

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Automatic Collection of Biographical Data on the Web for Public Personalities

José Pedro Azeredo Lopes de Moura Paixão

Master in Informatics and Computing Engineering

Supervisor: Eugénio de Oliveira (PhD)

Co-Supervisor: Luís Sarmiento (PhD)

17th January, 2011

Automatic Collection of Biographical Data on the Web for Public Personalities

José Pedro Azeredo Lopes de Moura Paixão

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Rosaldo Rossetti (PhD)

External Examiner: José Moreira (PhD)

Supervisor: Eugénio de Oliveira (PhD)

11th February, 2011

Abstract

In the last decades we have been living in an era of ever-growing information. With the evolution of information technologies and communication, the human being has been generating more and more data and, likewise, is being overloaded with ever more data. Given that 80% of the available information in the world is in text format [Jac00], it makes sense to try to extract the information that matters to us by performing effective searches in these sets.

To provide search engines or any other system that uses information in text format with the ability to understand synonyms of public entities, first we need a new system able to acquire the relations between persons and roles / ergonyms. Person's roles / ergonyms can be seen as synonyms or words that refer to the same entity.

This thesis is focused in providing machines with the ability to understand relations between names and their respective professions / jobs / occupations. This was accomplished by building a prototype that can accomplish the collection of biographical data on the web for public personalities in a semi-automatic way.

Similar projects like *Freebase* perform this task through manual introduction of information as a community based effort. In our case, the main motivation for a semi-automatic system is due to the fact that we do not have the community nor the resources that these projects have. To implement a similar system for the Portuguese language, a semi-automatic system is needed because a totally manual system is unviable due to the lack of possible users.

The new system has three main components: a database where all relations and entity information are stored; an automatic information extraction system able to extract new relations along with the entity names and their binding roles; and a graphical user interface that allows the manual validation of information by an human operator. Additionally a biographical information retrieval system needs to acquire extra information like article abstracts from wikipedia, article links and photos.

Having the complete system specification, a prototype was developed with codename *OntoExtract*.

To evaluate *OntoExtract*, several tests took place. For these tests 30 persons volunteered and actively participated in them. All testers were able to accomplish the validations, and the results of their validations revealed that the main objective, acquiring person-role relations, was well accomplished with accuracy levels greater than 80%.

The developed work allowed us to conclude that *OntoExtract* implements most of the features and stands as a functional prototype that validates the proposed solution for the problem raised, which was of being able to, in a semi-automatic way, obtain biographical information for public entities.

Resumo

Nas últimas décadas, vivemos cada vez mais numa era de informação. Com a evolução das tecnologias de informação e comunicação, o ser humano tem vindo a criar cada vez mais dados e, do mesmo modo, também é cada vez mais "bombardeado" com dados. Dado que 80% das informações disponíveis no mundo se encontram em formato de texto [Jac00], faz sentido tentar extrair a informação que nos interessa e executar pesquisas eficazes nestes conjuntos de informação.

Para capacitar os motores de busca ou qualquer outro sistema que usa a informação no formato de texto com a possibilidade de compreender sinónimos de entidades públicas é necessário primeiro um novo sistema que estabeleça as relações entre pessoas e cargos / ergónimos. Cargos / ergónimos de pessoas podem ser vistos como sinónimos ou palavras que se podem referir a uma mesma entidade.

O objectivo principal desta tese enquadra-se na área de capacitação das máquinas com a possibilidade de compreender as relações entre os nomes e respectivas profissões / trabalhos / ocupações construindo um protótipo que pode realizar de forma semi-automática a recolha de dados pessoais na web para personalidades públicas.

Projectos semelhantes como o *Freebase* realizam essa tarefa, fazendo-se apenas a introdução manual de informações através de um esforço comunitário. No nosso caso, a principal motivação para um sistema semi-automático, deve-se ao facto de que não temos a comunidade, nem os recursos que estes projectos têm e, assim, implementar um sistema semelhante para o idioma Português passa por um sistema semi-automático porque um sistema manual é totalmente inviável devido à falta de possíveis utilizadores.

O novo sistema tem três componentes principais: uma base de dados onde todas as relações e informações relativas às entidades são armazenadas, um sistema de extracção automática de informação que é capaz de extrair novas relações com os nomes de entidades associadas aos seus cargos e uma interface gráfica para permitir a validação manual de informações por um operador humano. Além disso, existe um sistema de recuperação de informação biográfica para adquirir informações adicionais, como resumos de artigos da wikipedia, links de artigos e fotos.

