures and historical context.

Throughout different historical periods, societies introduced sensible measures to support vulnerable children and reduce the number of abandonments. These actions were extremely important as a life saving measure, since evidences suggest a correlation between anonymous child abandonment and infanticide. A substantial group of children die during the abandonment process.

This project focuses on the archived documents from the Foundling Wheel of Fafe, institution introduced to support exposed children. An important note is when foundling wheels were replaced officially by hospices and the Municipal Archive of Fafe merge its local registry books with those produced in similar institutions acting in its neighborhood.

There is a vast of research work concerning the Portuguese foundling wheel (da Fonte, 2004; Panter-Brick et al., 2000; Brettell and Feijó, 1991). But this project approach is the development of a digital platform to disseminate the information contained in these historical documents, in opposition to a profound analysis over the foundling domain.

The development of the digital archive is important to preserver historical information, from documents in advanced state of degradation. Additionally, the integration of documents, ontology and knowledge repository in a digital platform, offers the dissemination of the archive.

## 1.2 OBJECTIVES

The main objective is the development of a digital platform supported by a knowledge repository.

The ontology and the information extraction component were identified as the two main aspects of this project. An analysis of sample documents is required to identify the concepts and the respective relationships, the backbone of the knowledge repository.

In order to successfully accomplish the main objective of this project, a set of sub objectives was identified:

- Development of an ontology, used to define the relationships between all the different concepts contained in the documents,
- Development of a knowledge repository, used to store all the different instances of extracted entities and respective relations,
- Development of a web platform, where users are able to access the digital documents, the knowledge repository, search according to different criteria and insert new information.

## 1.3 RESEARCH APPROACH

To accomplish the main objectives of this project, the development phase was organized according to the following steps:

- Bibliographic study of cutting-edge technologies used to complete each goal of this project. More specifically, the study of semantic web, natural languge processing, web development frameworks and storage solutions.
- Analysis of sample documents. This step is fundamental in the organization of the data in two perspectives. Separate the different documents in different groups, for instance, documents concerning foundlings from the institutional management records. The second role is the identification of concepts, relations, entities and attributes.
- Development of an ontology used to describe the knowledge, by structuring the information according to different concepts and respective relationships. In this project, Foundling and Institution are the main concepts.
- Development of a storage solution as a main knowledge repository. The database stores the information, modelled accordingly to the ontology. The concepts described in the ontology are converted into entities in the database. Since the concepts are

entities, the relations between concepts in the ontology are expressed as relations between entities in the database. The attributes of each concept are defined by the columns of their respective table in database.

- Development of the digital archive[1], the web platform responsible for establishing the connection between the repository and the end user.

After the last, if the result attained or the performance do not comply with the desired quality, the previous steps should be repeat until a satisfactory solution is reached.

## 1.4 Research Hypothesis

From the identification of concepts and relations in the Foundling Wheel archive it is possible to develop a digital platform supported by a knowledge repository.

## 1.5 Document Structure

The structure of this document follows the development process of the entire project, describing each step in a separate chapter.

In Chapter 2, the child abandonment problematic is introduced, according to the historical and geographical context, by analysing both former and current approaches. After the historical contextualization, an overview of the current debate towards the reintroduction of anonymous abandonment mechanisms. This is an active debate topic, since multiple countries are considering the adoption of measures to combat the raising abandonment number. Opposite groups discuss the legality, human rights violations and the effectiveness of these measures. This chapter is concluded with references to similar projects currently under development.

Chapter 3 provides the initial analysis over the sample documents provided. The content of these documents is analysed in two main perspectives, organizing documents according to their topic and an initial identification of concepts and relations. This step is the foundation for the development of the ontology and the knowledge domain. This chapter is concluded with an overview of the system architecture.

In Chapter 4, the proposed solution is described and the structure of this chapter chronologically follows the development of this project. Initially, the extraction of information from documents is explored and this chapter culminates in the integration of the processing pipeline and the knowledge repository. While each step is being described, the different concepts, development decisions and main obstacles faced are accordingly introduced.

1 Similar project supported by a knowledge repository. Official page available at: https://epl.di.uminho.pt/~gepl/GEPL_DS/UEF/, Accessed in October 2020.

Chapter 5 focuses on the web application. The system architecture is explored and this chapter is concluded with a navigational example.

To conclude, Chapter 6 provides an overview of the entire project, result analysis, a discussion on the state of the project and future work.

# 2

# BACKGROUND ON CHILD ABANDONMENT

The goal of this chapter is to contextualize the anonymous child abandonment, by analysing different social programs and strategies introduced to combat this problematic.

The analysis is organized according to historical periods. Initially focuses on the current debate towards the reintroduction of anonymous child abandonment mechanisms in different parts of the world. Then contextualization is proceeded by analysing the foundling wheel, a historical institution banned over a century ago.

The historical contextualization is important to comprehend the effectiveness of social programs and to introduce the domain of this project, since the sources are archived nineteen century documents from foundling wheels.

Towards the end of this chapter, similar projects currently under development are presented.

## 2.1 PRESENT

Child abandonment is a problematic that occurs throughout history and across different societies around the world.

In order to combat the abandonment problematic and support one of the most vulnerable groups, the study and introduction of public institutions to raise children, otherwise left behind on the streets, and social programs to support families in need, were fundamental measures to reduce the high child mortality rate, consequent of the abandonment.

More recently, developments in a multitude of fields was extremely important to further reduce the child abandonment problematic.

The introduction of preventive measures, from the mass distribution of birth control methods to family planning, in general reduces the number of unplanned pregnancies and consequent number of abandonments. Simultaneously, the development of women's rights and legalization of abortions[1], present an alternative to control unwanted pregnancies. Abortions are illegal in 26 countries. In the remaining countries, the legality of abortions

1 Additional details on abortion, according to country and respective gestational limits, available at: https://reproductiverights.org/worldabortionlaws, Accessed in December of 2021

varies according to different contexts, from only legal when woman's life is at risk to fully legal abortion upon request, within gestational limits.

Further development of social programs constitute a support system to assist vulnerable families. The decentralization of social-economic programs and focus on family-based support, keeps families together, instead of forcing families to abandon their children due to a lack of resources.

Social factors and changes in attitudes towards sensitive topics, including illegitimacy, divorce or single parenthood, positively impacted the reduction of social and peer pressure towards mothers, enabler factors for the abandonment of children.

Although these developments helped families at risk, the abandonment problematic is not solved, cases of child abandonment happen till this day[2].

Some countries are debating and reintroducing strategies to combat the anonymous child abandonment. In order to contextualize this problematic, in general, the anonymous child abandonment is illegal. Children rights declare that every person has the right of knowing their origins and establish relationships with their family.

Regardless of potential criminal repercussions, children are abandoned in public places and this process is extremely dangerous in the perspective of potential infanticides. Therefore, the introduction of safer alternatives for the fully anonymous child abandonment are currently being studied.

## 2.2 BABY BOXES

Baby boxes, also referred as baby hatches or angel's cradle, are modern mechanisms that preserve the anonymity of the family, while providing a safer abandonment process Cochrane and Ming (2013).

This system consists of a box, usually fixed in one position similar to a mailbox and located next to emergency services, for instance, next to a hospital or maternity.

The process of abandoning a newborn on a baby box consists in opening the door and placing them inside. By closing the door, the box is automatically locked, in order to protect the foundling from any outside element.

From the inside perspective, the baby box is safe and offers all the survival conditions, with features including climate control and bedding. The combination of these conditions results in a safe environment, providing the survival conditions for multiple hours, until the medical support arrives.

2 Newborn abandoned in a trash container in Lisbon, November 5th 2019. The exposed child was found in critical conditions by a homeless man. Full news report available in the following link: https://www.publico.pt/2019/11/05/sociedade/noticia/recemnascido-encontrado-caixote-lixo-lisboa-1892658, Accessed in January 2021.

In terms of technology, the box contains cameras and sensors used to alert the presence of a new arrival. These technologies are only used as an alert system, they are not used as a mechanism to track down the perpetrators of the abandonment.

The authorities are immediately notified every time a newborn is placed inside the baby box, in order to assist them as soon as possible.

Although the baby box offers all the survival conditions for a long period of time, the combination of the alert system with the closed proximity to the public emergency services results, on average, a five minutes elapse time between the abandonment and human support.

### 2.2.1 *Approaches According to Geo Regions*

This section provides examples of countries that recently introduced the anonymous child abandonment mechanism baby box. This analysis focuses on the motivations that led to the introduction.

#### 2.2.1.1 *Europe*

In Europe[3], approximately two hundred baby boxes were indroduced in the last decade. Germany, Switzerland, Poland and Czech Republic are few examples of those European countries. The goal is to combat the child abandonment epidemic.

Since the year 2000, Germany introduced approximately eighty boxes across their territory. The main factor that led to the introduction of baby boxes in this country was the dramatic number of infanticides, resulting from the child abandonment epidemic. For instance, in the year 1999, three out of five abandoned babies died during the abandonment process or from collateral damages.

The introduction of baby boxes is a measure to prevent the high number of infanticides during the abandonment process.

#### 2.2.1.2 *China*

Besides the European context, China[4] is another country that recently introduced the baby box. The main factor that led to the introduction was a concern by the authorities towards the alarming number of abandoned babies.

In China, per year around a thousand newborns are abandoned and two out of three babies die during the abandonment process.

3 News reports on the widespread of baby boxes across Europe, available in the following link: https://www.theguardian.com/world/2012/jun/10/unitednations-europe-news, Accessed in December of 2020

4 Details on the Chinese scenario available at: https://www.bbc.com/news/world-asia-china-26219171, Accessed in December of 2020

According to the official data, the vast majority of abandoned babies have health problems. Parents worried by the potential financial obligations, find the abandonment as the only viable option.

The Chinese context has a distinguish factor, since the population control policies introduced may justify the disproportional gender bias found within the foundling population.

In China, between the year 1976 and the year 2015, the one child policy was applicable[5]. The introduction of an upper limit on the number of descendants per couple, led to a gender discrimination during the abandonment process. Families preferentially kept a boy as their only child. The introduction of policies as the one child policy, may justify the disproportional number of females found within the Chinese foundling population.

An important note is that the introduction of child abandonment mechanisms, for instance the baby box in China, does not equate to the legalization of anonymous abandonment. The baby boxes are a last resort solution to combat the raising number of infanticides, subsequent to the abandonment.

### 2.2.2 *Discussion on the Introduction of Baby Boxes*

The introduction of baby boxes is extremely controversial. Groups with opposing views, including human rights activist and politicians, discuss the impact of the reintroduction and legalization of anonymous child abandonment mechanisms.

#### 2.2.2.1 *Against the Introduction of Baby Boxes*

The United Nations (UN) are against the introduction of baby boxes Ramesh (2012). One of the main factors that justifies their stance is the violation of key points of the children rights convention. More specifically, the fundamental right of children knowing their ancestry and the establishment of connections with their relatives.

According to Maria Herczog, child psychologist and member of the UN committee, the introduction of baby boxes represent an incentive to hidden pregnancies and childbirths without any medical support Ramesh (2012). There is no correlation between baby boxes and a decrease on the number of infanticides. The solution is to support mothers under these circumstances.

Kevin Browne, of the Centre for Forensic and Family Psychology of University of Nottingham, completed a study about the introduction of baby boxes during a two-year period Ramesh (2012); Evans (2012).

His study indicates that out of the twenty seven countries part of the European Union, eleven of these countries already introduced the baby box.

5 Additional details on the one child policy available at: https://en.wikipedia.org/wiki/One-child_policy, Accessed in December 2020

One of the biggest issues pointed in his study is a significant number of the abandonments are committed by a male figure. The nonexistence of medical nor psychological support and the lack of an official abandonment process, can be abused by family members and pimps. Without any official consultation nor consent from the mother's part, the anonymity of the baby box offers a legal breach, abandonment against the mother's will.

Another topic pointed in his study is the integration of the baby boxes within the legal framework. In countries, including Germany, every single citizen has the right of knowing their origins and establish relations with their family. The baby box breaks both of these laws. German ministers indicates that the solution is a legal reform, by including an exception to the confidential abandonment.

Michalle Oberman, law professor at the University of Santa Clara, California, indicates that vulnerable groups, such as young or adolescent mothers, are extremely susceptible. The baby boxes may be interpreted as their only viable option Baker (2019).

The combination of social shame towards teen pregnancy and the introduction of baby boxes, present a dangerous alternative to vulnerable young mothers, as an incentive hidden pregnancies and childbirth without any medical support.

#### 2.2.2.2 *In Favor of the Introduction of Baby Boxes*

Supporters argue that the introduction of the baby box is important, as a mechanism that saves lives. For desperate mothers, where the abandonment is the only viable option, baby boxes are a safe alternative, much safer than abandoning a newborn in a public place.

Gabriele Stangl, from the Hospital of Waldfriede in Berlin, claims that baby boxes save lives. Saving lives is more important than tracing the origin of the baby for a potential reconciliation with their family Evans (2012).

The Baby Boxes are extremely safe, in Berlin, the baby box is placed next to the maternity. This proximity reduces the elapse time between the abandonment and the respective assistance by the medical staff.

After the medical support, the baby enters the national adoptive system. The adoptions are final. In case of regret, the reconciliation with the biological family is only permitted before the completion of the adoptive process.

In the last decade, forty two babies were abandoned at the Hamburg Baby Box. Within this group, seventeen mothers contacted the institution and fourteen of them reconciled with their child.

The Danish center for social science research (VIVE)[6], after observing the introduction of baby boxes across Europe, conducted a study to analyze the effectiveness of the anonymous abandonment mechanisms and determine if Denmark should follow the footsteps of those European countries. Evidences suggest that baby boxes reduce the numbers of

6 Official VIVE webpage available at: https://www.vive.dk/en/about-vive/, Accessed in January 2021

infanticides. For instance, according to the analyst Marie Jakobsen, since the year 2000, when Germany introduced the baby box, there were no infanticides registered in consequence of an anonymous abandonment Baker (2019).

## 2.3 SAFE HAVEN

The safe haven law is another approach to combat the anonymous child abandonment problematic Baker (2019).

In the United States, the abandonment of children in a public place is illegal. In order to combat this problematic, a new exception to the law was introduced, usually referred as safe haven.

This law was introduced in the year 1999 in the state of Texas. According to this law, it is possible to abandon a child in a restrictive group of public institutions, including hospitals, police stations or generally next to any public emergency service providers.

In these restrictive areas, a child is safely abandoned, following a no question asked policy towards the perpetrator. Currently, 49 American states implemented the safe haven law.

In the state of Texas, on average one hundred babies are abandoned in these restrictive areas per year. Since these solutions are very effective, studies towards the expansion of anonymous child abandonment mechanism are currently being analysed.

## 2.4 HISTORICAL CONTEXT

After introducing the current debate towards the adoption of laws and mechanics to handle the increasing abandonment numbers, this section focuses on the extinct and historical foundling wheel.

The foundling wheel or foundling hospital are terms used to describe institutions that provided an anonymous and secure abandonment process.

The abandonment process consists in placing the child on a wheel. After executing a rotative motion, the child is left inside the institution, protected from the outside elements. Next to the wheel, usually a bell is placed in order to notify the presence of a new arrival.

The first foundling wheels were introduced in the XIII century in Europe. Usually, the wheel is an integral part of public institutions or catholic run.

The main reasons behind the child abandonment in those historical periods were social-economic problems. Poverty and family without resources to support their children, before the existence of the foundling wheel, simply left them on the streets or doorsteps of affluent people.

Another factor was strong religious believes. God had a protective role of the abandoned and exposed children. A common practice was abandoning the weaker children and keep the strongest in the family.

### 2.4.1 *Foundling Wheel in Portugal*

#### 2.4.1.1 *Introduction of the Foundling Wheel*

In the first half of the XVIII century, Portugal was suffering from an alarming number of child abandonment and infanticides. Families saw the child abandonment as the only viable option. Social programs were extremely scarce and insufficient to tackle the magnitude of this problematic.

In order to face this dramatic situation, in 10 of May of 1783, Pina Manique declared the establishment of the foundling wheel and support of the exposed children in Portugal Brettell and Feijó (1991).

#### 2.4.1.2 *Life on The Institution*

Right after the arrival, the foundling wheel had the role of assisting and supporting the abandoned child ion (1970). A common practice was giving the abandoned child to a wet nurse, which breastfeeds and raises the foundling. These nurses received money in exchange for their caretaker service.

The development process of each foundling from their arrival to their adulthood and respective independence varied.

Some foundlings spent the rest of their developing years with their nurse. While in opposition, some foundlings after finishing their breastfeeding year or the contractual obligations with a nurse, returned to the institution. Within the latter group, some foundlings found a job and their employer took care of them. Other foundlings, according to their age and gender, learnt a profession or domestic work. Unfortunately, foundlings that were not able to learn or find a profession, usually children with disabilities, some eventually ended up going back to the streets.

#### 2.4.1.3 *Problems in the Institution*

The foundling wheel was introduced to control the alarming number of infanticides and improve the quality of life of the abandoned children. But these institutions originated another set of problems.

The child mortality rate in these institutions was extremely high. Excessive administrative costs, lack of wet nurses willing to breastfeed newborns and nurses to raise abandoned children, drastically diminished the living conditions in the foundling wheel.

Cases of multiple children sleeping in the same bed, delayed payments and health problems were common practices in these institutions Paulino (2017). For example, the *Real Casa dos Expostos*, in Lisbon, at one point only had thirty one nurses available to raise one hundred and fifty two children.

All of these problems amplified the voices of critics. António Henriques Secco was extremely critical, claiming that the foundling wheel did not improve the life of the abandoned children.

#### 2.4.1.4 *Termination of the Foundling Wheel*

During the decades of 1860 and 1870, a group of strategical changes in the governmental funds and legislation culminated in the termination of the foundling wheel in Portugal. In the governmental intervention in 1862 and the decree of 21 of November of 1867, the abolished of the Foundling Wheel was declared and these institutions were replaced by Hospices Paulino (2017).

The admission process in the hospices occurred during the day and required the identification of the parents. Admission had to be justified and exclusively accepted under certain circumstances, such as absent or death of the parents and for cases where the identity of the parents was truly unknown.

In order to combat the anonymous abandonment, rewards were given to whom disclosed information to the authorities, helping the identification process of the perpetrators. Twenty thousand *reis* were given in exchange for information.

Other strategy introduced was a substitution of government funds to nurses by directly support families in need. For example, the introduction of incentives to reconcile families by giving a thousand *reis* salary, during a twelve month period, if the biological family retrieved the abandoned child from the institution. These family-centered policies were cost-effective. Paying nurses for long periods of time was more expensive than simply supporting families for one year.

The remaining children were institutionalized, instead of searching for a nurse willing to raise them. In the institution, foundlings studied in order to learn a profession. After the age of twelve, some foundlings initialized their professional career, others kept studying. When foundlings reached their eighteenth birthday, they became fully independent adults.

#### 2.4.1.5 *Successful Termination of the Foundling Wheel*

Between 1871 and 1910, the number of abandoned children was reduced to ten percent of the register foundlings from the period between 1850 and 1870, according to the Foundling Wheel of Lisbon (Paulino, 2017, p. 15). The family based programs reduced the number of foundlings and kept families together.

## 2.5 SIMILAR PROJECTS

The project titled *Voices Through Time: The Story of Care* is an example of a similar project currently in development.

This project consists of the digitization of the foundling hospital achieve, institution originally located in the city of London and established in the XVIII century. This archive preserves all the surviving documents from this institution, currently stored in the London Metropolitan Archive.

The Foundling Hospital was established by Thomas Coram, to help families without resources, forced to abandon their children.

### 2.5.1 *Foundling Hospital*

Thomas Coram was born in 1668 in the city of Lyme Regis in England fou (2018). After returning from the United States of America, Coram was shocked by the state of the city of London, with an alarming number of abandoned children, left on the streets without any support. In order to combat this problematic, Coram promoted the establishment of the first institution for the abandoned children in England.

Initially, the support was denied by the king and influential people. Social stigmas towards illegitimacy were the main reason behind the lack of support.

Throughout a decade of raising awareness, slowly Coram started to obtain support from influential women, culminating in the establishment of the foundling hospital in the year of 1739, with the support of the king George II.

After the introduction of the 1948 child act, the governmental funds were directed to support families, culminating in foundlings reunited with their family. In 1950 the foundling hospital closed.

The Coram institution is currently active, focusing on education, supporting families and assist the adoptive process. More than twenty five thousand children were saved by this institution cor.

### 2.5.2 *Foundling Museum*

The Foundling Museum[7] was established in order to disseminate the history of the Foundling Hospital.