Tendo a especificação completa do sistema, foi desenvolvido um protótipo com o nome de código *OntoExtract*.

Para avaliar o *OntoExtract* foram realizados vários testes, para os quais se ofereceram 30 pessoas que participaram activamente. Todos os voluntários foram capazes de realizar as validações que foram propostas e os resultados das suas validações revelaram que o objectivo principal, estabelecer relações pessoa-cargo, foi bem conseguido com níveis de precisão superiores a 80%.

O trabalho desenvolvido permitiu-nos concluir que o *OntoExtract* implementa a maioria das funcionalidades propostas e permanece como um protótipo funcional que valida

a solução proposta para o problema levantado, que era o de poder, de um modo semi-automático, obter informações biográficas de entidades públicas.

Acknowledgements

I would like to sincerely thank the people who kept me "on tracks" and to those who helped make this an interesting project.

First of all, I would like to express my gratitude to my thesis supervisor Eugénio de Oliveira for allowing this project to exist and for the always sapient words and advices I needed to ear in order to conquer the problem.

Secondly, I would like to thank my co-supervisor Luís Sarmento which was the best supervisor I could ask for, always striving with new great ideas and equally great solutions for found problems. I do not know where I would be without his motivational speeches.

Thirdly, I'm indebted to the people at Sapo Labs for all their support and many times precious help in solving many problems. Particularly I would like to thank Filipe Coelho, for helping me with my English by reviewing my document many times, José Devesas for the crucial Perl hints along the development and Jorge Teixeira for all the patience helping me deal with the server.

All the volunteers who participated on the tests, thank you for your time and comments, some of you in the weekend. All your feedback was of great value to improve and evaluate my work.

A special word of gratitude goes to my family and girlfriend, for all the encouragement in moments of greater fatigue.

Finally, to all my colleagues in this fight who spent countless nights and days next to me, thank you for the company in those coffee breaks.

All of you made this last semester a memorable one.

José Pedro Azeredo Lopes de Moura Paixão

*“Whoever ceases to be a student
has never been a student.”*

George Iles

Contents

1	Introduction	1
1.1	Context	1
1.2	Verbatim/Voxx	2
1.3	Motivation	3
1.4	Objectives	4
1.5	Structure of the Document	4
1.6	Summary	5
2	State of the Art	7
2.1	Entity Relation Extraction	7
2.1.1	Knowledge-Based Entity Relation Extractions	8
2.1.2	Social Network-Based Entity Relation Extractions	9
2.1.3	Other Methods	9
2.2	Entity Relation Extraction Applications	10
2.2.1	Automatic Biography Generators	10
2.2.2	Automatic Question Answering Systems	15
2.2.3	Expert Finding Systems	20
2.2.4	People Search Systems	23
2.3	Similar Projects	25
2.3.1	Freebase	26
2.3.2	KnowItAll	27
2.4	Summary	28
3	System Specification	29
3.1	Overview of the New System	29
3.1.1	Database	30
3.1.2	Automatic Information Extraction System	32
3.1.3	Graphical User Interface	38
3.2	Use Cases	38
3.2.1	Regular Access	41
3.2.2	Operation Access	42
3.2.3	Administration Access	44
3.3	Main Desired Features	47
3.3.1	Automatic Information Extraction System	47
3.3.2	Graphical User Interface	48
3.4	Global Architecture	49
3.4.1	Technologies	51

CONTENTS

3.5	Summary	51
4	System Implementation	53
4.1	Database and Ontology decisions	54
4.2	Biographical Information Extraction	56
4.2.1	Match Patterns	57
4.2.2	Name Extraction	60
4.2.3	Name Cleaning Functions	62
4.2.4	Name Similarity Functions	70
4.2.5	Certainty Calculation	73
4.3	Biographical Information Retrieval	76
4.4	Graphical User Interface	77
4.5	Implemented Features	82
4.5.1	Automatic Information Extraction / Retrieval System	82
4.5.2	Graphical User Interface	83
4.6	Summary	84
5	Tests and Results	87
5.1	System Module Evaluation	87
5.2	Results	89
5.2.1	Graphical User Interface	90
5.2.2	Information extraction	92
5.3	Summary	94
6	Conclusions and Future Work	97
6.1	Work Summary	97
6.2	Future Work	98
	References	101
A	Survey	105

List of Figures

1.1	Voxx website	3
2.1	Network Community Structure	9
2.2	ArtEquAkt's Architecture	12
2.3	Wolfram Alpha's home page	17
2.4	Example of XisQuê utilization	18
2.5	MITRE's Prototype Expert Finder "Data Mining" Example [May06]	22
2.6	TACIT Active Net [May06]	22
2.7	ZoomInfo person's profile example [P ⁺ 09]	24
2.8	ZoomInfo's people search example [P ⁺ 09]	25
2.9	Spock's people search example [P ⁺ 09]	26
2.10	Freebase's home page	27
3.1	Diagram that illustrates the sequence of search events	33
3.2	Certainty diagram illustrating sequence of discovered elements	37
3.3	Certainty update event diagram	38
3.4	Usage Groups Diagram	40
3.5	User Inheritance Diagram	40
3.6	Authorization Process Use Case Diagram	41
3.7	Normal User View Access Diagram	41
3.8	Normal User View of a Person / Organization / Role profile	42
3.9	Normal User edit of own profile	43
3.10	Operator User adding use cases	43
3.11	Operator User editing use cases	44
3.12	Operator User bind, unbind role and delete use cases	45
3.13	Editing nicknames, related links and references use cases	45
3.14	List and search all users use cases	46
3.15	Administrator's user management use cases	46
3.16	Administrator's search system management use cases	46
3.17	Global Project's Architecture	50
4.1	Generic representation of the Database connections	56
4.2	Fine Pattern Regular Expression Diagram	57
4.3	Example of Fine Pattern Match Diagram	58
4.4	Example of Fine Pattern Match Diagram for <i>Role</i> "presidente"	58
4.5	Example of result for Fine Pattern Match Diagram for <i>Role</i> "presidente"	58
4.6	Rough Pattern Regular Expression	59
4.7	Organization's name extraction regular expression	61

LIST OF FIGURES

4.8	Entity name extraction process general sequence of events	62
4.9	Regular expressions used in first stage <i>Person's</i> name cleaning	65
4.10	Second stage <i>Person's</i> name cleaning regular expressions	65
4.11	Final stage <i>Person's</i> name cleaning final verifications	66
4.12	Sequence of filtering stages in name cleaning	66
4.13	<i>Person's</i> name cleaner sequence of events	67
4.14	First stage <i>Organization's</i> name cleaning regular expressions	68
4.15	Final stage <i>Organization's</i> name cleaning final verifications	68
4.16	<i>Organization's</i> name cleaner sequence of events	69
4.17	First stage <i>Role's</i> name cleaning regular expressions	69
4.18	Final stage <i>Role's</i> name cleaning final verifications	70
4.19	<i>Role's</i> name cleaner sequence of events	70
4.20	<i>Nickname identifier</i> sequence of events	73
4.21	Logarithm of base fifty used to calculate increase in certainty	75
4.22	Generic Information Retrieval Function Diagram	77
4.23	Main OntoExtract's GUI page	77
4.24	OntoExtract's <i>Person List</i> page after searching for name " <i>Bill</i> "	78
4.25	OntoExtract <i>Person's Profile</i> page, in this case of " <i>Bill Gates</i> "	79
4.26	OntoExtract <i>Person's Profile</i> page, in this case of " <i>Teixeira dos Santos</i> "	80
4.27	OntoExtract's pop-up windows for some history graphs	80
4.28	OntoExtract <i>Person's Profile</i> page example with edit options available	81
5.1	OntoExtract's testing moments at <i>Sapo Labs</i>	90
5.2	Distribution of answers to the SUS questions	90
5.3	SUS results for each tester	91
5.4	Worst SUS evaluation values for each question	91
5.5	Percentage of validated <i>Persons</i> in OntoExtract after the tests	92
5.6	Graph comparing <i>Portuguese</i> an <i>Foreign</i> persons validations	94
A.1	Survey's first page	105
A.2	Survey's SUS questions page one	106
A.3	Survey's SUS questions page two	107
A.4	Survey's validation information for Portuguese <i>Persons</i>	108
A.5	Survey's validation information for foreign <i>Persons</i>	109
A.6	Survey's additional suggestions and comments	110