7 Official museum webpage: https://foundlingmuseum.org.uk/about/the-museum/, Accessed in October 2020

The museum has a fundamental role in reuniting families[8]. People supported by the Coram institution or their family members are able to contact the foundling museum for reconciliation purposes, by accessing the archived private and personal information.

### 2.5.3 *London Metropolitan Archives*

The London Metropolitan Archive (LMA)[9] is the main archive of this city. Extremely rare documents are preserved in this institution. The oldest document is traced back to the year 1067.

The archive contains more than three million documents, including books, maps, movies and other historical resources. These documents are stored in one hundred kilometers of rows, from which two hundred and forty meters are exclusively dedicated to preserve the foundling hospital archive.

All documents since the implementation of the foundling hospital are stored in the LMA, including registration, inspection, petitions or letters from their mothers.

Some documents collections are already digitized and are accessible online. An important note is the access to some of these collections are restricted, for instance, only available to academics and researchers.

### 2.5.4 *Program Voices Through Time: The Story of Care*

The foundling hospital archive is one of the most requested collection of the LMA.

Currently, these documents are undergoing a digitization process, part of the project *Voices Through Time: the Story of Care*[10], a 4-year process backed up by the National Lottery Heritage Fund[11], which allocated 1.26 million pounds to support this project.

This project consists in the digitization of 112,000 documents between the period of 1739 and 1900, transcribing the digitized images and the development of the website, estimated to be launched in the beginning of 2021.

8 More information on the reconciliation program and birth records available at: https://www.coram.org.uk/adoption/your-birth-records, Accessed in October 2020

9 Details on the history of the foundling archive available at: https://www.cityoflondon.gov.uk/things-to-do/history-and-heritage/london-metropolitan-archives/collections/family-history-at-lma, Accessed in October

10 Additional details on the project Voices Through Time: the Story of Care, available at: https://www.coram.org.uk/news/coram-tell-story-care-through-digital-project-funded-%C2%A3126m-national-lottery, Accessed in November 2020

11 Further details on the digitization of the archive and the allocation of funds available at: https://www.heritagefund.org.uk/news/uks-oldest-childrens-charity-tell-story-care, Accessed in November 2020

# 3

# PROPOSED APPROACH

The goal of this chapter is to introduce the foundation of the proposed solution. Initially, the analysis focuses on a set of sample documents. This analysis has two main objectives, the classification of documents according to their content and the identification of the main concepts and respective relations.

The identification of entities and a discussion on possible relations, culminates in the original iteration of the ontology, the structure of the knowledge repository.

This chapter is concluded with the analysis of the global architecture of the application.

## 3.1 DOCUMENT ANALYSIS

The groundwork for the entire project is an analysis of sample documents. These documents are concrete examples used in the initial identification of concepts and relations.

The domain of this project is centralized in one particular topic, the preserved documents concerning foundling wheels. These documents are dated to the nineteenth century and originally from institutions located in the Northern Region of Portugal.

The aggregation of documents occurred during the process of ceasing the foundling wheels in Portugal. The assistance was centralized and these institutions were replaced by Hospices. As a result, the Municipal Archive of *Fafe* preserves documents from other municipalities, including *Cabeceiras de Bastos* and *Celorico de Bastos*.

The individual registries are compiled into books. These books aggregate different types of documents and sort them by date.

Figure 1 displays a book of documents. The cover contains important information, including the name of the institution, identifier of the book, the main topic and time period. In this particular example, the book contains new entries of abandoned children in the Foundling Wheel of *Fafe*, between 6 of June of 1879 and 19 of April of 1880.

Figure 1: Book of Documents

One of the most important set of documents are the records of different life events of each foundling, from the registration in the institution, foster care, to the departure from the institution. Furthermore, the management records of the institutions are preserved, for example vaccination registries and payments to nurses.

A special type of documents are the *sinais*, or *signals* in English. A signal is a term used to describe an object or a document left with the foundling during the abandonment process. These documents are extremely important, since they may be the only identifiable link, for potential reconciliation with the biological family.

As an example, Figure 2 contains two documents. The main document is the record of a new foundling in the institution. Additionally, a complementary note describing the *layette*, the original belongings of that particular foundling.

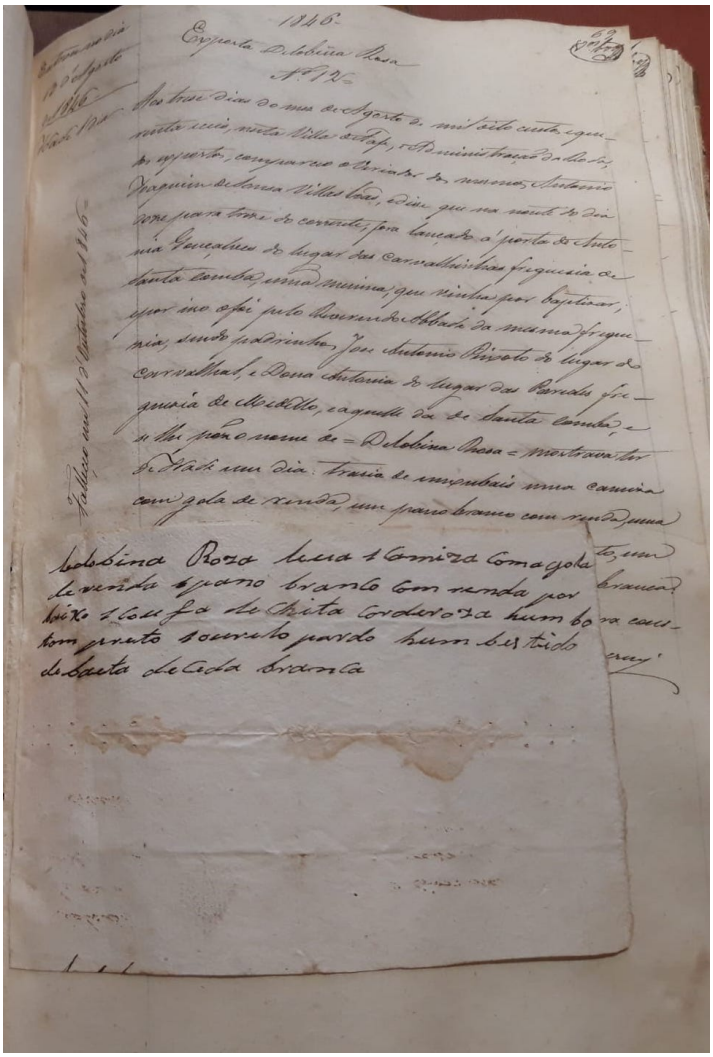


Figure 2: Registration Document and Layette Description

Table 1 compiles the main types of documents preserved. The classification proposed is based on the main topic of each document:

| Document | Description |
|---|---|
| Entrance Register | New foundling record. Contains information such as the date of registration, clothes, baptism, name, godfathers,layette, description of signals (notes, jewelry , religious symbols) |
| Departure Register | Books concerning foundlings leaving institution |
| Medical Records | Registries of medical support given to each foundling |
| Vaccination Records | Contains information such as registration number, name, age and date of vaccination |
| Register of Expenses | Books containing the expenses and budget management of the foundling wheel |
| Record of Nurses | Registries of new nurses and wet-nurses that raise foundlings |
| Payment to Nurses | Books containing the name of the nurse, respective foundling and payment information |
| Foundlings given to Nurses | Books containing the name of the nurse, identifier of the foundling (name and number), initial and final date of services, table of payments/salary |
| Foundlings given to their Mothers | Registries of foundlings reconciling with their mothers and leaving the institution |
| Record of Letters | Registries of letters sent and received by the institution. For example the foundling wheel communication with the authorities, including the police, government or the church |
| Movement of Foundling | Movement of foundlings, for example to another institution. These documents contain information including the name of the foundling, identifier and name of the nurse. |

Table 1: Topic Based Document Classification

The process of digitizing the physical documents and incorporating them in a digital platform, provides an important role in the dissemination of the archive, but a desirable outcome is an additional layer, a direct access to important pieces of information within each document.

The goal is the development of a knowledge repository, where a set of entities and the relations between them are permanently stored.

In the following sections, the semantic web and supporting technologies are introduced. The objective is to contextualize the usage of knowledge repository in the development of applications.

After the introduction of the semantic web, the sample documents are revisited, in order to outline a first iteration of the knowledge repository domain, the ontology.

### 3.1.1 *Semantic Web*

In a simplistic manner, the internet can be interpreted as a group of pages with content. These pages may contain links, which connect one webpage to the next, effectively creating a network of pages.

The main issue with this structure is most webpages are extremely human centric, in other words, most webpages were developed to be read and interpreted by humans. The information is scattered across the different elements within a web page and surrounded by text. Humans can easily read and interpret the meaning of the content, but in opposition, this format is not optimized for machines.

The solution is the introduction of a standard and structured format, in order to organize the information in an easily machine understandable format, according to statements. These statements constitute a knowledge graph, and this internet is usually referred as the semantic web.

Similarly to how a link connect one webpage to the next, the information follows a similar pattern. Each element contains a unique identifier, analogous to the unique URL that identifies a webpage. A relation simply is a link that connects one piece of information to the next, unlocking machines to interpret and extract meaning without heavy processing, in opposition of extracting information dispersed across blocks of text.

The introduction of semantic web improves the internet usability by humans. Machines being able to connect different pieces of information and extract meaning from text, refine search engines results and data gathering systems.

As an example, the Google search engine introduced in 2012 the Google knowledge graph[1] cards in their results. When a specific term is searched, a card with valuable information is displayed on the right-hand side. The content is the known related terms to the searched entity, accordingly to the Google knowledge graph. Date of birth or date of death of a person, are a few common examples of relations displayed in these cards.

Figure 3 displays the Google card obtained after searching the Portuguese poet *Camões* on the Google search engine.

1 More information about the Google Knowledge Graph and respective cards available: https://en.wikipedia.org/wiki/Google_Knowledge_Graph, Accessed in December 2020

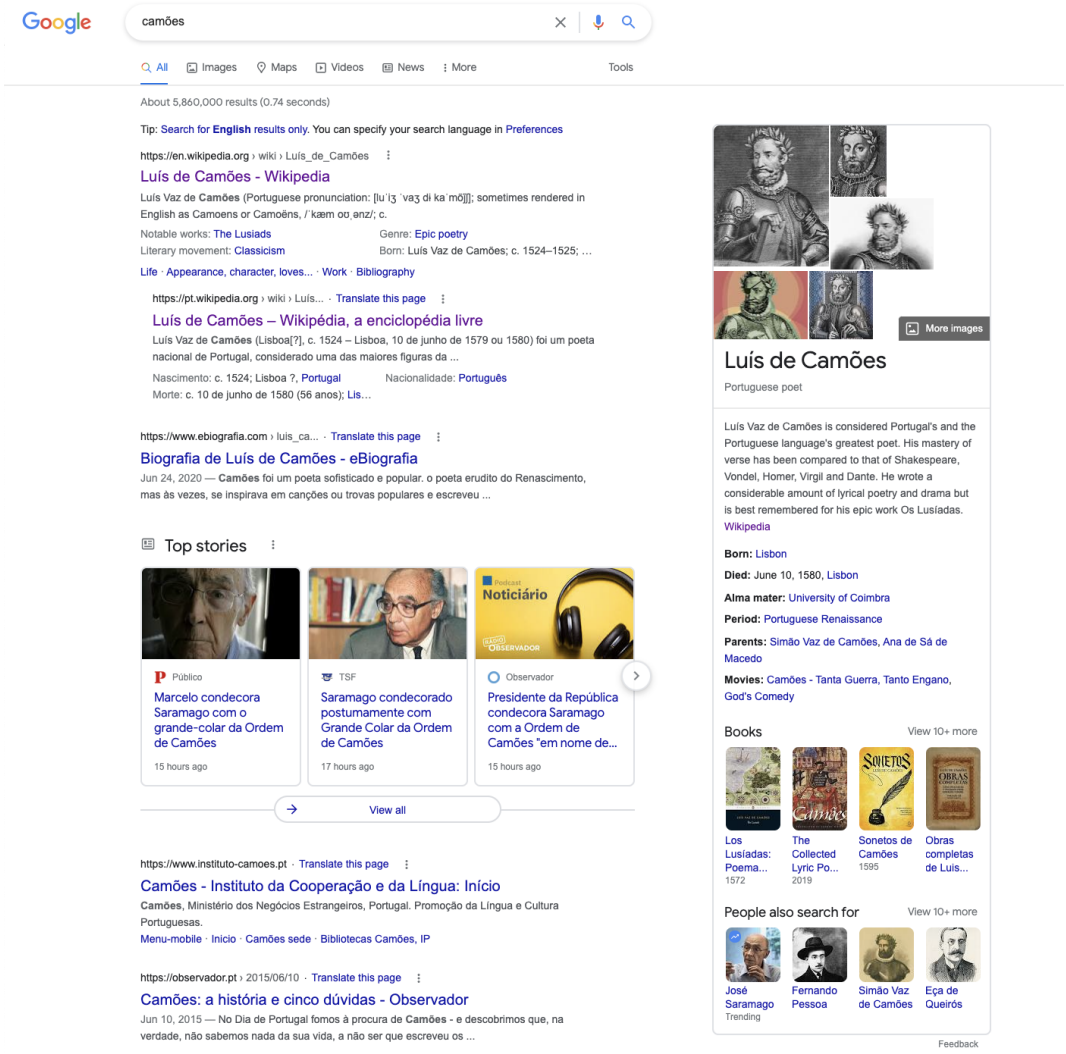



Figure 3: Google Knowledge Graph Card

### 3.1.2 *Introduction to Knowledge Representation Standards*

The World Wide Web Consortium (W3C)[2] is the organization responsible for the development and maintenance of the standards used across the internet.

When the semantic web was approaching, the introduction of a new standard to define and share knowledge was necessary. In the year 1997, W3C began the development of the new standard to describe semantics, the Resource Description Framework (RDF)[3]. This standard is one of the oldest standards introduced by the W3C in use today.

2 Official W3C webpage available at `https://www.w3.org/`, Accessed in October 2020
3 Additional details on the RDF standard available at: `https://www.w3.org/RDF/`, Accessed in October 2020

#### 3.1.2.1 *RDF*

RDF is the foundation for other more complex knowledge representation standards, introduced furthered down the line, similarly to how the HTML standard still is the fundamental element to display content on the internet till this day. Even modern full fledged web application, fundamentally display all the information in an HTML format.

As previously mentioned, each piece of information on the semantic web is identified by a Universal Resource Identifier (URI). A relation between a pair of entities, in RDF is specified as a triple of elements, a subject-predicate-object triple.

As an example the following sentence *John likes to eat apples*, the meaning can be expressed accordingly to a triple. John is the subject of the sentence, consequently, this entity is also the subject of the relation. In a similar manner, the term apple is the object of the relation. The predicate is the relation itself, the link between the subject and the object. In this particular example, a possible predicate is *likes*, or if the vocabulary allows a more specific term, *likesToEat* is another possible predicate.

The knowledge graph is composed of these triples. By connecting additional triples such *John likes Mary*, *John livesIn Spain* and *John studies Medicine*, a network of information centered around John starts to emerge.

#### 3.1.2.2 *RDFS*

The RDF Schema (RDFS)[4] is an extension of the RDF standard, but instead of being used to describe information in a triple format, this standard is used to describe vocabularies. As an analogy, the RDFS is the database schema, while RDF the tables with the content.

The expressiveness of RDFS allows the definition of class entities, respective subclasses, properties, restrictions over the previously mentioned elements and the possibility of adding annotations.

#### 3.1.2.3 *OWL*

The Ontology Web Language (OWL)[5] is another standard used to describe vocabularies, while providing a greater expressiveness in comparison to RDFS. Everything that is expressible with RDFS, it is also possible to be expressed with OWL, but an additional level is provided to refine the specifications.

Initially, it is possible to define classes as disjoint. An instance of the class A cannot be simultaneously an instance of the class B, in case of disjointedness. In order to illustrate, for the musical instrument domain, the different musical instruments sub classes should be

4 Official website available at:https://www.w3.org/TR/rdf-schema/, Accessed in October 2020

5 Knowledge representation language, more information available at https://www.w3.org/OWL/, Accessed in October 2020.

defined as disjoints. A string instrument such as a guitar cannot be an instance of the class wind instruments simultaneously.

The additional expressiveness allows properties restriction. For instance, a musical instrument should be classified as a string instrument, only if strings are part of the composition of the instrument.

Furthermore it is possible to classify relations in OWL as functional, transitive, inverse and symmetric. A functional relation, for a particular subject, at most, has one object. For instance, the biological mother relation is functional, since it is impossible to define one person with two biological mothers. Ancestry is a good example to describe transitivity. For instance, John is father of Pete and Pete is father of Kate. Since John is ancestor of Pete and Pete ancestor of Kate, automatically John is also ancestor of Kate, her grandfather. Inverse properties denotes that for a relation from A to B, a relation from B to A must exist. For instance, the triple *John hasParents Mary* automatically has the inverse property *Mary hasChild John*. Symmetry is similar to inverse, but the relation from A to B, it is the same from B to A. For instance the relation siblings is symmetric.

An important note is that OWL distinguishes object properties, relations between entities with a URI, and data properties, relation between a concept and a data-value attribute, elements without identifiers, such as an integer or a string.

Further details on ontology development are available in the Horridge et al. (2009) guide. While an example ontology is defined, all the different concepts are carefully introduced. The example project was developed with Protégé, an open source ontology editor.

## 3.2 ONTOLOGY CONTEXTUALIZATION

The ontology emphasizes the relations between the different concepts, the structure of the knowledge domain.

The conceptualization is composed of two stages. Initially, the sample documents are revisited and these documents are annotated. The goal is to determine the set of concepts that the knowledge domain must support. The following step is the identification of relations between the defined concepts.

The followed approach was the definition of an ontology, in a light weight description language. The result is an initial representation of the knowledge domain.

### 3.2.1 *Annotation of Documents*

The process of identifying concepts within a textual document was initialized resorting to the Doccano[6] text annotation tool. This tool runs on a local environment, where the interface is accessible via most internet Browsers.
The annotation process consists in defining a set of entities and annotate sequences of characters with a label. The result of the annotation process is stored in a database, but it is possible to export as a JSON Line (JSONL) file. Each line corresponds to a JSON object, with the original text and a list of triples. These triples are composed of the label annotated and the position in the text, the initial and final character index.

Figure 4 contains a sample document annotated with Doccano.


Figure 4: Annotated Document With Doccano

Since the case study concerns the foundling wheel, this term was identified as the main concept. From this point, the identification of the remaining entities was an iterative process. The second group of concepts identified was references to the roles that a person assumes, from the perspective of the institution. Terms including foundling, nurse, clerk, councilor and all the different family hierarchy are a few instances of possible roles.

6 Doccano is a free and open source textual annotation tool. Used to highlight and mark sequences of characters in a document, with a respective label. The application runs on a local environment, accessible via most internet Browsers. More information available in the official page: https://github.com/doccano/doccano, Accessed in February 2021

Other fundamental group of concepts is the main life events of each foundling. Events including the registration or departure from the institution were identified. The identification of the concept event was followed by complementary concepts, such as dates or places. The final group of concepts identified was the layette and signals.

### 3.2.2 *Initial Ontology*

After identifying the main concepts contained in the documents, the process was proceeded with the identification of the relationships between the different concepts. The strategy used was the development of an ontology to define the knowledge domain, identifying the relations through the analysis of the transcribed documents.

The development of the ontology was completed resorting to the Ontology Description Language (OntoDL)(Fonseca et al., 2014; Pereira et al., 2016). The decision of choosing the OntoDL in opposition to other popular formats was the light-weight and simpler notation. Another advantage provided is OntoDL automatically exports the ontology defined in other industry standard formats, including OWL and graph description language (DOT) (Hamdan et al., 2019).

The process of defining an ontology according to the OntoDL notation, consists in dividing the declarations in four main blocks, as shown in the Listing 3.1, the *conceitos*, *individuos*, *relacoes* and *triplos* (Araújo et al., 2020).

```
Ontologia Ontology_Name

conceitos { ... }
individuos { ... }
relacoes { ... }
triplos { ... }
.
```

Listing 3.1: Structure of an OntoDL File

In the *conceitos* block, all the different concepts are declared, separated by commas. Attributes are added to each concept by defining them between square brackets. For example, the declaration *Ball[color : string, radius : int]* denotes the *Ball* concept with two attributes, a color attribute of the string type and an integer radius attribute.

The *individuos* section, the identifiers of individuals are declared. These declarations are used during the process of defining instances of concepts.