List of Tables

3.1	Graphical User Interface Specification	39
3.2	Information extraction feature list	47
3.3	Relation mapping feature list	47
3.4	Complementary information extraction feature list	48
3.5	View lists feature list	48
3.6	View specific profiles feature list	49
3.7	Edit specific entities feature list	49
3.8	Authentication system feature list	50
3.9	Graph view feature list	50
4.1	Fine Pattern person's name identification test	60
4.2	Rough Pattern person's name identification test	60
4.3	Lists of names that cannot appear in a <i>Person</i> name	64
4.4	Lists of names that can not appear in an <i>Organization</i> 's name	67
4.5	Percentage of accepted differences for each entity type	73
4.6	Information extraction feature list implementation status	82
4.7	Relation mapping feature list implementation status	82
4.8	Complementary information extraction / retrieval feature list implemen- tation status	83
4.9	View lists feature list implementation status	83
4.10	View specific profiles feature list implementation status	83
4.11	Edit specific entities feature list implementation status	84
4.12	Authentication system feature list implementation status	84
4.13	Graph view feature list implementation status	84
5.1	Examples of <i>Roles</i> and <i>Ergonyms</i> used for finding <i>Persons</i> / <i>Roles</i> / <i>Or-</i> <i>ganizations</i> with the defined news patterns	88
5.2	Number of entities found in the start <i>roles</i> / <i>ergonyms</i>	89
5.3	Aggregated Total Validation Results	93

LIST OF TABLES

Abbreviations

ADT	Abstract Data Type
AIRS	Automatic Information Retrieval System
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CORBA	Common Object Request Broker Architecture
DB	Database
GUI	Graphical User Interface
UNCOL	UNiversal COmpiler-oriented Language
WWW	<i>World Wide Web</i>

ABBREVIATIONS

Chapter 1

Introduction

With the evolution of information technologies and communication, the human being has been creating more and more data and, likewise, was also "attacked" with ever more data. [Jac00] In recent years there has been increasingly the need to take this data, organize it and turn it into information. Information is a data set that is properly organized.

Given that 80% of the available information in the world is in text format [Jac00], it makes sense to try to extract the information that matters to us and perform effective searches in these sets of information.

To be able to extract this type of information with quality from a text, it is necessary, first of all, to structure and organize it [Jac00].

The text-mining aims to search for relevant information and meaningful knowledge discovery from textual documents. This process involves a significant degree of difficulty, considering that the information is typically available in natural language, without regards to patterning or structuring of data. In general, this process is divided into three stages: pre-processing of Data, Analysis and Knowledge Extraction and Post processing.

1.1 Context

Text Mining emerged in the 1980s; it is generally defined as a process that uses methods for navigating, organizing and finding information in text databases written in natural language. With Text-Mining it is possible to, in an easier way, handle unstructured information such as news, texts on websites, blogs and documents in general.

Historically, the importance of Text-Mining gained momentum in the 1990s, with the growth of digital storage and the Internet (Web-Mining). At the same time, analysts began to notice the lack of text-mining tools to handle unstructured information environments.

Nowadays we are walking towards a semantic web where the understanding of information plays a crucial part. Systems that are able, for example, to understand questions or queries are starting to emerge.

To one day have a system that truly understands natural language and the meaning of a text one of the necessary steps is to understand synonyms in order to be able to understand contexts. In the Portuguese language as well as any other, in this context of understanding the meaning of what is written, we need to give importance to the interphrasal cohesion. By this we mean that it is important to understand all the connections between sentences to create a context and give meaning to the global text or group of organized sentences. To achieve that we need to take many things in consideration, we give the name of cohesion to the process which, on the surface of the text, explicitly allows you to resume previous information, and combine it with what follows it. [APAL10]

One of the hardest synonyms to understand are role / ergonym synonyms of an entity, since every person can be also sometimes referred on by their role / ergonym. This is where this thesis project comes in.

1.2 Verbatim/Voxx

One of the major projects that LIACC (Artificial Intelligence and Computer Science Laboratory) develops is Verbatim.

Verbatim is an application of information extraction focused on the collection of quotes from media sources available on the Web. It consumes emissions of on-line organs of various media (eg Lusa, JN, Público) and tries to find citations (direct or indirect) made by an open set of public personalities. After finding these quotes, Verbatim also tries to find the topic which the citation talks about. Verbatim has no human intervention: it extracts the citations and attributes them a thematic in a fully automatic way (the active topics in the media are also automatically identified). [ver]

The user of Voxx has at his disposal the ability to browse quotes uttered by various organizations on various topics and consult the record of past citations. Verbatim is assumed as a tool for the analysis of the information disclosed in the media and, more particularly, the views and opinions expressed by organizations that usually accompany these means. [ver]

Summarizing this, Voxx:

- extracts information from the web;
- finds quotes uttered by an open set of entities;
- automatically associates a topic to this quote.

Introduction



Figure 1.1: Voxx website

1.3 Motivation

Voxx is not able to automatically identify quotes of individuals by their roles or ergonyms.