In the *relacoes* block, the relation identifiers are defined, indicating the domain and codomain between square brackets (this last indication is optional). For example, *loves [domain: Person, codomain: Sport]*, introduces one relation named loves which requires as

subject the concept Person (or an instance of it) and requires as object the concept Sport (or an instance of it).

In the *triplos* block, relations are defined following the notation *Subject= Predicate => Object*. The subject is the domain of the relation and the object the codomain. The predicate specifies the relation between two concepts. Furthermore, the OntoDL specification provides reserved keywords to specify special relations.

The keyword iof (instance of) is used to define instances of a concept. For example, *Carl = iof => Person* indicates that *Carl* is an instance of the concept *Person*.

The keyword isa (is a) is used to define subclasses. For example, *Car = isa => Vehicle* indicates that the concept *Car* is a subclass of the concept *Vehicle*.

A final note is that in the *triplos* block, the square brackets are used to group relations with the same domain. For example, the declaration *Person=[loves=>Sport, owns=>Car]* is equivalent to *Person= loves=>Sport* and *Person=owns=>Car*.

The Listing 3.2 describes the initial ontology developed in OntoDL.

```
Ontologia Exposto

conceitos {

    Casa_Da_Roda[nome:string, dataInicio : string, dataFim : string, Descricao :
    ↳ string, regulamento: string],
    Evento[Data:string, tipo:string],
    Local[Descricao:string],
    Pessoa[nome:string, funcao_pessoa:string, genero:string],
    Exposto[id:int, historia:string],
    Sinal[Descricao:string, tipo:string],
    Enxoval,
    Item[Descricao:string],
    Imagem[Folio:string, Referencia: string, Ficheiro : string]
}

individuos {

    Bilhete,
    Casa_de_Fafe

}

relacoes{

    participa[domain:Pessoa,codomain:Evento],
    composto[domain:Enxoval,codomain:Item],
    possui[domain:Exposto, codomain:Sinal],
    pessoas_associadas[domain:Casa_Da_Roda, codomain:Pessoa],
    ocorreu[domain:Evento, codomain:Local],
```

```
    localizada[domain:Casa_Da_Roda, codomain:Local],
    reside[domain:Pessoa, codomain:Local],
    padrinho[domain:Pessoa,codomain:Pessoa],
    madrinha[domain:Pessoa,codomain:Pessoa],
    pai[domain:Pessoa,codomain:Pessoa],
    mae[domain:Pessoa,codomain:Pessoa]
}

triplos {

    Pessoa =[
            participa => Evento,
            padrinho => Pessoa,
            madrinha => Pessoa,
            pai => Pessoa,
            mae => Pessoa
            ];

    Exposto =[
        isa => Pessoa,
        possui => Sinal,
        possui => Enxoval];

    Bilhete = iof => Sinal[Descricao='Bilhete que contem o nome do exposto', tipo='Escrito'];

    Enxoval = composto => Item;

    Casa_de_Fafe =[
        iof => Casa_Da_Roda[nome='Casa da Roda de Fafe', dataInicio='', dataFim='', Descricao='', regulamento='' ]
    ];

    Casa_Da_Roda =[
        pessoas_associadas => Pessoa,
        localizada => Local
    ]
}
.
```

Listing 3.2: Ontology Defined in OntoDl

As previously mentioned, OntoDL exports the ontology in multiple formats, including DOT. Graph visualizers, such as Graphviz[7], are capable of rendering the ontology, while

7 Open source graph visualizer, used to render graphs defined in DOT. More information available at: https://graphviz.org/, Accessed November 2020

abstracting the underlying extensive ontology definition. This abstraction facilitates the communication with the archivists, by providing a common and easily understandable language.

Figure 5 is the result of rendering the previously defined ontology (Listing 3.2) by the Graphviz software.

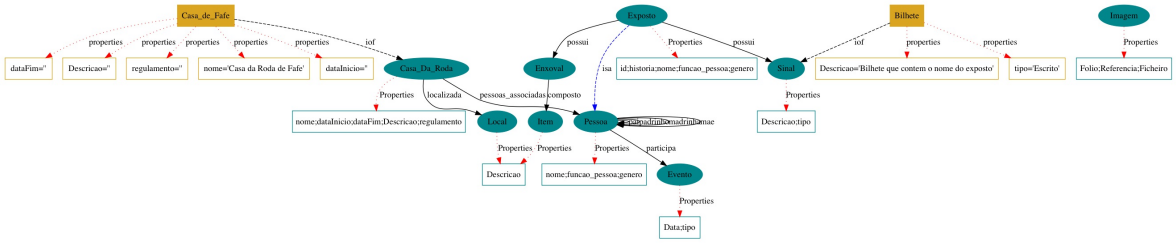



Figure 5: Graphical Ontology Representation

The definition of concepts and relations was supervised by Dr. Mónica Guimarães. Her expertise in the archival domain was fundamental in the process of defining a knowledge domain, based on the sample documents provided.

## 3.3 System Architecture

After identifying the important information, according to the knowledge domain defined, the following step is the design of the entire application.

The system architecture globally describes the integration between the components responsible for processing the documents and the resulting application, the communication link between the end user and the knowledge repository.

Figure 6 exhibits a block diagram that depitcs the architecture of the overall system.

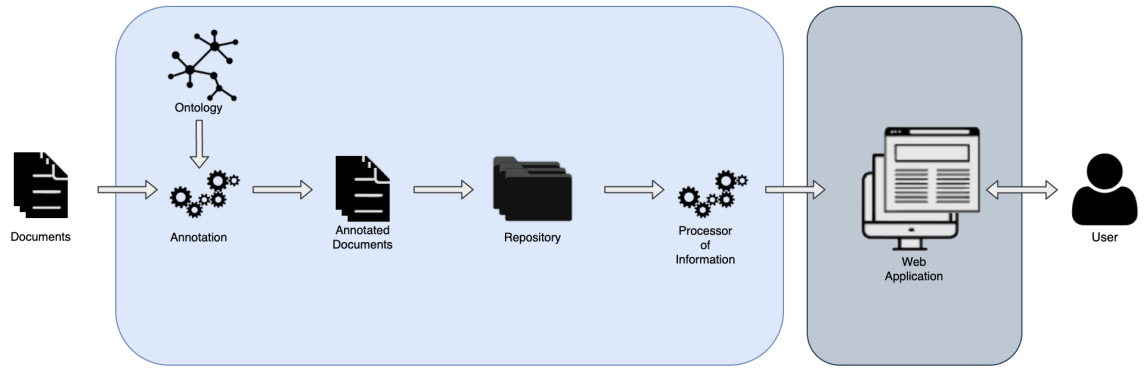



Figure 6: System Architecture

The documents are the result of transcribing the original physical documents.

The annotation process consists in identifying instances of important information contained in each document, more specifically, identifying instances of concepts and respective relations. The annotations are set accordingly to the conceptual model, defined in the ontology.

The extracted entities and relations are permanently stored in a repository. This is the knowledge repository, the infrastructure that supports the web platform.

The web platform is the component responsible for establishing the communication between the knowledge repository and the users. By using that interface, users are able to search over the available content and update the repository.

# 4

# DEVELOPMENT

This chapter focuses on a detailed description of the entire development process, by chronologically follow every step of the solution. Additionally a discussion on possible alternatives and obstacles faced.

The structure of this chapter combines the solution and the introduction of different technologies, techniques and respective decisive factors. The goal is to follow along the development process, and in parallel, introduce the different concepts, accordingly to the current step under analysis.

## 4.1 DEVELOPMENT DECISIONS

This project is composed of three main components, the information extraction, the ontology and the web application.

The entity and relation extraction are the components responsible for automating the identification process of concepts and relations from documents. The extraction was automated resorting to Spacy, a natural language processing toolkit.

The second element is the knowledge representation component, composed of an ontology, a triple storage, query engine and a respective reasoner to derive additional information. The proposed solution utilizes the OWLReady toolkit, as a one stop solution to manage all of the different semantic components.

The third component is a web application, responsible for establishing the connection between the end user and the knowledge repository. The web application was developed with the Django framework.

After generally describing the three main components and technologies, the remainder of this chapter chronologically follows the development process. Every time a new concept, or any other detailed analysis is required, the development description is briefly stopped and new concepts are accordingly introduced.

## 4.2 DATASET DEVELOPMENT

The archivists were responsible for providing the transcriptions set, the result of carefully inspecting and manually typing each of the physical document contained in the archive. Additionally, was provided a set of attributes that characterize each book. These books organize the content of the archive, grouping documents according to a specific criteria, including the main topic and time period.

The following list contains all the different attributes used to describe each book:

- *Título* - Title of the book,
- *Cota* - Reference number of the book,
- *Termo de Abertura* - Initial charter,
- *Termo de Encerramento* - Final charter,
- *Observacões* - Observations,
- *Datas Extrema Inicial* - Date of oldest document contained in the book,
- *Datas Extrema Final* - Date of newest document contained in the book,
- *Dimensoes* - Dimentions of the book.

The digital documents were provided in the Microsoft Word format, as a result, the content requires to be extracted into a standardized format, without the unnecessary metadata. The Microsoft Word format contains additional marks, for instance a wide range of the stylistic features, which from the perspective of this project, are undesirable.

Furthermore, during the process of extracting the content from word documents, it is necessary to decide what information should be kept and how the resulting data should be organized. The criteria used was only preserving the content concerning each book and their respective transcriptions. The objective is to organize the extracted content in two main sets, the book and the transcription set.

### 4.2.1 *Data Extraction*

As previously mentioned, all the documents were provided in a Microsoft Word format and a desirable outcome is to automate the content extraction from these files.

Since 2007, MS Word[1] documents follow an XML based format. By interpreting a file as a known standard format, the processing of each word document is simplified. The solution

1 Additional details on the underlying XML-based formats available at: https://support.microsoft.com/en-us/office/open-xml-formats-and-file-name-extensions-5200d93c-3449-4380-8e11-31ef14555b18, Accessed in February 2021

consists in automating the content extraction from the word documents, with a script written in the Python programming language. The combination of loading the standard XML Python module and the XML Word file schema, provide the necessary tools to parse each paragraph as a string, stripping the content from any stylistic feature or any other word document mark.

The content within each paragraph requires to be filtered, only preserving the desired transcriptions and books. An important note is during the process of transcribing each document, books and transcriptions were surrounded by horizontal lines. These lines assume a delimiter role, used to fragment the content. The body of the word document is extracted, fragmented into smaller components and these components are separated in two groups, the books and the transcriptions.

As shown in Listing 4.1, a book is structured accordingly to a set of descriptive attributes. This structure is similar to the JSON format, key-value pairs separated by a colon character.

```
Cota: 1-1-24-24
Datas extremas: 1865-11-11 a 1866-12-26
Título: «Nº7 - Entradas dos Expostos na Roda de Celorico de Basto».
Termo de abertura: “Este livro hade servir para nelle se lançarem os termos d´entrada
dos Expostos na Roda deste Concelho e leva no fim o competente termo d´enserramento.
Celorico de Basto, 5 de Novembro de 1865.”
Serafim da Fonseca Sylveira M.(?)
Termo de encerramento: “Tem este livro cento e cincoenta e oito folhas, que todas
vão numeradas e rubricadas com o meu apellido de =.Celorico de Basto, 5 de Novembro
de 1865.”
Serafim da Fonseca Sylveira M.(?)
Observações: Contém documentos soltos (relação do Expostos que tem de ser recenseados
pelo Concelho de Mondim para o recrutamento militar; idem de Cabeceiras de Basto
e outros)
Dimensões: 33,3cm+22,6cm+3,3cm
```

Listing 4.1: Book Fragment Extracted From a Word Document

The transcriptions in an initial state require a simple cleaning process, for instance, removing the excessive line breaks and spaces. As an example, in Listing 4.2, it is shown a transcription extracted from a word document by the script. As it is shown, the empty lines between the main transcription body and the signatures or the title were removed.

```
Cota: 1-1-24-29
Capa «Nº12 - Entrada dos expostos na Rode de Celorico de Basto
Data: 1871-10-24 a 1871-12-28
```

```
Fólio 1º verso
Nº 95 de 1871
Eliza Evangelina de Lorena Idade 20 de Outubro de 1871
Aos vinte e quatro dias do mês de Outubro de mil oito centos setenta e um; nesta
villa de Basto de Basto e Administração do Hospício d'Expostos perante mim Escrivão
compareceo a Directora delle e disse que hoje lhe fora apresentada uma criança de
sexo feminino que pela meia-noite de vinte e dois para vinte e três fora abandonada
à porta do hospício d'Expostos desta villa, representando ter de nascida quatro dias,
e que trazia um escripto de theor seguinte:
Apontamentos que devem ser cumpridos =Esta exposta deverá ser baptizada com o nome
de'Eliza Evangelina de Lorena. Leva no braço direito um fio de (?) amarello
- Oxalá que della tenhão (?) sem o que (?) a eternidade. Trazendo d´enxoval duas
camisas de linho e uma de (?) , um lençol de pano cru novo (?) de cor puído, uma
baeta seda branca, um lenço branco e (?) vermelha, um (?) de linho e (?) de pano
cru. Foi baptizada no dia de hoje na Igreja de Britello pelo Padre Claudio Alves
Cardoso, ...., forão padrinhos o mesmo baptizante e Carolina Rosa Exposta assistente
no hospício, e lhe poserão o nome recomendado no escripto assim copeado.
E para constar se lavrou o presente termo que a rogo da Directora vae assignar Joaquim
António Pacheco Pereira desta villa. E eu Francisco Alves Machado de Carvalho, Escrivão
da Camara que (?)
Fran.co Alves Mach.do Carv-º
Joaquim António Pacheco Pereira
```

Listing 4.2: Transcription Fragment Extracted From a Word Document

These resulting extracted data is stored in a pure textual format. Each parsed fragment is stored in a separate file, organized in different folders, a folder for each word file, since most word files group documents from the same institution. The decision of storing the results in a set of text documents is justified by the necessity of an initial dataset. This dataset is the groundwork for the following steps of the development process.

### 4.2.2 *Data Cleaning*

Most documents provided are dated to the nineteenth century. The advanced nature of these documents, automatically bring linguistic and degradation issues that must be carefully analyzed.

Towards the degradation issue, the content of some documents became unreadable. The degradation varies from a missing character inside a word to complete unreadable sentences within a document. During the process of transcribing each document into a digital format,

the degradation issues were marked with the usage of additional punctuation. Three dots denote an unreadable section, a dot inside a word mark a missing or unreadable character in the original document. A question mark is also used to denote a missing word. An example of this additional punctuation can be seen in Listing 4.2.

The second main issue is the evolution of the Portuguese language. Throughout the last few centuries, the Portuguese language evolved, and as a result, contemporary Portuguese differs significantly to the written Portuguese contained in the transcriptions.

An important note is some documents were previously annotated. During the transcription process, some concepts were surrounded by an equal sign. Textual references to a particular person or a location found in the document are a few examples of the annotations provided.

Following the identification of the main issues, it is necessary to decide what alterations should be made over the available transcriptions.

The first modification was stripping the transcriptions from any additional punctuation used to mark entities. The goal is to preserve the tokens and the original punctuation.

The second set of modifications was fixing the missing characters and spellcheck the transcriptions according to contemporary Portuguese. The decision of updating the documents according to contemporary Portuguese is justified by the integration with natural language processing technology, since most of these models are trained on contemporary Portuguese. The original content is always preserved, since it is possible to access the transcription prior to the processing and a picture of the physical document. The cleaned version of each document is used as an input to the entity and relation extraction, components of the data extraction pipeline.

Initially, the process of cleaning the data involved the usage of standard spellcheckers available in a Python environment [2]. But the oddly placed punctuation inside words and the gap between former and current Portuguese provided disappointing results. For instance, a word replaced by another word with a completely different meaning and tokenization issues, resulting from the odd punctuation and white spaces.

To overcome these issues, misspelled words were analysed. Since the documents provided are constrained to a small domain and most documents follow similar patterns, the same spelling issues and missing characters were frequently found. A possible solution is simply replacing that small pool of common errors by the correct form, instead of training or building a complex auto correct and spellchecking system.

The approach of simply finding and replacing common errors for this specific domain, provided positive results. The replacement table is stored in a two column CSV file, the first column stores the incorrect form and the second column the version with the correct spelling.

2 Pyspellchecker, ContextualSpellCheck and Hunspell were the spellchecking packages tested. More information available at: https://pypi.org/project/pyspellchecker/, https://pypi.org/project/contextualSpellCheck/ and https://pypi.org/project/hunspell/, Accessed in April 2021

When a close match is found, for instance after ignoring case sensitivity and comparisons against the lemmatized form, the incorrect tokens are replaced.

## 4.3 Entity Extraction

After accomplishing the task of developing a clean dataset, the following step is the extraction of important information within each document. Since one of the main goals of the project is the development of an ontology, where the end user is able to navigate through information via a web application, the solution of simply storing the clean transcribed documents is insufficient. The important information must be easily accessible, in opposition of forcing the end user to read an entire document, or set of documents, simply to find the answers for what they are seeking.

Before embarking on the information extraction process, it is required to defined what information should be extracted from each document.

The backbone of the desired information is usually referred as named entity. A named entity is a word or a group of words, immediately adjacent, that usually mark an object from the real world, labeled with a respective identifier.

To further illustrate the entity concept, Figure 7 contains the entities identified from an excerpt of the Universidade do Minho Wikipedia[3] page. These entities were automatic inferred with the usage of a pre-trained English[4] model, capable of automatically identifying entities in an English text. The figure was rendered by an entity visualizer[5].

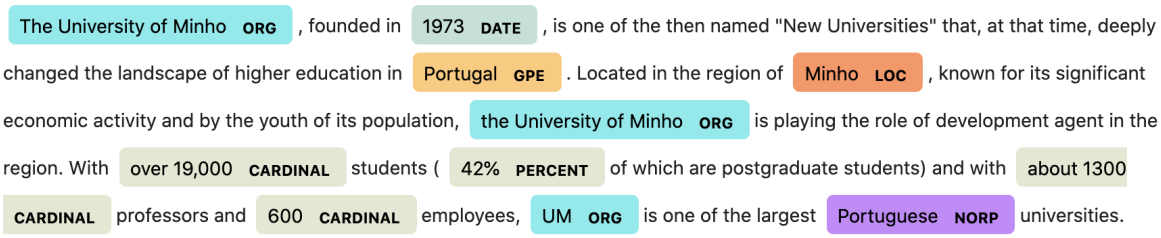



Figure 7: Entity Detection

As Figure 7 suggests, a group of entities were automatically identified, with the respective label written in all capitals, right next to each entity.

3 Full text available at the following page: https://en.wikipedia.org/wiki/University_of_Minho, Accessed in October 2021

4 Small pre-trained Spacy model for the English language. More information available at: https://spacy.io/usage/models, Accessed in October 2021

5 Standard Spacy entity visualizer, more information available at: https://spacy.io/usage/visualizers#ent, Accessed in October 2021

The entity *The Univídade do Minho* and the respective initialism *UM*, were both labeled as an *ORG*. This label is used to mark organizations such as companies, agencies and institutions.

Another set of entities identified were the numbers contained in these sentences. Some numbers were labeled as cardinals, others as percentages or as dates. The reason behind the usage of different labels to describe numbers is labels are set according to contexts. For instance, the number 1973 was not only identified as a number, but more specifically as a date, the founding year of the university. Since this model was trained to identify numbers within that particular context as a date, every time a number is found in a similar context, that particular number is also labeled as a date.

The entity *Portuguese* was labeled as *NORP* (Nationalities, religious or political entities), the entity *Portugal* as a GPE (Countries, cities or states) and finally the entity *Minho* as a *LOC* (Non-GPE locations), such as mountain ranges and bodies of water.

The entity concept is extremely important to understand the information extraction process. The first step is to define a set of entities, pieces of desirable information to be extracted from each document. The second step is extracting the relations between the different entities. All the different entities and their respective relations constitute a knowledge graph, a network of important information. This network is the end goal, an underlying knowledge base to support the digital archive.

For the specific domain of this project, foundling wheels and hospices, the desired entities are centered around these institutions, since they are the source of the archive.

The process of defining the important entities to be extracted from each document, was supervised by the archivists. After meetings and respective discussion process, the following list compiles the agreed entities that should be extracted:

- *Pessoa* - Identify a person contained in a document. This label marks the name of a person,
- *Funcao Pessoa* - Role of a person in this particular context. Terms such as foundling, nurse, priest or all the different family relationships referenced in a document are a few examples of possible instances,
- *Evento* - Events described in the documents. The analysed documents contain a wide range of referenced events such as the entrance of a foundling in the institution, birth, baptisms, adoption, reclaims by the biological family or death of a person. Events are extremely important to study the life story of each person,
- *Item* - Items or belongings of each foundling. Usually this label denotes the items that constitute the layette,

- *Local* - Label used to identify locations. This entity is extremely important for example to denote where each event occurred, such as the place of birth,
- *Identificador* - Identifier of each foundling. Each institution marks each foundling with a unique number,
- *Data* - Dates, entities used to identify two main elements: the year of the identifier and the date of a particular event, such as date of birth, date of death or date of entrance in the institution,
- *Instituição* - Institutions referenced in the documents, such as a foundling wheel or hospices.

After defining the entity set, the next step is effectively extract these elements from each document.

The previous *Universidade do Minho* example, as shown in Figure 7, demonstrates that language models are capable of automatically identifying entities from documents. That particular example utilizes an English model, but models for different languages or multi-language are widely available, including models for the Portuguese language.

Although these models are capable of identifying entities contained in a text, these models were not trained to identify entities for the specific domain of nineteenth century foundling wheels and hospices. For instance, professions such as nurse or administrator and family relationships such as mother or father, in this specific domain, should be labeled as *função pessoa*. But for a general purpose usage, family relations and professions are a completely different set of elements and consequently labeled differently.

The main goal is to automate the entity extraction process, by adapting and retraining a preexisting Portuguese model to identify not generic entities, but the entities for the specific domain of this project.

In the following section, a brief introduction to natural language processing and a discussion on the technologies used in the solution.

### 4.3.1 *Natural Language Processing*

Natural Language Processing, also known by the initialism NLP, it is an area of the computer sciences focused on the study and understanding of human languages by computers. NLP technologies are used in a wide range of applications, from simple word or sentence fragmentation, to the development of more complex systems, including text generation, used in the development of automatic chat-bots.

Most popular programming languages provide modules and libraries useful for the development of NLP applications, as an example the Stanford CoreNLP[6] for the Java programming language.

In this particular project, the Python programming language was chosen as the main development language. This decision was narrowed down to two main factors.

The first main reason is in the last few years, the Python programming language became one of the most popular programming language in the world. As a result, a wide range of libraries with an extremely active community and extensive documentation are readily available.

For the specific domain of NLP, the NLTK[7], Stanza[8] or Spacy[9] libraries, became extremely popular both in research and development of real world applications. Additionally, NLP technologies have strong ties with mathematics and machine learning technologies, used during the training process of linguistics models. Python provides a wide range of packages in these domains. The machine learning packages Scikit-learn[10], PyTorch[11] or Tensorflow[12] and scientific ecosystems such as the SciPy[13], which include multiple mathematical and data analysis models, are readily available for a Python developer.

The second main reason is personal preference and experience. In the last few years, the Python programming language became my main language of choice. Since I previously developed projects in Python that utilize both NLTK and Spacy packages, the decisive factor was simply to use familiar technologies, instead of embarking in a whole new environment during the development of this project.

After determining the main programming language, the following decision was choosing which natural language toolkit to use during the development of this project.

As previously mentioned, Python offers multiple NLP packages, including NLTK and Spacy. These packages include similar functionalities and development practices, but ultimately, the Spacy was chosen as the main NLP toolkit.

6 Multi language NLP toolkit for a Java environment. An important note is currently, the Portuguese is not supported. The official webpage available at: https://stanfordnlp.github.io/CoreNLP/, Accessed in March 2021

7 NLP toolkit for the Python programming language. More information available at: https://www.nltk.org/, Accessed in April 2021

8 NLP toolkit built on top of the PyTorch. Official page available at: https://stanfordnlp.github.io/stanza/index.html, Accessed in April 2021

9 NLP toolkit, oriented for the development of NLP powered applications. More information available at: https://spacy.io/, Accessed in April 2021

10 Machine learning toolkit for a Python environment, More information available at: https://scikit-learn.org/stable/, Accessed in April 2021

11 Open source Machine Learning framework. Official webpage available at: https://pytorch.org/, Accessed in April 2021

12 Open Source library for machine learning and artificial intelligence. More information available at: https://www.tensorflow.org/, Accessed in April 2021

13 Scientific environment for the Python programming language. More information available at: https://www.scipy.org/, Accessed in April 2021

Spacy is a high level package oriented for the development of NLP powered applications. Since the main goal of this project is the development of an application, instead of research in the natural language domain, the advantages that a higher level application provide, simplify the development process. An impressive groundwork is readily available, including pre-built language models and useful methods.

To exemplify the power of a small bare-bone Spacy project, the previous named entities extraction example from a Wikipedia page, as seen in Figure 7, was developed with Spacy. The pre trained small English model is loaded, followed by processing the Wikipedia example through a natural language pipeline. The results are rendered via the standard Spacy visualizer.

```
import spacy
from spacy import displacy

#Load the English model
nlp = spacy.load("en_core_web_sm")

#Process the text
doc = nlp("""The University of Minho, founded in 1973...""")

#Render the results, the named entities extracted
displacy.render(doc, style="ent")
```

Listing 4.3: Entity Extraction Script

As the Listing 4.3 suggests, with just a handful of lines, the combination of loading a standard language model and invoking high level functions, instantly provide fairly powerful results.

The functionalities provided by the Spacy toolkit are not strict to a small set of immutable models. The toolkit itself provides the functionalities of adapting, retraining or build from scratch, any component or model. For instance, replace the standard complex tokenizer by a simpler white space recognizer or adapt the sentence fragmentation component. Furthermore, it is possible to insert new components, which updates the model to perform new tasks previously unavailable. For instance, insert a Relation Extraction (REL) component to a model not trained to perform that particular task.

The main idea to retain is the possibility of adapting a pre existing model or simple develop a blank language model from scratch, both approaches are available.

The approach followed in this project was taking as much advantage as possible of the pre built Spacy models and functionalities. Instead of building an entire new model, the standard large Portuguese model[14] was adapted strictly where adjustments were necessary.

14 Standard Portuguese language model. This project utilizes the large version, provided by the Spacy toolkit. Additional details on this base model available at: https://spacy.io/models/pt, Accessed in May 2021

The goal is utilize available natural language processing technologies, adapt and reshape a pre existing Portuguese base model, in order to automate the annotation of named entities from documents.

Before embarking on the entity extraction process, a small introduction to the Spacy pipeline is required, the core component responsible for processing pure textual documents.

### 4.3.2 *Spacy Pipeline*

As previously mentioned, the Spacy module provides a wide range of models for more than 60 natural languages, including Portuguese.

A Spacy model, in a practical sense, it is a natural language processing pipeline. This pipeline is composed of a set of components, each responsible for a particular NLP task. The input of the first step is a string, each component feeds the immediate adjacent component and the final output is a Spacy document object.

Figure 8 describes a general Spacy pipeline, color coding the different NLP taks.

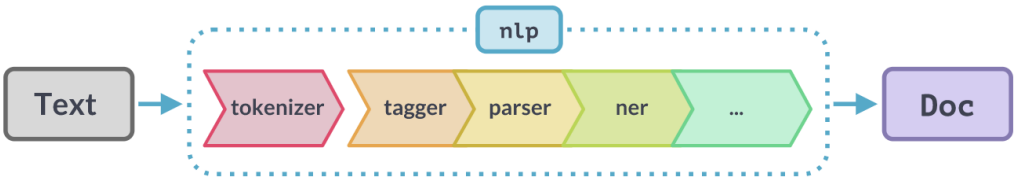



Figure 8: Spacy Pipeline[15]

The first step in pipeline is the tokenizer, responsible for fragmenting a text into tokens, usually a word. The rest of the pipeline is composed of a set steps, and the in Figure 8 example, the last step shown is the Named Entity Recognizer (NER), responsible for identifying named entities.

The pipeline is completely modular, a step can be disabled or modified, and the process of inserting a new component in the pipeline, usually corresponds to a new functionality.

The proposed solution consists in adapting the NER component, in order to shape the Portuguese model to recognize a new set of named entities.

The output of the pipeline, the Spacy document object, it is an iterable object, where each element is a token, and over each token, a range of methods are accessible. An important note is Spacy objects are immutable, for instance, filtering tokens from a document is a bad

15 Image source and additional details on the Spacy pipeline available at: https://spacy.io/usage/spacy-101#pipelines, Accessed in May 2021

practice. Instead, these tokens should be filtered before the pipeline or the desired outcome stored in a new object.

The main goal is to built and permanently store the new model. After accomplishing this task, the standard use case of processing information usually consists in loading the model, processing all the different textual documents through the model pipeline and extract the desired information from the resulting objects.

### 4.3.3 *Entity Extraction Component*

As previously mentioned, it is necessary to reshape the standard NER component, in order to extract a new set of entities.

The entire process was initialized with the development of a training dataset, a set of annotated examples. Previously, the transcriptions were extracted from word documents and the resulting clean document were temporarily stored in a pure textual format. These documents were annotated with the Doccano, similarly to the concept identification process shown in Section 3.2.1, but this time the process was repeated over a newer and larger dataset, more specifically, fifty clean transcripts were annotated. Although this is a relatively small number for a training dataset, during the development of this component, not all transcriptions were available and some transcriptions must be reserved to analyse the system behavior when it is faced with unseen data.

In order to develop the trainable NER component, multiple approaches are available, from the development of an entire system from scratch to the adaptation of a pre existing system.

In the third version of the Spacy toolkit, Spacy projects were introduced. In a simplistic manner, a Spacy project is a system to structure and organize the different components of a project, accordingly to a well defined folder and file dependencies, commands line tools and a range of other pre-built components. An analogy is a web development framework, where by instantiating a new project, a project folder with some groundwork is automatically initialized.

Since Spacy is oriented for the development of NLP powered applications, it is possible to instantiate not only a new blank Spacy project, but also a new project specialized to perform known standard natural language task, where the underlying machine learning models are provided. One of the available pre built Spacy projects is a trainable NER project. The proposed solution consists in adapting the standard trainable component of the processing pipeline to identify entities from the foundling wheel domain.

The modifications mostly consisted in adapting the trainable component to the Portuguese language, adapt the machine learning parameters and match the Doccano output format to the expected by the trainable component. The output of this process is a new Portuguese based model, with an updated entity recognizer.

The new language model is permanently stored and the entity extraction process consists in loading the updated language model and process each document through the updated pipeline. All the identified entities are accessible in the *.ents* attribute, available in every document after being processed by the pipeline.

Towards the results obtained an empirical analysis was made via processing transcriptions, including unseen documents, through the updated model. The results were analysed with the visualization tool provided by Spacy. As an example, Figure 9 shows a transcription after being processed through the pipeline.

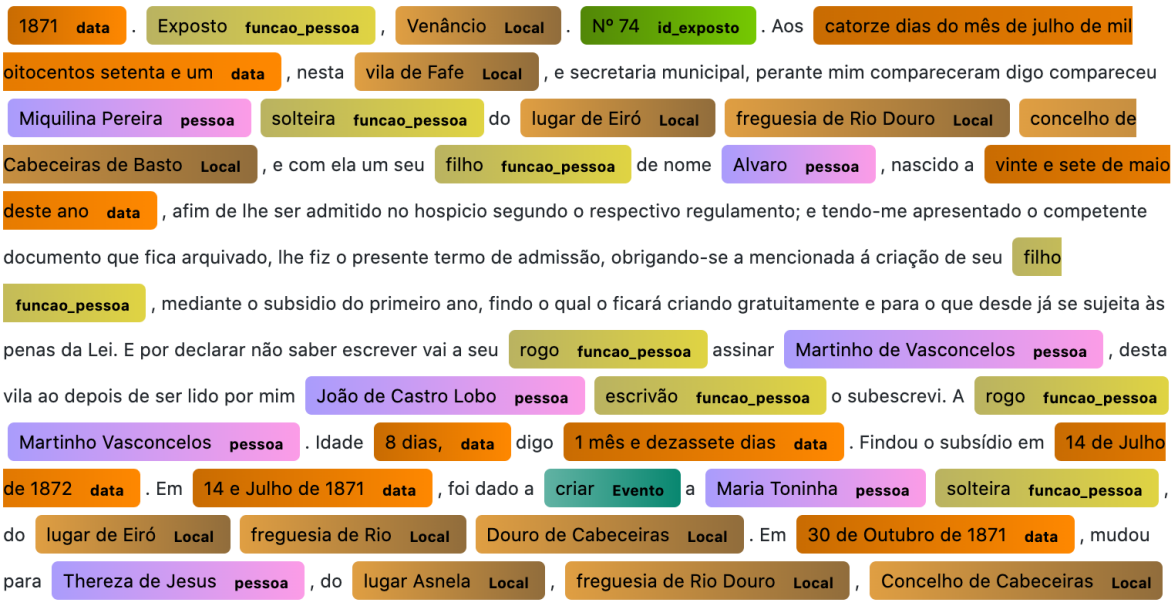



Figure 9: Entities Automatically Identified in a Transcription

The automatic entity recognition provided positive results, even though the training dataset was fairly small. The main issues found were unreadable tokens being wrongly labeled, which must be ignored, and numbers being wrongfully labeled. To overcome the latter issue, a small set of patterns were developed to correct the wrongfully identified entities. For instance a pattern that labels a number followed by the word *reis* as *dinheiro* and the token *Nº* followed by a number as *id_exposto*.

## 4.4 RELATION EXTRACTION

After accomplishing the the entity extraction, the goal is to expand the processing pipeline by automating the identification of meaningful relations between named entities.

Relation extraction is an active research area in the natural language domain, and the available solutions are generalized into two main approaches.

The first and more traditional approach is a pattern based solution. This extensive approach usually consists in a group of patterns, such as regular expressions, to find matches in a text. Similar approaches include token level analyses, for instance filtering bigrams or trigrams and filtering the predicate-subject-object from sentences.

The second main approach is a machine learning based solution. Similarly to the process of training the entity recognizer, state of the art NLP technologies train a model to extract relations between named entities.

In this project, both possibilities were carefully inspected, and in the end, both approaches were followed.

Initially, the proposed solution followed a pattern based extraction. Since all the named entities were automatically annotated and the format of most documents is fairly structured, the relation extraction task was a viable solution. The goal was to develop a relation extraction component without the intensive model training and additional pipeline components. Unfortunately, this solution did not provide the expected results, and the development process was stuck in this stage. The main difficulty found was the development of patterns capable of extracting relations between fairly distant entities within a document.

To overcome this issue, a machine learning based solution was followed, a similar approach to the entity recognition. Although the machine learning solution was capable of extracting relations between named entities, towards the end of the development cycle, the pattern based solution was revisited, resulting a set of better results.

In conclusion, a pattern based and a machine learning based solution were developed to extract relations from documents.

In the following sections, the different approaches are described. After analysing both cases, a discussion on the proposed solution and the model currently in use.

### 4.4.1 *Pattern Based Relation Extraction*

The pattern based solution is a combination of the matcher and the dependency matcher, parsers provided by the Spacy toolkit.

The regular expression concept is a good entry point to understand these parsers. A regular expression is a pattern used to match a section of a text. For instance, the regular expression *[0-9]{9}* matches numbers with nine digits, expression useful to extract the civil id number of a Portuguese citizen.

The matcher works in a similar manner to regular expressions. A pattern consists in describing token level rules to be matched in a Spacy document. The advantage provided by this tool is the possibility of developing patterns not only at the textual level, but also developing patterns over the attributes accessible by a token. Attributes such as Part Of

Speech (POS), lemmatized form, shape, morphological analysis and named entity label are just a few examples of possible components to be match with a pattern.

To illustrate a matcher use case, the goal is to extract places that a person likes or visited. A valid approach if filtering the name of a person by the part of speech, more specifically, filter proper nouns and pronouns. The destination is labeled as *LOC* or *GPE* by the standard entity recognizer. For relations, the lemma removes the variations caused by the tense, aspects and mood of verbs. This patter could be written as following:

```
pattern = [
  {"POS": {"IN": ["PROPN", "PRON"]}}, #filter Proper Nouns and Pronouns
  {"LEMMA": {"IN": ["like", "love", "go"]}}, #any form of the verb like, love
    ↳ and go
  {"POS": "ADJ", "OP": "?"}, #optional adposition
  {"ENT_TYPE": {"IN": ["LOC", "GPE"]}} # named entity labeled as Location
    ]
```

Listing 4.4: Matcher Pattern Example

The pattern shown in Listing 4.4 should match the following sentences: *John likes Spain*, *He loved France* or *Mary went to Portugal*. As the previous example suggests, with a fairly compact pattern, a wide range of relations are extracted from a document. In opposition, writing the equivalent pattern with text level regular expressions is a difficult task, since it is required to handle all the different forms of a verb, specifying the possible locations and extensive list of valid names.

The dependency matcher is a similar to the standard matcher, but provides a different approach to link sequences of matched tokens. A Spacy model provides a dependency parser component, which develops a tree of words from a sentence, structured accordingly to the syntactic dependency. For instance, Figure 10 contains a dependency tree of a sentence. As the Figure 10 suggests, the verb *visited* is the root of the tree.

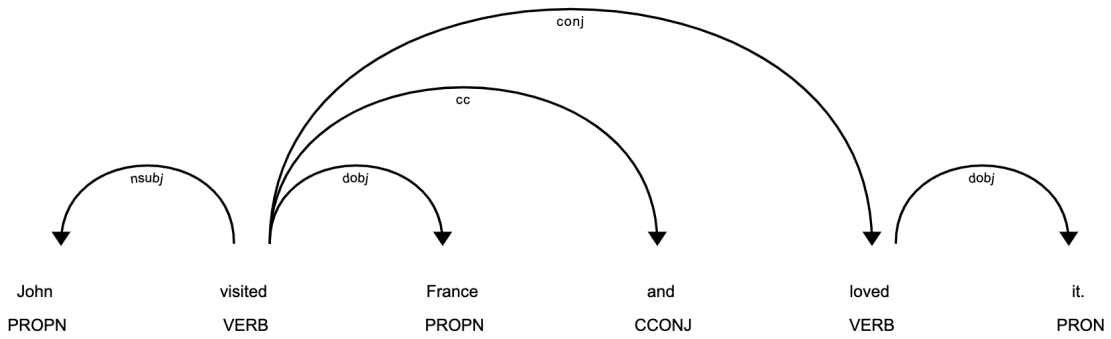



Figure 10: Dependency Tree of a Sentence

The dependency matcher is a combination of the standard matcher and the dependency tree. Instead of matching adjacent tokens, the pattern matches the tokens accordingly to the structure of the tree. The navigation through the tree is set by Semgrex operators[16]. These operators set the depth of the comparison, for instance, comparisons against the immediately precedent token, direct relation between two tokens or navigate through the entire subtree.

The proposed solution consists in finding patterns that link two named entities. Each entity is interpreted as an anchor, and according to the label of the named entity, the matcher finds the respective relation. As a concrete example, two important entities in this project are the *Pessoa*, used to denote the name of a person, and the entity *Evento*, which marks important events. For every person found in the subtree of the event, the system automatically infers them as a participant.

The remainder of the relation extraction process consists in the development of patterns to extract other meaningful relations.

The following list compiles the main patterns. Each relation is described accordingly to the domain and codomain entity label.

- *Função Pessoa - Pessoa* : Person and their role,
- *Função Pessoa - Local* and *Pessoa - Local* : Location or residency of a person,
- *Função Pessoa - ID* and *Pessoa - ID* : Personal identifier,
- *Função Pessoa - anoID* and *Pessoa - anoID* : Personal identifier year component,
- *ID - anoID* : Identifier and year of the identifier relation,
- *Evento - Data* : Date of an event,
- *Evento - Local* : Location of the event,
- *Evento - Pessoa* : Participant of the event,
- *Data - Local* : Location and date association, pattern used to indirectly extract event locations,
- *Instituição - Local* : Location of the institution,
- *Local - Local* : Connects sub locations. For instance, a locations with a street, city and country.

Each pattern developed contains a unique label, used for automatically infer the nature of the relation. The matcher is invoked with all the available patterns and the output is an

16 Additional details on the Semgrex operators and supported relations available at: https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/semgraph/semgrex/SemgrexPattern.html, Accessed in June 2021

iterable object with all matches. For instance, the *Função Pessoa - Pessoa* relation is labeled as *fp_pessoa*. During the process of iterating through all the matches, when a relation matched by this pattern is found, that particular label is accessible and the nature of the relation is automatically inferred, without additional processing.

### 4.4.2 *Machine Learning Based Relation Extraction*

The process of developing a relation extraction component follows a similar approach to the identification of named entities from Section 4.3.3.

As previously described, Spacy provides a set of pre built projects, extremely adaptable and capable of solving standard natural language tasks. One of the available pre built projects is a trainable relation extraction component, based on a bioinformatics example[17]. The goal is to adapt and incorporate the relation extraction component in the pipeline of this project.

The development process was initiated with the standard large Portuguese linguistic model, previously modified to identify a new set of entities. This model is not capable of extracting relations, since a Relation Extraction (REL) component is not available in the processing pipeline. As a result, the relation extraction component is inserted after the entity extraction, since the goal is to extract relations between named entities.

Updating the model consists in inserting the REL in the pipeline, training the newly added component and permanently store the updated model.

Similarly to the entity extraction, a new dataset with samples of positive relations is required for training purposes. In order to access the previously identified entities and ensure the usage of the same tokenizer during the annotation and training process, a Jupyter notebook replaced the Doccano annotation tool. The annotations are exported as a JSONL file, a line for each document with a list of entities pairs, linked by a particular label.

The updated model contains the retrained NER and the newly added REL component. By processing a document through the pipeline, the identification of named entities and respective relations is automated.

### 4.4.3 *Final Notes on Relation Extraction*

In order to automate the relation extraction from documents, a pattern (Section 4.4.1) and machine learning (Section 4.4.2) approach were explored. In this section, both alternatives are briefly analysed by taking in consideration the trade-offs of each approach. In the end of this section, the solution currently in use is introduced.

17 More information on the base relation extraction model available at: https://github.com/explosion/projects/tree/v3/tutorials/rel_component, Accessed in June 2021

The pattern based is a simple solution, providing positive results since the structure of most documents is fairly similar. The main disadvantage is the dependency tree is sentence bound, as a result, it is not possible to related elements beyond the sentence limits.

In opposition, the machine learning approach offers the cross sentence extraction. The main disadvantages is the training process requires large datasets, with a lot of positive examples, which it is beyond the current state of the archive with circa 250 documents.

Due to time constraints, one of the solutions had to be chosen, developing a hybrid solution or improving the machine learning component was not viable. Ultimately, the proposed solution is pattern based.

Although cross sentence relations are not automatically extracted by the dependency tree matcher, an additional patterns were developed. These patterns contain optional punctuation, to extract relations between elements close to the end of a sentence and the beginning of the next. Furthermore, the pattern based solution provides on the fly adjustments. In the later stages of the development process, a new pattern is easily added or updated, without retraining the entire model nor providing a large dataset to train a new relation type.

## 4.5 ENTITY LINKING

The final component of the information extraction process is solving possible ambiguities.

Over a document, it is fairly common for an entity to be referenced multiple times. The system should be able to determine that all of those references, in reality, are instances of the same entity. For example, a simple document composed of two sentences: *John was born in Scotland* and *John was born in 2021*. Both *Johns* are an instance of the same entity. By acknowledging that these pieces of information are referencing the same person, the system starts to connect the dots, developing knowledge over a particular entity.

The opposite scenario also provides ambiguities issues, since similar terms are used to describe distinctive entities. For instance, references to people with the same first name or a document comparing Paris, France to Paris, Texas.

In this particular project, the entities impacted the most by the ambiguity issue are the locations and people. Most of the remaining concepts are attributes in the knowledge repository. Wrongfully linking independent entities introduces inconsistencies in the knowledge repository. Simultaneously, not providing sufficient linkage, introduces repetition in the knowledge domain and a lack of interconnectivity. The goal is to establish a balance, automatically linking entities without compromising the consistency of the knowledge repository.

Spacy provides an entity linker and machine learning based approaches[18]. The goal is the development of a dataset composed of attributes that characterize each entity. A model is trained in order to match unambiguous entities to a unique identifier, according to context. Although this approach is a viable solution, some obstacles were faced.

A dataset with all the locations found in the sample documents was developed and used to train the model. The main issue is the insufficient training examples. For locations such as the residency of a person, that only appear a handful of times, the system is not capable of matching them to the identifier. Recurring municipalities such as *Cabeceiras de Basto*, *Celorico de Basto* or *Fafe*, were the only few learnt the locations.

Another issue is the world is open. Since the goal is to develop a web application where new information is constantly inserted, unknown entities are an issue. The system is not capable of performing the entity linking task over untrained entities, where their attributes and context are unknown to the model.

The development of periodic training mechanisms and updated datasets to contextualize recently unlinkable entities are beyond the scope of this project. Additionally increases the complexity of the system.

The proposed solution was the develop a simple system to solve ambiguities. The previous developed locations dataset[19], was repurposed, with the development of a repository. The locations with known variabilities, for instance *São Romão do Corgo*, *Corgo*, *São Romão* and *freguesia de S. Romão* are stored as a reference to the same location. The attributes used to contextualize the entities for the training model, were repurposed as a location description. Every time a close match is found, the entity linking occurs, otherwise, a new location is instantiated in the repository.

The document level ambiguity of personal names is solved with the development lists of possible candidates. The list of candidates is filtered according to the part of speech, morphological analyser and token level similarity. After the filtering process, if a candidate remains, the entity linking occurs. Otherwise, when multiple candidates are available, the set of candidates is ambiguous and the linking does not occur.

An important note is the process of solving ambiguities is integral part of the knowledge repository. After introducing the ontology, the storage solution and the integration between the information extraction and the semantic component, the entity linking problem is revisited in Section 5.3.2 and the underlying technologies to match ambiguous terms are furthered analysed.

18 ML based entity linking Spacy project available at: https://github.com/explosion/projects/tree/v3/tutorials/nel_emerson, Accessed in July 2021

19 Dataset combines both current and former Portuguese geographical locations, Sources used: https://www.csarmento.uminho.pt/site/s/arquivo-digital/item/94131, https://www.codigo-postal.pt/, https://cabeceirasdebasto.pt/concelho-freguesias-alvite-passos, https://www.wikidata.org/, Accessed in August 2021

## 4.6 ONTOLOGY INTEGRATION

The ontology, and the remaining semantic components of the application, are responsible for the underlying knowledge base that supports the entire project.

In a previous Section 3.2.2, an introduction to the semantic web, exploration of ontology standards and an analysis of sample documents, culminated in the original iteration of the ontology. But some technical aspects remain unexplored, more precisely how to integrate the process of extracting data from documents with the knowledge repository and how to effectively store these triples.

In the following sections, the remaining components, technology selection and the development of the repository are carefully described.

### 4.6.1 *Semantic Application Components*

The ontology is the central component of the knowledge repository, where the domain is defined. But in order to effectively materialize a repository, a set of supporting components are essential. The goal of this section is to briefly introduce the remaining unexplored components.

One of the main aspect of the knowledge repository is the storage solution, where different instances of entities and relation are permanently stored (del Mar Roldan-Garcia and Aldana-Montes, 2005).

A common solutions is storing all the different relations in a specialized database for semantic triples, a triple storage. This solution consumes more disk space than a relational databases and usually it is slower for very large datasets. The upside is these storage solutions are optimized for handling data in a triple format and export the results in semantic standards, including RDF.

Other possible solution is storing the knowledge in a graph database. Neo4j[20], Apache TinkerPop[21], Titan[22], GraphDB[23] are standard and readily available graph database implementations. In opposition of storing just the triples, the entire graph is permanently saved.

Data manipulation requires a query language, and different databases provide different solutions. The standard query language for triples and knowledge bases is SPARQL[24]. This

20 Official Neo4J webpage available at: https://neo4j.com/, Accessed in July 2021
21 Official TinkerPop webpage available at: https://tinkerpop.apache.org/, Accessed in July 2021
22 Distributed graph database, more information available at: https://titan.thinkaurelius.com/, Accessed in July 2021
23 Semantic graph database. Additional details available in the official webpage: https://graphdb.ontotext.com/, Accessed in July 2021
24 Additional details on the SPARQL query language available at: https://www.w3.org/TR/rdf-sparql-query/, Accessed in July 2021

language has a similar notation to the SQL[25] query language and provides the possibility of retrieving data from multiple ontologies simultaneously, within the same query.

Another unexplored aspect is the addition of reasoners in the knowledge repository. This component offers the possibility of deriving additional truths and automate the reclassification of concepts with class equivalences. Logical expressions are used to set the necessary conditions.

This additional exploration introduced the remaining components of the knowledge repository. The definition of a well establish knowledge model, the ontology, and the integration of the processing pipeline, combines the process of extracting triples from documents and matching them accordingly to the ontology model. The rest of the knowledge repository is composed of a storage solution, a query language to manipulate the data and a reasoner to automate the reclassification process.

### 4.6.2 *Selected Technologies*

The main programming language used during the development process was Python, as a result, the selected semantic technologies should be compatible with this environment.

The Spacy toolkit provides a one stop solution to manage the entire annotation and extraction process. A desirable outcome is finding an integrated system to manage the entire semantic components. Protégé is a known and well establish knowledge management environment, but the goal is to manage the entire ontology within a Python environment. The main decisive factor was finding a technology where the functionalities of defining the ontology domain, storage, query and reasoning are available.

Although Python provides multiple knowledge management packages, including the RDFlib[26], the development was supported by OwlReady[27], since it combines the entire semantic component as one integrated system.

The ontology is defined by loading a standard OWL file or by directly defining the ontology in the Python environment. The development follows a design pattern similar to the object oriented paradigm.

The class *Thing* is the superclass of all entities. In order to declare a new entity class, it must be declared as subclass of *Thing* or subclass of other previously defined entity. All of these definitions constitute the concept hierarchy of the Ontology.

25 Additional details on the SQL query language available at https://www.iso.org/standard/63555.html, Accessed in July 2021

26 Python package to manipulate data in the RDF format. Persistency is provided via the BerkeleyDB triple storage solution. More information concerning the RDFLib and presistency available at: https://rdflib.readthedocs.io/en/stable/persistence.html, Accessed in July 2021

27 Latest Owlready documentation available at: https://owlready2.readthedocs.io/_/downloads/en/latest/pdf/, Accessed in July 2021

```
with onto:
    #Class Person is subclass of Thing
    class Person(Thing):
        namespace = onto

    #Class Foundling is subclass of Person
    class Foundling(Person):
        namespace = onto
```

Listing 4.5: Concept Declaration in OWLReady

The relation definition is also class based, by defining it as subclass of *ObjectProperty*. The name of the class is the relation itself. Within the class body, the domain and range attributes set the subject and object of the relation respectively.

```
with onto:
    # Triples (Person,participatedIn,Event)
    class participatedIn(ObjectProperty):
        domain     = [Person]
        range      = [Event]
        inverse_property = attendBy
```

Listing 4.6: Object Property Declaration

Similarly, entity attributes are defined by declaring a subclass of *DataProperty*. The codomain is one of the available default types, including integers or string among other standard types. The OwlReady library is responsible for matching those types with the correspondent RDF element.

```
with onto:
    # Define datetime attribute
    class hasDate(DataProperty):
        range = [datetime.datetime]
```

Listing 4.7: Data Property Declaration

Class restrictions and equivalences are set by defining a logical expression within the class definition. These expressions are utilized by the reasoner to deduce additional information.

```
with onto:
    #A GodFather is a Male Person with at least one godson or goddaughter
    class Godfather(Person):
        equivalent_to = [ Person & Male & hasGodchildren.some(Person)]
```

Listing 4.8: Logical Expression for Class Equivalence

As described, by extending the correct pre built Owlready classes and matching the correct attributes expected by Owlready, the knowledge domain is defined, following object oriented practices.

All the different entities and relations instances, reside in a world object. A world is permanently stored in a SQLite3 database[28].

Although SQLite is a relational database, the Owlready manages the storage solution similarly to a quadstore. Since the ontology requires constant small calls to the database, the storage solution must provide fast responses with low overload. As a result, the OWLReady development group decided to use SQLite3 as the main storage solution.

OWLReady offers multiple methods to manipulate the content. Since the ontology is defined accordingly to classes and the content is stored in a database, the knowledge component is manipulated similarly to a standard object oriented project, for instance, new entries by invoking a class constructor. Additionally, pre built methods including *get_properties* or *get_inverse_properties*, that return all the relations for a particular entity, remove the necessity of developing common queries. Another approach is the usage of SPARQL queries, since Owlready provides a query engine.

Over a world object, the pre built Hermit reasoner is activated. The process of inferring additional knowledge and asserting the consistency of the ontology, consists in synchronize the reasoner. The inferred information is automatically added to the world object. An important note is the possibility of opening multiple worlds simultaneously, in order to separate the defined and inferred content in different worlds. For reasoning purposes, it is possible to set a world as opened or closed. Open world reasoning assumes that the non explicitly declared knowledge as unknown. In order to state false statements, they must be explicitly declared. In opposition, closed world reasoning assumes that the unknown is false.

Figure 11 describes the entire OWLReady architecture. As the figure suggests, all components are fully integrated and accessible in a Python environment.

28 OWLReady forum post, made by a member of the development team, justifying the usage of SQLite3 as the main storage solution, in opposition of a standard triple storage. Webpage available at: http://owlready.306.s1.nabble.com/connection-to-triplestore-td512.html, Accessed in October 2021

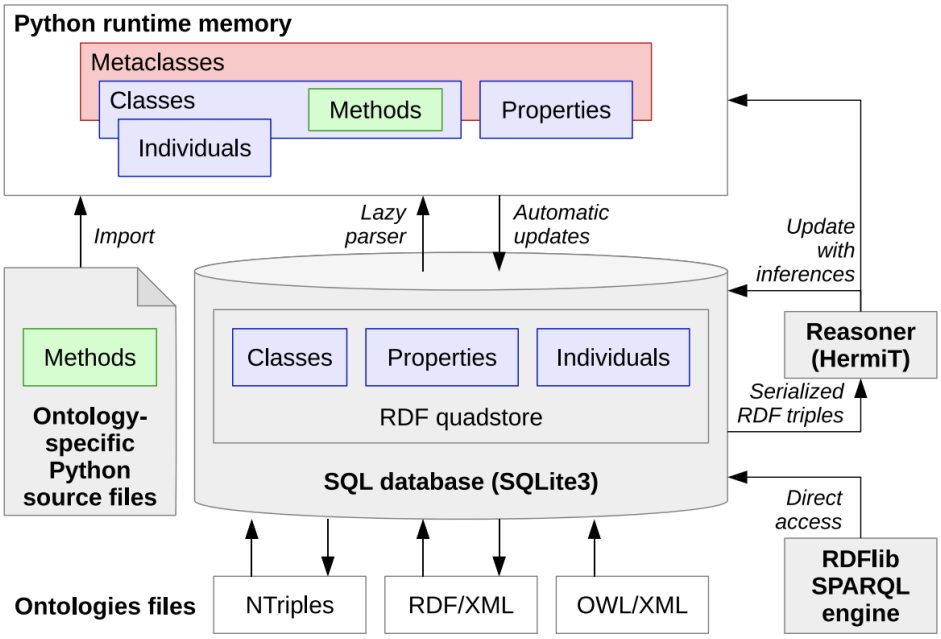



Figure 11: OwlReady Architecture (Lamy, 2017)

### 4.6.3 *Ontology Development*

During the analysis of sample documents, an initial ontology (Section 3.2.2) was developed, according to the OntoDL notation and exported as an OWL file.

Although the OWLReady toolkit offers the possibility of loading the previously defined ontology, the original model lacks important definitions, including relational properties and class equivalences. Consequently, the entire ontology development process was revisited. Most of the base concepts are the same and the updated ontology was directly written in Python.

While revisiting the ontology definition, an important issue emerged. Should the proposed solution incorporate previously defined and available ontologies, or in opposition, define a new ontology from the ground up. Adapting pre existing ontology offers an easier integration process with the semantic web and saves development time. In opposition, design a new ontology provides additional flexibility.

In the original model, the set of personal relations and all the different life events registered were identified as the main concepts to be manipulated.

For personal and genealogical relations, a wide range of stable solutions are available. The Friend Of A Friend (FOAF) ontology[29] is a known standard ontology to describe personal relations.

Towards genealogical relations, both free and commercial solutions exist. Ancestry[30] is a commercial project, where users according to a membership fee and DNA testing, are able to access the services of this platform, which include a large genealogy, ancestry and family trees knowledge base. Free alternatives include the nonprofit organization FamilySearch[31] and the genealogy website WikiTree[32].

The WikiTree is a collaborative project, started in 2008, and the main goal is to develop a world wide family tree. Any user is able to insert new relations or modify the preexisting content, by providing sources. Additionally, this platform provides a read only API[33] to access the knowledge base.

Although the WikiTree project is extremely interesting, the anonymity nature of the child abandonment problematic provides vague descriptions, insufficient to truly built a genealogical tree for each foundling. Most documents simply reference a cousin or uncle, without mentioning the side of the family.

The proposed solution consisted in adapting a pre existing ontology, as described in the guide (Tan, 2015), where personal relations are directly defined, ignoring the side of the family.

A final note on the integration with existing genealogical repositories is the possibility of exporting the relations according to the GEDCOM[34] standard. This format was developed to specify and exchange genealogical relations. Most genealogical platforms, including WikiTree and GEDmatch[35], find matches and update their knowledge repository after uploading a GEDCOM file.

Additionally, integration solutions, including the Mix'n'match[36], provide a tool to match a particular dataset to the WikiData knowledge base.

29 Specification of the Friend Of A Friend Ontology. More information available at: http://xmlns.com/foaf/spec/, Accessed in August 2021
30 Official Ancestry webpage: https://www.ancestry.com/, Accessed in December 2021
31 More information available at: https://www.familysearch.org/, Accessed in August 2021
32 World wide family tree project. Official webpage available at: https://www.wikitree.com/, Accessed in August 2021
33 Details on the WikiTree project API available at: https://www.wikitree.com/wiki/Help:API_Documentation, Accessed in December 2021
34 Official GEDCOM specification website available at: https://www.gedcom.org/, Accessed in December 2021
35 More information available at: https://www.gedmatch.com/, Accessed in December 2021
36 Integration tool with the WikiData. More information available at: https://mix-n-match.toolforge.org/#/, Accessed in December 2021

#### 4.6.3.1 *Entity Hierarchy*

The entity hierarchy is structured accordingly to classes and subclasses, as expected by the OWLReady toolkit. The subclasses are used to further specify the superclass, for instance, a person is not only a person, but in the perspective of the institution, has the foundling role.

The following list describes the ontology domain, the entire hierarchy corresponds to the supported concepts in the ontology:

- *Person* - Generic concept to describe a person. Additionally, a person is furthered specified according to the following roles:
  - Genealogical roles:
    - *Son* and *Daughter* - Person with a parent in the system,
    - *Mother* and *Father* - Person with a direct descendant,
    - *Goddaughter* and *Godson* - Person with a godparent,
    - *Godfather* and *Godmother* - Person with a godchild,
    - *Granddaughter* and *Grandson* - Person with a grandparent,
    - *Grandfather* and *Grandmother* - Person with a grandchild,
    - *GreatGranddaughter* and *GreatGrandson* - Person with a great-grandparent,
    - *GreatGrandfather* and *GreatGrandmother* - Person with a great-grandchild,
    - *Sister* and *Brother* - Person with a sibling,
    - *SisterInLaw* and *BrotherInLaw* - Person with a sibling-in-law,
    - *SonInLaw* and *DaughterInLaw* - Person with a parent-in-law,
    - *MotherInLaw* and *FatherInLaw* - Person with a child-in-law,
    - *Husband* and *Wife* - Person with a partner,
    - *Widower* and *Widow* - Person with a partner that passed away,
    - *Niece* and *Nephew* - Person with a pibling in the system,
    - *Uncle* and *Aunt* - Person with a nibling in the system,
    - *Cousin* - Person with a cousin in the system,
    - *CousinInLaw* - Person with a cousin-in-law in the system,
  - Institutional role:
    - *Foundling* - Term to describe an abandoned person, left in the institution,
    - *Nurse* - Person that supports a foundling, usually receiving a fee for those services,

    * *Writer* - Person responsible for writing the documents,
    * *Priest* - Generic term to describe a religious, present in some life events and religious references such as the baptism of a person,
    * *Councilor* - Public office worker,
    * *InstituitionManager* - Manager of the institution,
    * *InstituitionWorker* - Generic term to describe a worker of the institution,

- *Gender* - To denote the gender of a person, divided in three main subgroups:
  - *Male* - To denote a male figure in the system,
  - *Female* - To denote a female figure in the system,
  - *Other* - To denote an unspecified or ambiguous person,
- *Layette* - Term to describe a set of belongings, usually the items left with the child during the abandonment,
- *Item* - Generic entity to describe a physical item entrance, for instance, each item that belongs to the layette,
- *Signal* - Term to describe an object or document left with a child, which contain an identifiable role. These elements can be divided in two groups:
  - *Written* - To denote a written document,
  - *Object* - To denote an identifiable object, for instance, a necklace with initials,
- *Event* - Term to describe important events in the perspective of the institution. These events are specified by the following types:
  - *Baptism* - Religious event that most foundlings have after entrance in the institution,
  - *EntranceInInstitution* - Event of entrance of a particular foundling in the institution,
  - *RegistrationInInstitution* - Registration process of a new foundling in the institution,
  - *Abandonment* - The event of abandoning a child, for instance in the foundling wheel or on the streets,
  - *LeavingTheInstitution* - Process of a particular foundling leaving the institution,
  - *Death* - Registrations of a particular person passing away,
  - *Adoption* - The adoption of a foundling,
  - *GeneralEvent* - Generic term to describe non defined events, which more information is available in the respective description,

- *Institution* - Term to define an institution, which are divided in two main groups:
  - *FoundlingWheel* - Older format of the institution, where the anonymous child abandonment was possible,
  - *Hospice* - Institutions that replaced the foundling wheel, which supported the abandoned children,
- *Location* - Term to describe all the different geographical references contained in the documents,

As the previous list suggests, the ontology is centered around people, personal roles and relevant life events.

An important note is the definition of gender as an entity. This decision was taken to simplify the class equivalence expressions. As an example, when a male figure has a godparent relation, regardless of the gender of the descendant, this person is reclassified as a godfather. In opposition, if this person was a female, the system should instantly classify them as a godmother. Additionally, since it is possible to be subclass of multiple classes, setting gender as a class facilitates filtering elements according to their classifications.

For all classes other than Person, the respective subclasses were defined as equivalent. The goal is to explicitly declare that each instance must be one of the available subclasses. For example, the class *Signal* has two subclass, the *Written* and *Object*. By declaring the equivalence, it is impossible to define an instance of *Signal* that is not one of two available options.

#### 4.6.3.2 *Relations*

Relation declaration according to the OWLReady specification are divided in two main groups, the ObjectProperties, to define relations between entities, and DataProperties, to associate attributes to entities.

The following list describes the ObjectProperties defined. The list is structured according to the properties of the relation.

- Symmetric relations:
  - *hasSibling* - Relation between two siblings,
  - *hasSiblingInLaw* - Relation between two siblings-in-law,
  - *hasPartner* - Relation to denote a pair of people as partners,
  - *hasCousin* - Relation to denote a people as cousins,
  - *hasCousinInLaw* - Relation to denote a people as cousins-in-law,
- Inverse relations:

  - *hasGodparent* and *hasGodchildren* - Used to denote the relation between godparents and godchildren,
  - *hasGrandparent* and *hasGrandchildren* - Used to denote the relation between grandparents and grandchildren,
  - *hasGreatGrandparent* and *hasGreatGrandchildren* - Used to denote the relation between great-grandparents and great-grandchildren,
  - *hasParent* and *hasChildren* - Relation to denote direct ancestor and descendent,
  - *hasParentInLaw* and *hasChildrenInLaw* - Relation to denote parents-in-law and children-in-law,
  - *hasPibling* and *hasNibling* - Relation between an uncle or aunt and their respective nephew or niece,
  - *providedBy* and *providerOf* - Relation between a foundling and their provider, for instance a nurse that raises a child,
  - *participatedIn* and *attendBy* - Relation to denote the attendance of a particular person in an event,

- Relations without additional properties:
  - *composedBy* - Relation to denote the set of items that constitute a layette,
  - *resides* - To denote a residency of a particular person,
  - *institutionalRelationWith* - An institutional relation with a person,
  - *hasObject* - Relation to associate a particular item, layette or signal to a particular person, usually a foundling,
  - *occuredAt* - Where a particular event occurred,
  - *locatedAt* - Relation to denote the location of an entity, for instance, the location of the institution,
  - *knows* - Generic relation between a pair of people.

As previously mentioned, the data properties denote a relation between a particular entity and an attribute, which does not contain a unique identifier, the URI. These relations are used to enrich each entity with additional information. The following list compiles the data properties available in the specification:

- *hasPersonalRole* - String format of the role of a particular person,
- *hasGender* - The gender of a particular person,
- *hasName* - The name of a particular person or institution,

- *hasId* - Numeric identifier of a particular foundling in the institution,
- *hasIDYear* - Year component of the hasId attribute,
- *hasDate* - Generic relation to define the date of an event or other temporal reference, which can be furthured specify:
  - *hasInitDate* - Initial date of a particular event,
  - *hasFinalDate* - The final date of an event,
- *hasLocationInfo* - Description of a particular location,
- *hasDescription* - Generic description for any entity,
- *hasFile* - Associate a particular file reference,
- *hasFolio* - Associate a particular document,
- *hasLifeStory* - Additional description of a particular person,
- *hasReferences* - Additional textual references to a particular entity,
- *hasRegulation* - Textual attribute to denote regulations and other general rules of the institution,

An important note is the date attributes were defined as a datetime object. Although most date references have no time reference, the Hermit reasoner in OWLReady does not support date only objects. As a result, all dates were defined as a datetime objects, automatically setting the time to zero hours, minutes and seconds.

#### 4.6.3.3 *Class Equivalences*

The class equivalence is a logical expression that sets the minimum requirements or the sufficient condition for a concept classification. These conditions are used by the reasoner to automate the reclassification process, according to the currently known attributes and relations of each concept.

In this particular project, every instance is initially set as generic as possible, since it is assumed that the newly added entity is unknown from the perspective of the repository. As an example a new person is always initially set as an instance of *Person*.

After an instance is updated with new attributes or relations, a sufficient condition may be met, consequently the reasoner is activated every time the knowledge repository is updated.

In order to illustrate the reclassification process, the evolution of a person instance is followed. Initially, the new person is defined as general as possible, simply as an instance of *Person*. The gender class contains two main equivalences: *Person & hasGender.value("Male")*

and *Person & hasGender.value("Female")*. By providing a gender attribute and activating the reasoner, the person is reclassified, the result is both an instance of *Person* and their respective gender. Similarly, the classification of *Mother* and *Father* have the following respective equivalences: *Person & Female & hasChildren.some(Person)* and *Person & Male & hasChildren.some(Person)*. When their descendent is detected by the reasoner, this person is also reclassified as a parent. The classifications *Person Male Father* and *Person Female Mother* are the possible outcomes of the followed example.

Over the remaining available classes, similar equivalence expressions were defined. By combining the defined ontology, the logical equivalences and activating the reasoner after insertion, the reclassification of concepts according to the known attributes and relation is automated.

### 4.6.4 *NLP Pipeline and Ontology Integration*

The natural language pipeline automatically annotates information from documents and the ontology manages the information in triple format. The remaining task is to integrate these separate components, link the results of the NLP pipeline to the correct ontology concept or attribute.

After processing a document through the Spacy pipeline, the entities, relations and respective labels sets are available.

The process of connecting the output of the Spacy pipeline and the ontology, consists in instantiate or load pre existing entities in the ontology, iterate over the newly extracted triples set and match the named label to the ontology concept constructor or relation predicate.

An important note is some entities require additional processing before the insertion process in the ontology. Since this project deals with text, ambiguity is an extremely common issue.

For instance, the entity recognizer is capable of identifying a date, but the ontology requires a valid datetime object. Since dates are written according to different formats, issues such as case sensitivity, misplaced or missing stopwords and written numbers, are not automatically parsed by the datetime object. Similarly, the identifier is written in different variants, for instance, *número 1*, *número um*, *nº 1* or *Nº um* are possible representations. The ontology expects an integer value.

Additionally, a set of words may be used to express a concept or relation with the same meaning. For instance, the event death in Portuguese may be written as *morte* or *faleceu*. Similarly, different ecclesiastical titles, such as priest or abbot, from the point of view of the ontology, are an instance of a generic religious figure.

The proposed solution to handle the parsing and ambiguity issues is a combination of different linguistic features from different packages.

Initially, the Spacy package provides a morphological analyser. This component predicts for each token the part of speech, grammatical gender or the grammatical number, among other attributes. These predictions are useful to process additional content, for instance, infer the gender based on the grammatical gender of the first name. Additionally, Spacy provides a lemmatizer, used to solve grammatical number issues and tense–aspect–mood of verbs.

The Hunspell was the second library used, which provides a spellchecker and an additional lemmatizer. The spellchecker is used to solve minor issues, for instance, spellcheck the name of a person or written numbers. In order to preserve the original content, personal names were not spellchecked during the cleaning process. But in the back-end of the gender guesser, the spellchecked form is used. Furthurmore, the additional token level lemmatizer is used as a backup tool. Since the Spacy lemmatizer takes in consideration the context of the word in the document, in the case of bad predictions, the Hunspell lemmatizer is used instead.

The functions responsible for parsing the attributes of concepts, combine all of the previously mentioned concepts. More specifically was developed a number, identifier, date and gender guesser.

In order to solve ambiguous terms, the proposed approach uses the Fuzzywuzzy library. This tool provides a similarity score for a word, or set of words, against a defined set. Initially, for each Ontology concept, a set of acceptable terms is defined, similar to the process of defining a set of synonyms. The similarity between the extracted entity and the set of synonymous is calculated. If the similarity is eighty percent or higher, the application accepts that term as valid and the lemmatisation occurs. This is the fundamental step for matching the extracted named entities from the processing pipeline and the respective class constructor of concepts in the ontology.

Fuzzywuzzy provides a range of matching algorithms[37], including the Levenshtein Distance, but the main algorithm used is the partial ratio. The classification is calculated according to the ratio of the most similar sub-string. This algorithm was chosen because some entities are multi token.

37 Article describing the available matching algorithms in the Fuzzywuzzy package. Available at: https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/, Accessed in April 2021

# 5

# WEB APPLICATION

The web application is the final component of the proposed solution, responsible for establishing the connection between the end user and the knowledge repository.

As previously noted in Section 4.6.4, the relation extraction pipeline and ontology are fully integrated. The main objective is to expand these components by incorporating them in a web application and effectively materialize a platform supported by a knowledge repository.

This chapter provides the description of the system architecture and culminates in a navigational example over the main webpages.

## 5.1 TECHNOLOGY SELECTION

A standard practice in web development is the usage of web frameworks, which provide additional resources, for instance database management or templates engine, within the same environment.

One of the main aspects taken in consideration is the integration with the previously developed components. Since the document processing pipeline and knowledge representation components were developed in Python, the development process was proceeded within the same environment. This decision mitigates potential integration and complexity issues.

Within a Python environment, well established web frameworks are available, including Django[1] and Flask[2]. Although both frameworks support the development of web applications, they provide different infrastructures and development processes. The framework selection is crucial, since directly impacts the development time frame.

The built-in functionalities provided by the framework, solve the necessity of developing core components from the ground up. Consequently, the main decisive factor for the framework selection was the amount of support provided.

1 Official Django framework page: https://www.djangoproject.com/, Accessed in June 2021
2 Official Flask framework page: https://flask.palletsprojects.com/en/2.0.x/, Accessed in June 2021

The main difference[3] between Flask and Django is Flask is considered a lightweight and easily extendable framework. In opposition, Django is a heavier full-stack framework. Flask includes a core developing tools with high composability, leading up to a flexible developing process. In contrast, Django is a heavier framework, where plenty of the ground work is already offered, resulting in a more restrictive developing process. The decision between these frameworks was reduced to a flexibility versus support comparison.

Since the this project is a convectional application, following the standard Model View Controller (MVC)[4] architecture and most of the operations are CRUD[5] manipulations, the restrictive developing environment with additional support provided by the Django framework benefits the developing process.

The following list describes important infrastructures and conventions provided by the Django framework, a summary of the additional support provided:

- *Built-In Object Relational Mapping (ORM)* - Available for a wide range of standard relational databases, including MySQL, PostgreSQL and SQLite. The ORM is responsible for the integration between the web application and the database management system. Includes methods to manipulate the database, instead of writing a new query for every CRUD operation. Additionally, simplifies the development of database models, forms and database migrations,
- *Apps Structure* - The project is structured accordingly to app components. The required organization reduces code duplication, increases app reuse and the scalability of the project,
- *Built in Template Engine* - Mechanism to inject variable elements in standard HTML page,
- *User Model* - Customizable User model with built in account management and authentication,
- *Admin Interface* - Customizable Admin page, supporting CRUD operations and statistics for every database model,
- *Security Features* - Provides security against known attacks and vulnerabilities, including clickjacking, cross-site request forgery (CSRF), cross-site scripting (XSS) or SQL injection. Additionally, forms with client side validation,
- *Sessions*, *messages*, a*syncronous handlers*, among other pre-built components.

3 Additional details on the main difference between the Django and Flask framework available at: https://www.imaginarycloud.com/blog/flask-vs-django/, Accessed in July 2021

4 Software design pattern to structure web applications. Additional details available at: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller, Accessed in October 2020.

5 Standard data manipulations, additional details available at: https://en.wikipedia.org/wiki/Create,_read,_update_and_delete, Accessed in October 2020.

## 5.2 WEB PLATFORM ARCHITECTURE

The application is semantic driven and by design must provide a user friendly interface to interact with the knowledge repository.

When an end user accesses an area of the application containing entities, the currently known attributes and relations are displayed via a web browser.

Additional details on design of semantic driven applications available at (Hyvönen et al., 2004; Shekhar et al., 2013).

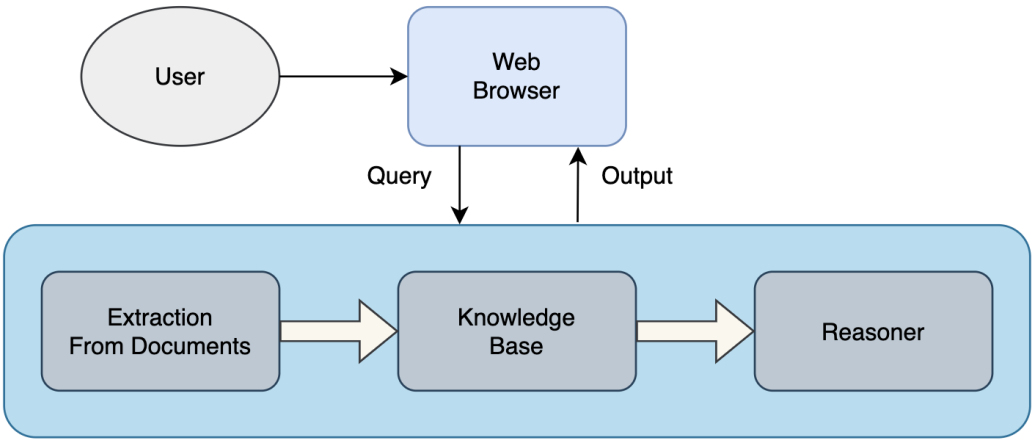



Figure 12: Interaction With the Knowledge Repository

Figure 12 describes the general architecture. The three main components match the processing pipeline responsible for the extraction of triples from text, then these triples become available to the application, accessed by querying the knowledge base.

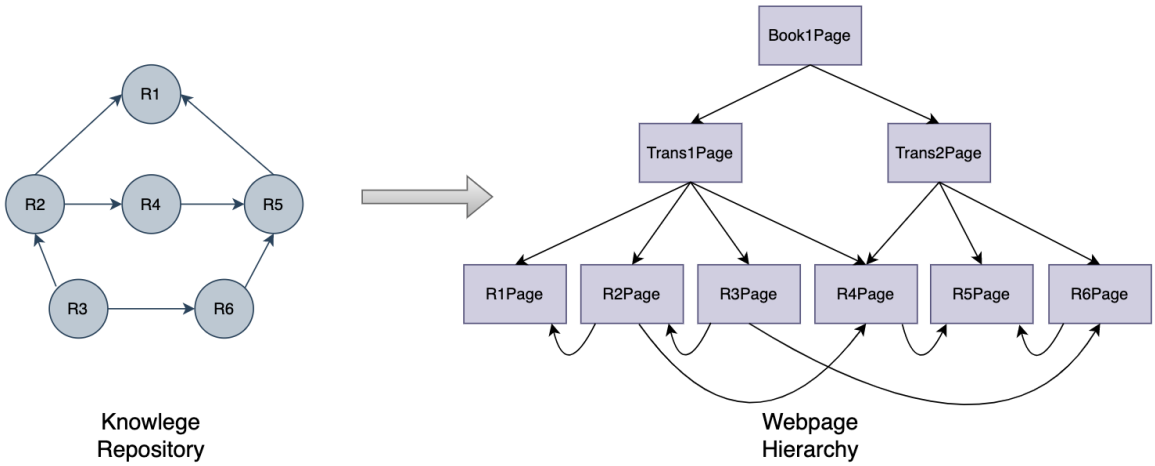



Figure 13: Resources Rendered as HTML Pages

As the Figure 13 suggests, by rendering the knowledge base, a network of pages emerge. Each resource is rendered as an HTML page, easily identifiable since each instance contains a unique identifier, the URI.

Relations between concepts are expressed as links between webpages, providing a presentation layer to navigate through knowledge repository. Additionally, Figure 13 describes a tree hierarchy between these webpages. Each book has its own page, which itself links to pages of that particular book, the transcriptions. Within each transcription page, instances of concepts are linked and the navigation through the knowledge repository is initiated.

The entire architecture of the application, from the information extraction to the interaction between the end user and the knowledge base, it is represented in Figure 14.

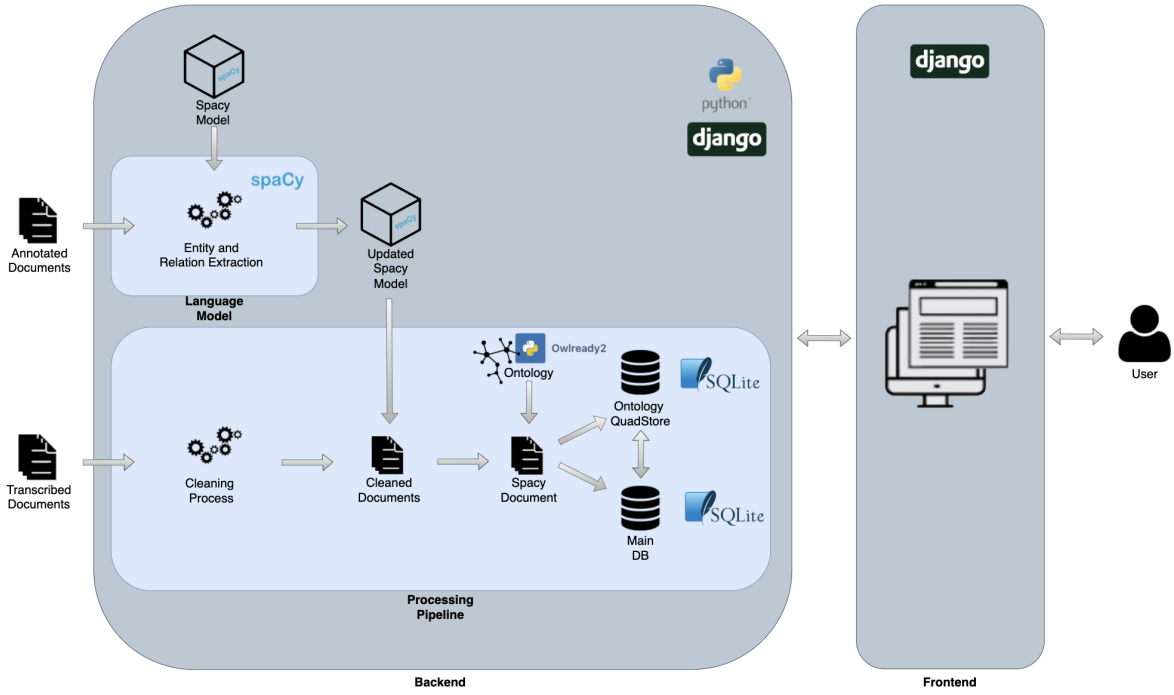



Figure 14: Updated Architecture

## 5.3 BACK-END COMPONENTS

This section introduces the main components of the back-end of the application, more specifically, focuses on the models and the storage solutions that support the knowledge repository. This section is concluded by analysing the integration of the processing pipeline in the web application.

### 5.3.1 *Models and Storage Solution*

The application requires the development of database models to define the digital archive. As analysed during the development of the datasets, as explained in Section 4.2.1, the documents are organized according to books and transcriptions. Additionally, the result of the processing pipeline is permanently stored in a knowledge repository.

The proposed solution preserves a separation between the knowledge repository and the remaining components. The OWLReady manages an independent database, where the World instances with all the concepts and relations reside. A second storage solution was developed to support the application, where instances of Books, Transcriptions and Users are stored.

#### 5.3.1.1 *User Model*

The application supports the standard CRUD operation, as a result, the content manipulations requires protection. Without any control, a malicious user is able to delete or wrongly modify the content preserved.

The proposed solution consists in the development of users restrictions in the application, limiting the content manipulation to a restrictive set of users with a specific set of permissions. A regular user is only able to read the available content.

The login process is a standard email and password procedure, and the additional permissions are set by the admin. A standard new account is set as an active account, without additional permissions.

The following list describes the User model:

- *id* - User identifier,
- *email* - Email of the user,
- *username* - Username of the user,
- *password* - Password of the user,
- *date_joined* - Account creation date,
- *last_login* - Last logged date,
- *is_admin* - Has admin permissions,
- *is_active* - Has active account,
- *is_staff* - Has staff permissions,
- *is_superuser* - Has superuser permissions,

#### 5.3.1.2 *Book Model*

The Book model is an extension of the descriptive attributes previously identified during the development of datasets, as explained in Section 4.2. The additional attributes are standard management information, including publishing date and poster. Furthermore, the model supports the addition of a picture of the physical book.

The following list describes the Book model:

- *id* - Book identifier,
- *titulo* - Book title,
- *cota* - Book identifier,
- *termo_abertura* - Open notes,
- *termo_encerramento* - Final notes,
- *observacoes* - Observations,
- *datas_extrema_inicial* - Date of the oldest document,
- *datas_extrema_final* - Date of the newest document,
- *dimensoes* - Dimensions,
- *image* - Cover Image,
- *date_published* - Publishing date in the platform,
- *date_updated* - Latest update date,
- *author* - Original poster,
- *modified_by* - User responsible for the latest update,
- *slug* - Resource identifier.

#### 5.3.1.3 *Transcription Model*

The Transcription model describes each document contained in a book. The Transcription model is similar to the Book model. The main difference is a foreign key to link the transcription to the respective book. Additional, the attribute *transcription_clean* stores the result of the processing pipeline and the *onto_marks* attribute links entities found in the document, known instances stored in the knowledge repository.

The following list describe the Transcription model:

- *id* - Transcription identifier,
- *folio_num* - Archival identifier,
- *transcricao* - Original transcription,
- *transcricao_clean* - Transcription after processing,
- *transcricao_lateral* - Side notes,
- *observacoes* - Observations,
- *onto_marks* - Links to referenced entities,
- *author* - Original poster,
- *image* - Picture of the document,
- *livro* - Book foreign key,
- *date_published* - Publishing date,
- *date_updated* - Last update date,
- *modified_by* - User responsible for the latest update,
- *slug* - Resource identifier.

Figure 15 shows the Entity Relation Diagram (ERD) of the main database:

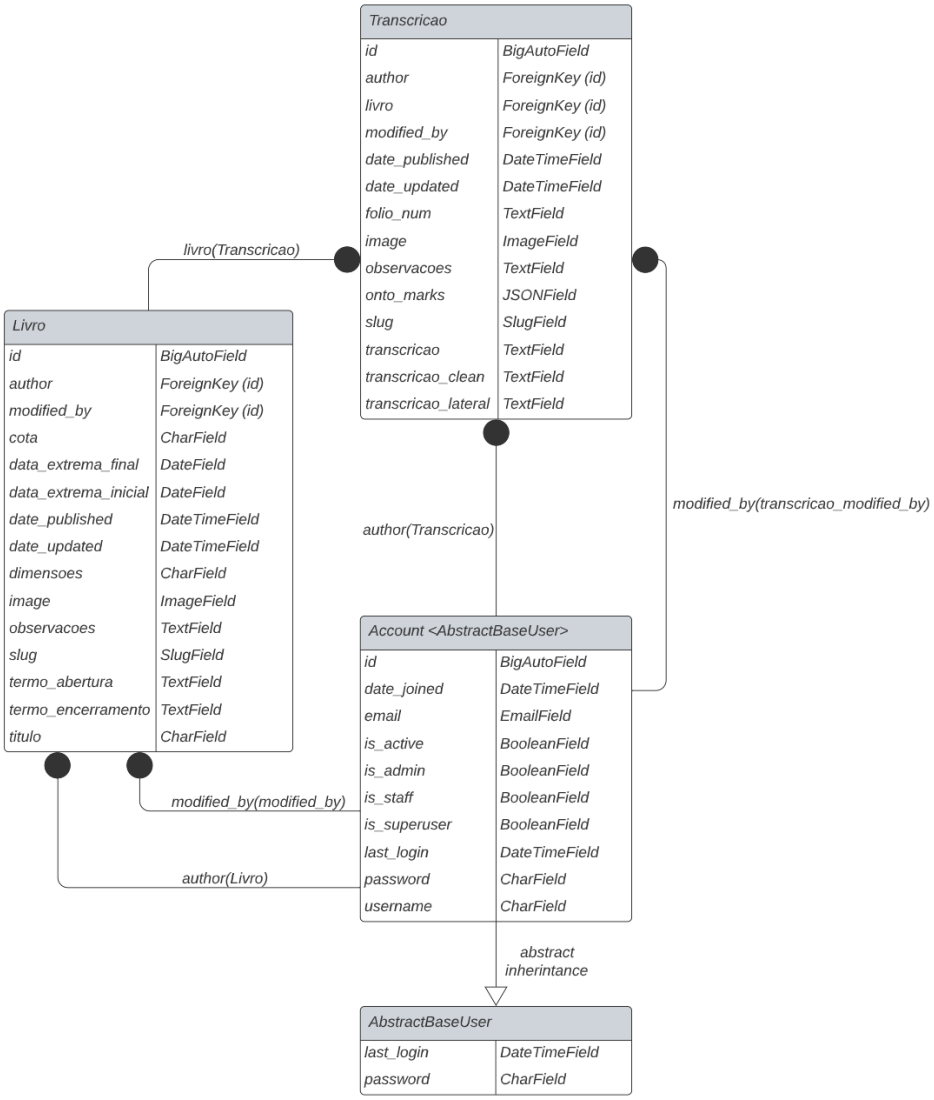



Figure 15: Main Database ERD[6]

The content is effectively stored in a SQLite database. The main decisive factor was the integration with the Django framework, which provide a built in ORM. This solution supports the current version of the project. In case of outgrowth, Django provides the additional tools to migrate to other compatible storage solution.

#### 5.3.1.4 *Knowledge Representation*

The second datasbase is the triple storage solution, where the entities and relations are permanently stored. The database model is the ontology concept hierarchy. Figure 16 captures the entire ontology conceptual model:

6 ERD rendered with a Django extention. More information available at: https://django-extensions.readthedocs.io/en/latest/graph_models.html, Accessed in November 2021

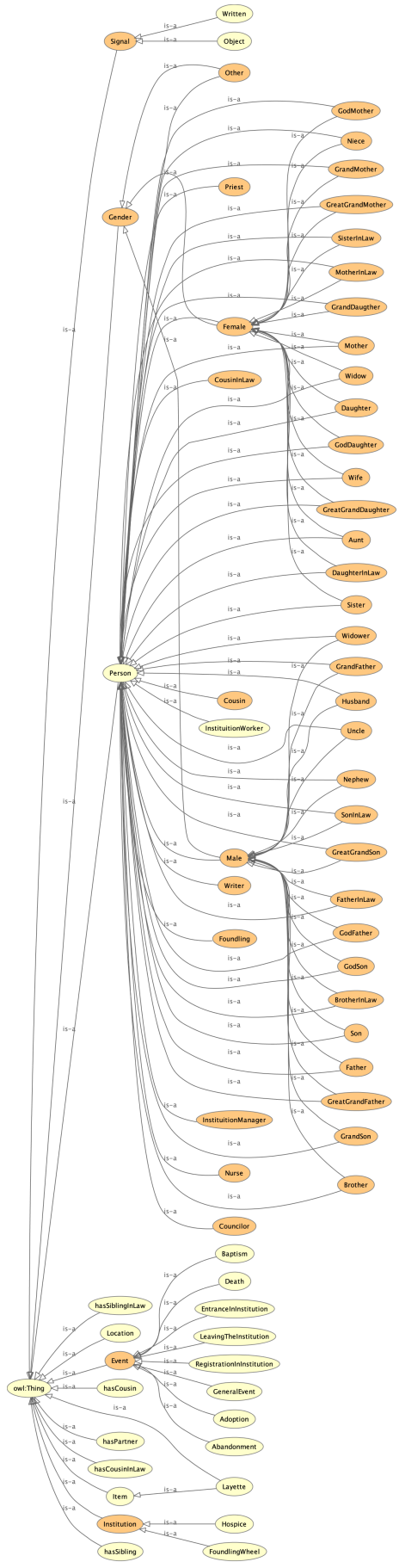



Figure 16: Ontology Concept Hierarchy

To conclude the model description, the application is supported by two databases. The integration between these seemingly independent repositories, requires the combination of previously developed components.

In the following section, the processing pipeline integration in the final application is described. This analysis is fundamental to comprehend the link between the knowledge repository and the main database.

### 5.3.2 *Pipeline Integration*

The integration of the processing pipeline in the web application is described by following the insertion of a new transcription. This example is followed since processing a transcription requires the combination of most components.

A new transcription is instantiated, for instance, via the standard form. The current state of the transcription, with no additional processing, it is stored in the *transcricao* attribute, according to the Transcription model.

The original transcription is processed as explained in Section 4.2.2. The cleaned version, after removing common issues and spellchecked, it is stored in the *transcricao_clean* attribute.

The clean transcription is the input for the natural language pipeline, where the entity (see Section 4.3.3) and relation extraction occurs, as discussed in Section 4.4. The ambiguities (for details, see Section 4.5) are solved and the extracted relations are temporarily kept in an object.

The extracted information is ready to be stored in the triple storage. The extracted relations are matched (remeber the explanation in Section 4.6.4), accordingly to the ontology model. Since the World was updated, the reasoner is activated, and the inferred content is stored.

A dictionary object is returned, containing the URI of each entity, the named entity label and the character index in the transcription text. This object is stored in the *onto_marks* attribute. This attribute assumes a foreign key role, by linking instances in the triple storage and the transcription.

After the entire processing, the information is available without additional processing. Accessing a transcription and the extracted relations is a read only process from the databases.

Figure 17 describes the entire transcription processing as a pipeline:

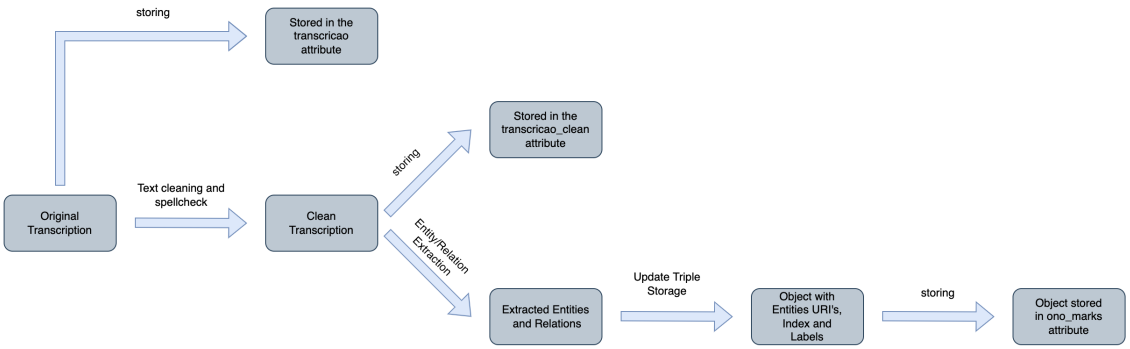



Figure 17: Transcription Processing Pipeline

## 5.4 FRONT-END COMPONENTS

After introducing the architecture of the web application, highlighting the integration of the data extraction and semantic components in the back-end, the following analysis focuses on the front-end, the client side of the application.

### 5.4.1 *Technologies*

The front-end component of the web application is fairly conventional. The different web pages are rendered accordingly to the standard Django template engine. A possible alternative is the development of an independent front-end engine to manage the client side, but the standard template engine satisfy the requirements of this project. Most pages are fairly static and the integration between the views and the page templates is fairly seamless, since the same underlying technology is used, the Django framework.

Most templates simply inject in the standard HTML the variable content. Only a small pool of pages required an additional Javascript, for instance, to open and close images and to render a small preview of the uploaded image in the forms, before submission.

An important note is besides the standard Django template, the bootstrap framework was used to handle the stylist features of each page, instead of developing an entire styling infrastructure from scratch.

## 5.5 REQUIREMENTS OF THE APPLICATION

During the process of outlining the main goals and the required components to successfully accomplish them, the use cases of the application started to emerge.

In order to substitute the physical access, the digital platform must simulate this process in a digital format. A user is able to see the all the available books and read them, effectively iterate through all the available transcriptions contained in a particular book.

Furthermore, the knowledge base must be accessible. For each entity, a page displays their respective relations and attributes in a readable format. The entities referenced in a page are linked to their own page. By developing a network of linked pages, the access to the knowledge repository is provided.

Towards the possible end users of the application, generally two types of users were identified. A technical user base, for instance archivists or historians, and the general public, which may use the application for recreational purposes. Since these different groups of users have different motivations, the design of the application should reflect that, providing an equilibrium between technical functionalities and ease of use. As a result, the base design of the website is fairly conventional, inspired by popular social media platforms. A navigational bar is available on the top of every page, where important links and a search bar are included. Additionally, the technological nature of the digital platform, should improve the access to the information, by offering search, filtering and sorting functionalities.

An important note is the restrictions introduced in the application. Although the general public may know additional information currently not present in the knowledge base, for security reasons, the modification of the knowledge base is restricted to a small pool of users with special permissions. Opening the knowledge base for everyone requires monitoring resources, for instance moderators willing to analyse the modifications performed by the general public. For these users with special permissions, it is granted the access to additional buttons and respective forms, which perform the standard CRUD operations. In the background, the entity and relation extraction components are activated and automatically insert the new content in the knowledge base.

Since some operations require authentication, the application contains the standard user management pages, including log in/log out, create a new account, update personal information and the mechanism to reset the password via email, in case the user is locked out of the account.

In order to compile the supported functionalities, the following list describes the main requirements. The agreed functionalities were achieved after a group of meetings with the archivists. These meetings were essential to outline the main features of the application, while taking in consideration the time and resources constraints.

- User management pages:
  - Create a new account,
  - Log-In by providing the credentials (email and password),
  - Log-Out from the current session,

  - Update personal information (username and email),
  - Password reset via email,

- Functionalities for users with additional permissions:
  - Insert a new book and transcription via a user friendly interface,
  - Update a preexisting book or transcription,
  - Delete a preexisting book or transcription,
  - Insert information to the knowledge base,

- Functionalities available to all users, including users without an account:
  - Page with the recent inserted and updated content (feed page),
  - Page with the entire archive, listing all the books contained in the archive,
  - Universal search bar over the attributes of each book and the content of the transcriptions,
  - Book page, with all the attributes that constitute a book and listing all the available transcriptions contained in each book. Additional a picture of the original book may be uploaded,
  - Read a book, by opening a book and within the book page, open each of the transcription,
  - Filter books from the entire archive, by choosing books where the content is in between a set of dates, and the results,
  - Page for each transcription of the archive. The original transcription and the cleaned version after the processing pipeline are available. Also, a picture of the original document may be uploaded,
  - Highlight the entities contained in each transcription, by developing a clickable component in order to access the page of each entity,
  - Page for each entity, where all of the related terms accordingly to the knowledge base are available in a human friendly format,
  - Semantic navigation, by clicking on the available entities displayed, which constitutes on iterating through the different edges of the knowledge graph,

## 5.6 NAVIGATION EXAMPLE

To conclude the web application description, some of the most important pages are introduced. The application, from where these pages were taken, it is accessible via the following link: `http://arquivoexpostos.epl.di.uminho.pt/`.

The platform contains a feed page, where the recently published and updated documents are displayed. Figure 18 shows that on the top of each page, a navbar provides a search box and links to the most important pages. In this particular example, the body of the page includes the feed.

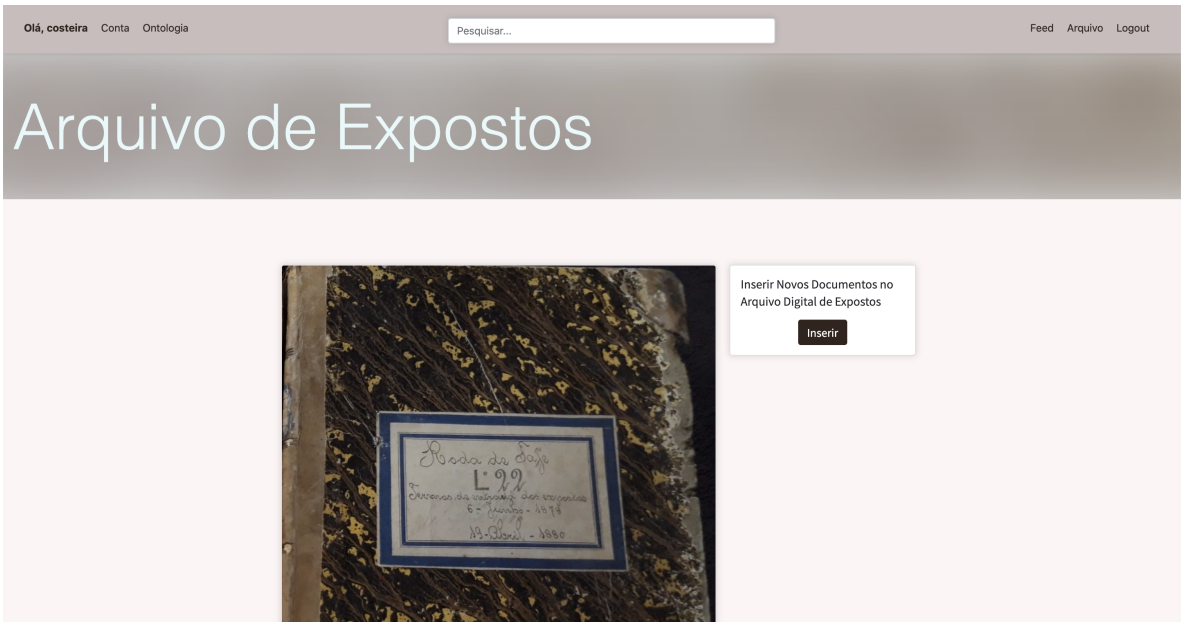



Figure 18: Feed Page of the Website

The feed displays each book similarly to a post on a social media platform. These posts are chronologically sorted, as Figure 19 suggests. Every time the content of the application is updated, these modifications are reflected in the feed.

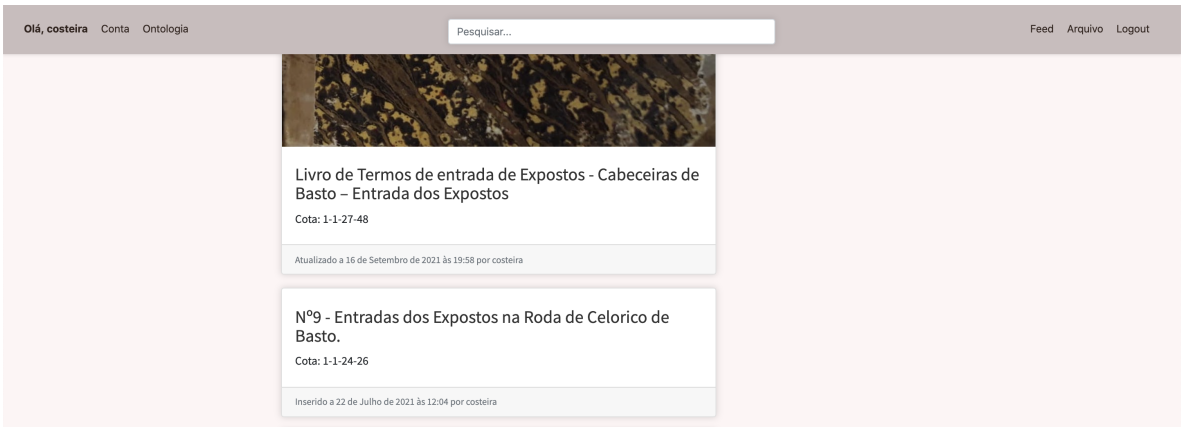



Figure 19: Feed With Sorted Content

The application includes a second central page, where all the currently available books are displayed in a tabular format. The book table is paginated, simply to avoid an extremely long and scrollable page. Additionally, the search bar and filter component offer the possibility of searching for a specific book and filtering the shown results, accordingly to dates.

The main objective is to simulate a physical access, where a user is able to choose a particular book and then read it.

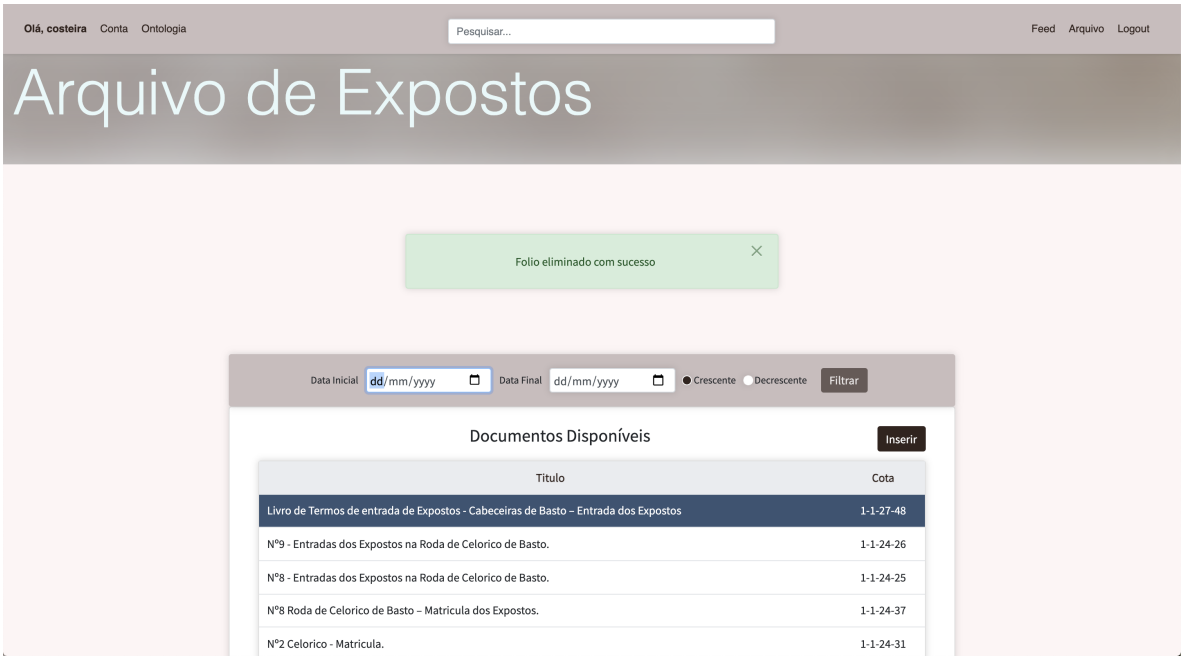



Figure 20: Table With All Available Books

As Figure 20 suggests, each row on the table is a book. Theses rows are links to the respective book webpage.

The book and transcription pages are fairly similar, composed of a group of cards, where important information is displayed. In the book page, the main card describes each book accordingly to their attributes, defined by the book model. The second card is a table, where each row is a link to a transcription contained in this book. The transcription page displays both versions of the transcription, the original version and the result of the processing pipeline.

Furthermore, in the presence of an image, a clickable thumbnail and a full page view of the physical document is available. For users with permissions, the buttons to update or delete the document are displayed at the bottom of each page, redirecting them to the respective form.

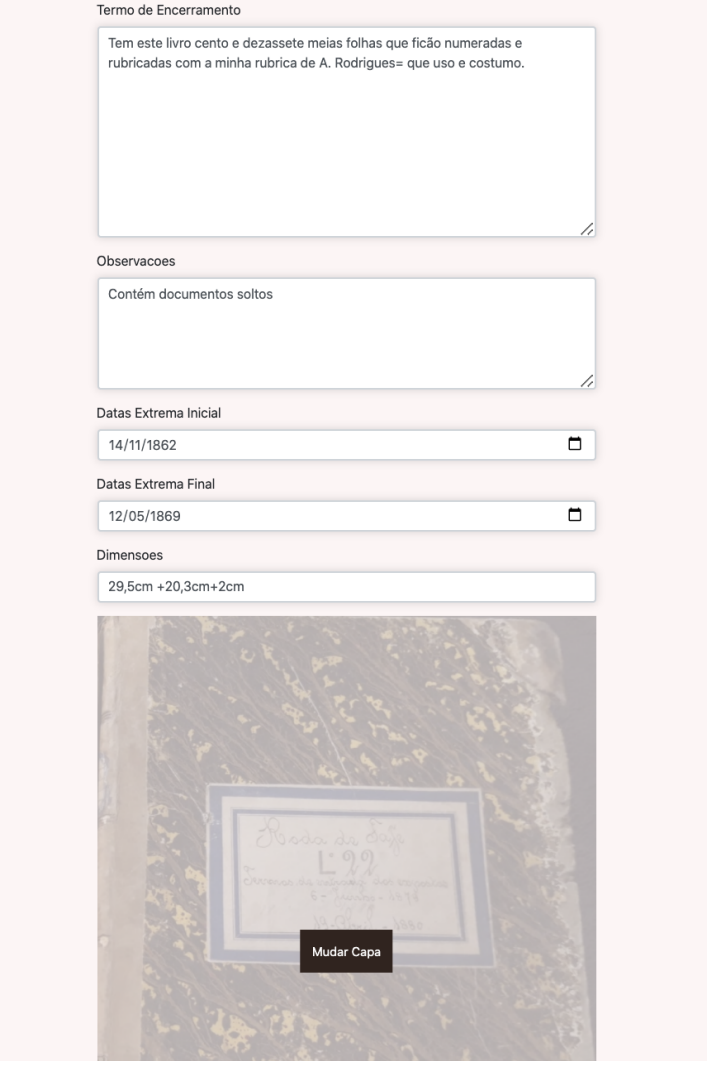



Figure 21: Book Update Form

In the transcription page, the knowledge repository is accessible. Similarly to a Wikipedia page, where within the text other pages are linked, the transcription references known entities.


Figure 22: Marked Entities

As Figure 22 suggests, entities are highlighted, with the respective label on the right hand side, written in all caps. These elements are color coded, following a similar approach to an annotation tool. An important note is in order to maintain the readability of the transcription, only entities with relations are highlighted. Additionally, attributes in the knowledge repository, data properties, are not highlighted.

The colorful blocks are links to other webpages. In these entity pages, known attributes and relations are displayed. For instance, Figure 23 displays the personal information of *Guergório*, the foundling referenced in the transcription shown in Figure 22.

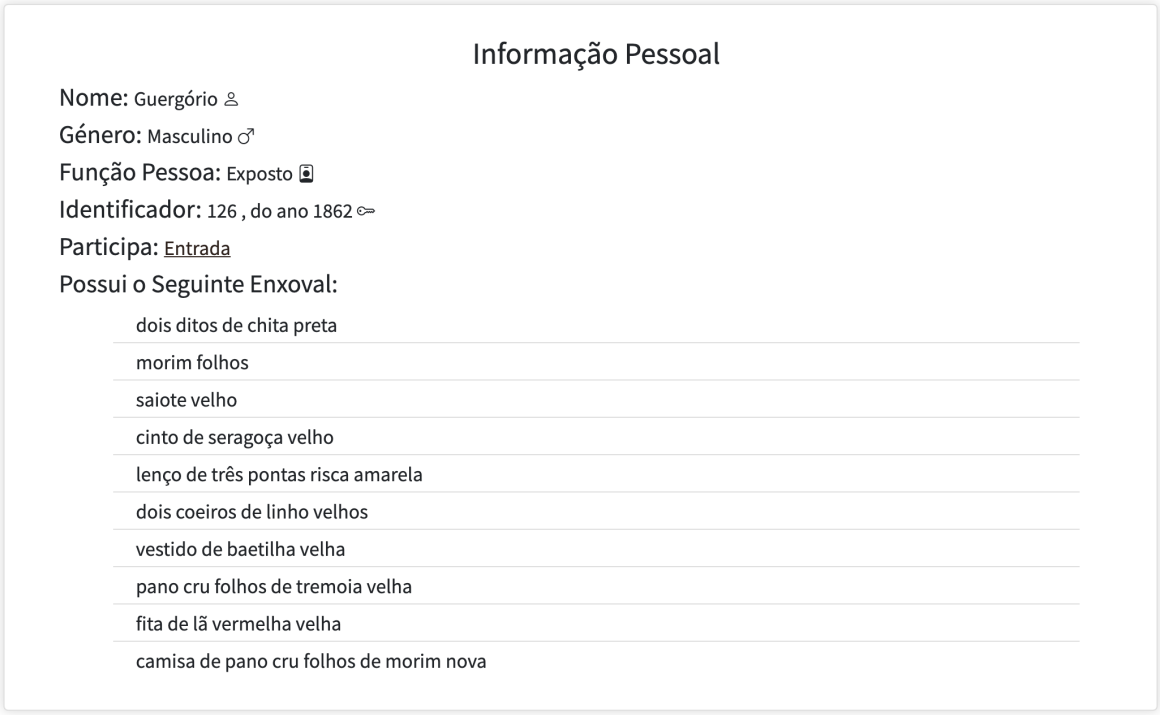



Figure 23: Personal Information Example

Linked entities are underlined in the cards displayed. As seen in Figure 23, the foundling *Guergório* participated in one event, his entrance in the institution. By following the event link, a new card with the event details is displayed. Figure 24 shows the resulting card.

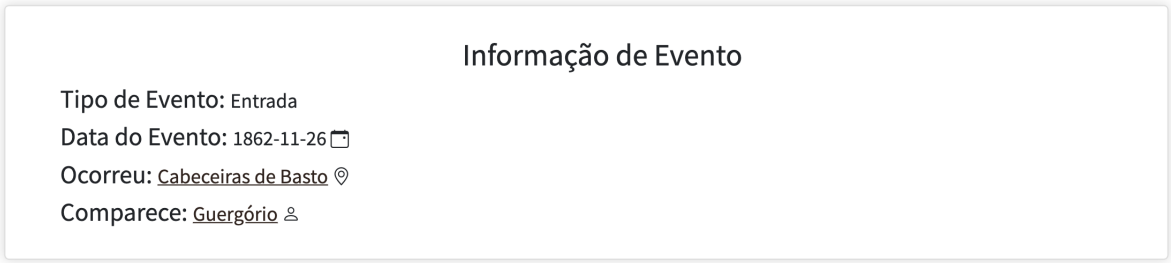



Figure 24: Event Page Example

## 5.7 SUMMARY

This analysis over the main webpages of the application, incorporates the essential use cases of the application. A user is able to simulate a physical access, by choosing a book and iterate through the available transcriptions. The transcriptions have an important role, as an entry point to the knowledge base. Entities are linked and displayed in the respective web page. Additionally, the application provides the standard features expected in a modern website, including user sessions, search functionalities and user friendly pages to update the knowledge repository.

# 6

# CONCLUSION

This document discusses the development of a web platform, focused on the foundling wheel domain.

Archivists and collaborators of the Municipal Archive of Fafe, have been preserving and digitizing historical documents, from different collections and domains. This process preserves unique records, filled with cultural significance, beyond the physical format.

One of these collections is the foundling archive, which includes documents dated to the nineteenth century, from historical institutions that supported the exposed children. These documents are the surviving records from foundling wheels and hospices dispersed across the Northern Region of Portugal.

Although at a first glance the foundling domain may seem insignificant, since these institutions no longer exist, the anonymous child abandonment is a problematic that societies across the globe are facing till this day. A recent increase in the abandonment numbers is pressuring countries to reintroduce mechanisms to handle this problematic. These new solutions essentially are a new iteration of the foundling wheel, simply replacing the former wheel and bell by sensors and cameras.

The challenge of developing a web platform, which simultaneously preserves and disseminates these mostly unique records, it is an important and fulfilling project.

The research hypothesis is the possibility of developing a digital platform supported by a knowledge repository, from the identification of concepts and respective relations, contained in documents concerning foundlings. The nature of the knowledge repository, offers an exciting technical project, by automating the content extraction via natural language processing.

The development of this project verifies the research hypothesis, and this document, describes the entire process.

Initially, the anonymous child abandonment problematic was contextualized. Children are one of the most vulnerable groups, and as a result of the abandonment process, some unsupported children unfortunately die. In order to overcome this issue and effectively save lives, centuries ago the foundling wheel was established. Theses institutions provided a safe

abandonment process, without compromising the anonymity of the perpetrators. After the abandonment, these children were institutionalized.

Although these institutions raised the foundling population, the poor living conditions provided, amplified the voices of critics and opponents, culminating in the termination of these institutions. Hospices replaced them and governmental support was redirected to families in need.

A recent resurgence of anonymous abandonment, lead to a reconsideration of the extinct foundling wheel. The baby box and the safe haven law are the most common solutions. Both essentially provide the main function of the foundling wheel, an anonymous abandonment without compromising the identity of the perpetrators nor criminal repercussions.

The application development was initialized with an analysis of sample documents. This process had the fundamental role of determining the main concepts and meaningful relations frequently found in these documents. The conceptualization defines the knowledge domain, the structure of the main repository. Life events and personal relationships are the main topics specified.

The Spacy natural language toolkit is the underlying technology used for automatically identify instances of concepts and relations, accordingly to the knowledge domain previously defined. This solution consists in a processing pipeline, composed of an entity recognizer and a relation extraction step. By processing a document, the desirable entities and relations are extracted, ready to be permanently stored in the knowledge repository. Prior to this processing, each document is cleaned, which removes additional punctuation and white spaces. Directly after, a document is spellchecked.

The knowledge repository management is supported by the OWLReady toolkit. This component permanently stores the extracted entities and relations in a database, accordingly to the ontology model. Additionally, this tools contains a query engine and a reasoner, used to reclassify and assert the consistency of the knowledge domain.

The final component is a web application, developed with the Django framework. This application incorporates all the previously developed components and establishes the link between the end user and the knowledge repository.

Each entity is shown in a separate webpage. These pages display known relations and attributes of a particular entity. Additionally, the books and transcribed documents are accessible, effectively simulating the act of physically grabbing a book and read it.

During the development process of this project, two main obstacles were faced. Automating the extraction of meaningful relations from documents was a difficult task. Linking distant entities in a document and solving ambiguities issues, set back the development of the relation extraction component.

The second main issue was the integration of the knowledge repository in the web application. During the integrating of these separate components, concurrency issues where found

while accessing the knowledge repository. The technical support and recommendations provided by the OWLReady development team, were essential to overcome these issues.

The main objectives of this project were successfully accomplished. The final application disseminates the archived documents. Additionally, the support provided by the knowledge repository, improves the access to important entities and relations extracted from documents. The direct access to the information, increases the usability of the application, by highlighting and interconnecting content dispersed across documents.

The main advantage provided by the proposed solution is the automation of the annotation and extraction of information from documents. The management of the knowledge repository was automated by the extraction pipeline and the usage of reasoners to reclassify the information, every time the repository is updated. In the point of view of the end user, processing information was fully automated, the only requirement is providing a new document in a pure textual format. In the back-end, the system automatically handles the entire processing.

The platform is accessible in the following link: http://arquivoexpostos.epl.di.uminho.pt/.

## 6.1 Future Work

Although the application provides the expected functionalities, possible improvements, in the perspective of usability and new functionalities, were identified.

One of the most important improvements is the incorporation of an annotation tool in the client side of the application, similar to Doccano. Currently a user is able to see color coded entities, and by clicking them, the user accesses the webpage of that particular entity. By further developing this component, a user could drag a relational arrow and label undetected entities and relations. This user friendly interface would drastically improve the knowledge repository and unknowingly, these users develop larger datasets to train the entity and relation extraction models.

A second improvement is the development of an API and SPARQL end point in the application. These functionalities provide to a technical user base custom access to the knowledge repository, besides the standard use cases. Offering direct access to data manipulation may compromise the consistency of the entire knowledge repository, as a result, additional security measures are necessary.

The final improvement is the integration of the developed knowledge base with popular and established ontologies. Accomplishing the integration, this project is opened to a whole new set of exciting challenges, beyond the standard dissemination of the archive. As an example, the WikiTree platform contains a wide range of projects, currently under

development, over their world wide family tree. The Adoption Angels[1] project has the goal of helping adoptees finding their biological family. Since the foundling archive preserves genealogical information concerning the Portuguese foundling population, the content extraction and repository developed in this project, could be integrated in a similar project. This exemplary extension of this project, would provide a tool to study the ancestry, roots and known facts about families impacted by the foundling problematic.

1 Official Adoption Angels project page: https://www.wikitree.com/wiki/Project:Adoption_Angels, Accessed in December 2021

# BIBLIOGRAPHY


Coram and the Foundling Hospital. URL https://www.coram.org.uk/about-us/our-heritage-foundling-hospital.

Roda dos expostos. Abandonar um filho sem que ninguém saiba, Jan 1970. URL https://ionline.sapo.pt/artigo/663730/roda-dos-expostos-abandonar-um-filho-sem-que-ninguem-saiba-?seccao=Portugal_i.

Learn About The Foundling Hospital: The Foundling Museum, May 2018. URL https://foundlingmuseum.org.uk/about/our-history/.

Cristiana Araújo, Salete Teixeira, Diana Barbosa, and Pedro R. Henriques. Using Ontologies to plan Computer Programming courses for different levels. In Simon Polovina, Rubina Polovina, Neil Kemp, and Ken Pu, editors, *MOVE'2020: Measuring Ontologies in Value-seeking Environments*. Association for Computing Machinery, October 2020. to be published.

Vicky Baker. Drop-off baby boxes: Can they help save lives in the US?, Jan 2019. URL https://www.bbc.com/news/world-us-canada-46801838.

Caroline Brettell and Rui Feijó. Foundlings in Nineteenth-century Northwestern Portugal: Public Welfare and Family Strategies. In *Enfance abandonnée et société en Europe, XIVe-XXe siècle. Actes du colloque international de Rome (30 et 31 janvier 1987)*, pages 273–300. École Française de Rome, 1991.

Joan Cochrane and Goh Lee Ming. Abandoned babies: the Malaysian 'baby hatch'. *Infant*, 9 (4):142–144, 2013.

Teodoro Afonso da Fonte. *No limiar da honra e da pobreza: a infância desvalida e abandonada no Alto Minho (1698-1924)*. Universidade do Minho (Portugal), 2004.

Maria del Mar Roldan-Garcia and Jose F Aldana-Montes. A tool for storing OWL using database technology. In *Proceedings of the OWLED 2005 Workshop on OWL: Experiences and Di-rections, Galway, Ireland, CEURWS. org*, 2005.

Stephen Evans. The 'baby box' returns to Europe. https://www.bbc.com/news/magazine-18585020, Jun 2012. Accessed: October 2020.

João M. Sousa Fonseca, Maria João Varanda Pereira, and Pedro Rangel Henriques. Converting Ontologies into DSLs. In Maria João Varanda Pereira, José Paulo Leal, and Alberto Simões, editors, *3rd Symposium on Languages, Applications and Technologies*, volume 38 of *OpenAccess Series in Informatics (OASIcs)*, pages 85–92, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-68-2. doi: http://dx.doi.org/10.4230/OASIcs.SLATE.2014.85. URL http://drops.dagstuhl.de/opus/volltexte/2014/4561.

Al-Hakam Hamdan, Mathias Bonduel, and Raimar J Scherer. An ontological model for the representation of damage to constructions. In *7th Linked Data in Architecture and Construction Workshop*, 2019.

Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2. *The university of Manchester*, 107, 2009.

Eero Hyvönen, Markus Holi, and Kim Viljanen. Designing and creating a web site based on RDF content. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.

Jean-Baptiste Lamy. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80:11–28, 2017.

Catherine Panter-Brick, Malcolm T Smith, et al. *Abandoned children*. Cambridge University Press, 2000.

Joana Vieira Paulino. O abandono infantil na Lisboa da segunda metade do século XIX. *Revista de Demografia Histórica*, 35(II):101–134, 2017.

Maria João Varanda Pereira, João Fonseca, and Pedro Rangel Henriques. Ontological approach for DSL development. *Computer Languages, Systems & Structures*, 45:35–52, 2016. ISSN 1477-8424. doi: http://dx.doi.org/10.1016/j.cl.2015.12.004. URL http://www.sciencedirect.com/science/article/pii/S1477842415300270.

Randeep Ramesh. Spread of 'baby boxes' in Europe alarms United Nations. https://www.theguardian.com/world/2012/jun/10/unitednations-europe-news, Jun 2012. Accessed: October 2020.

Monica Shekhar et al. Semantic Web Search based on Ontology Modeling using Protege Reasoner. *arXiv preprint arXiv:1305.5827*, 2013.

Mee Ting Tan. *Building a family ontology to meet consistency criteria*. PhD thesis, Universiti Tun Hussein Onn Malaysia, 2015.