One possible practical application of this thesis project is to provide Voxx with the referred ability. If it could be able to do it, much more information and quotes could be gathered and associated with the one individual. This is why the ability of understanding synonyms is so important, in this case, synonyms in terms of personality and jobs/occupations/positions that it refers to. Having this in mind it becomes clear that an automatic biography generator is necessary to capacitate Voxx with the ability to understand biographic synonyms.

This understanding opens a whole new field of possibilities, even in other related fields. This kind of system is needed for other purposes like giving support for intelligent search engines, support journalistic tools that identify the people in the news, support citation analysis tools like the *Sapo Voxx*, and so on. Idealistically we can say that providing the machines with the ability to understand synonyms is a necessary step to one day be able to create a human- machine communication close to natural language.

The main motivation for this dissertation's project is the fact that similar project's solutions, like *Freebase* [fre], that accomplish this task by having only manual introduction of information through community based effort, are not viable for our Portuguese language solution due to our lack of same community effort and resources.

Another motivational reason concerns the fact that we live in an information era. Being able to understand the biggest amount of information is being a step ahead. Information is power.

1.4 Objectives

The main objectives of this thesis are related to the gathering of relevant information for the better understanding of information.

In this particular case in the automatic collection of biographical data on the Web for public personalities, as well as establishing relations between the personalities. The information collected must have a certain amount of discernment.

The information gathered will be stored in a database which must be created. There is also the need to implement methods of extracting information from various sources like Wikipedia and news feeds or magazines.

At a final stage, an interface must be developed to allow a human actor to validate the information gathered. An interactive graph showing all the connections between entities should also be developed at a final stage.

This thesis project purpose is to capacitate Voxx system as any other that can use synonym information of public personalities concerning their jobs to better understand and retrieve online information.

In summary the main goal is to develop a self filling database that contains biographical information about public entities and their relations that could be used to integrate in Verbatim.

1.5 Structure of the Document

This document is organized in 6 different chapters.

Chapter 1, the current chapter, gives us an introduction, describes the Voxx Project, the projects motivation and main goals are also discriminated and the overall document structure.

The second chapter gives an overview of the state of the art of biographical information retrieval systems and some related systems.

Chapter 3 gives a general overview of the system specification, describing its use cases, the ways of gathering data, the technologies and explaining the way it is going to be implemented.

In chapter 4, the project implementation is described. All main algorithms and heuristics implemented are described.

The tests and results done for the validation of this project are described in chapter 5.

Finally, in the last chapter, conclusions about the presented project thesis are drawn. The future work is also addressed in this chapter.

Additionally an appendix containing the survey used for the test purposes is described.

1.6 Summary

In this chapter an introduction to contextualize this thesis project is done.

A description of the Voxx Project is done to understand where the project is to be incorporated and which needs are there to attain.

The projects motivation and main goals are also discriminated as well as the overall document structure.

Introduction

Chapter 2

State of the Art

This dissertation project's scope is framed in various extraction and retrieval systems and paradigms. The one that approaches the most to its aim is relation extraction systems, particularly entity relation extraction systems. This kind of systems focus mainly in acquiring relations between elements from unstructured text sources.

The Information extraction can be defined as the task of extracting information of specified events or facts, and then stored in a database for the users querying. Only with the correct relations between the various entities, the database can be correctly populated. Entity relation extraction becomes a key technology in information extraction systems.

In this section we will do an overview about the state of the art of similar projects. We will also talk about projects that solve part of the problem here described and finally do a small introduction about related projects. In this section we talk about projects that were not built for the Portuguese language, nonetheless we think it is relevant to talk about them.

2.1 Entity Relation Extraction

In the field of information extraction, entity is the basic information elements of the text, and it is the basis of proper understanding of the text [Zha07]. Narrowly defining, the entity is the concrete or abstract entities in the real world, such as person, organization, company, location, etc. Generally, it is expressed by unique identifiers (proper name), such as person's name, organization's name, company's name, location's name and so on. Broadly defining, the entity can also contain the time or other quantifiers. The exact meaning of the entity can only be determined by its specific application, for example, in specific applications fields like the address, e-mail, telephone number, ship number, conference name can be use as named entity. [CCH11]

Relation is seen as the link of two entities within a period of time or space [FS]. In the Research of information extraction, relation detection plays a key role in the identification and description of events. Thus, the extraction of semantic relations between entities is an important information extraction in the field of basic research. It is used in many research domains, such as information retrieval, question answering, ontology construction, information filtering, machine translation. [CCH11]

Before discussing the entity relation extraction, firstly we define what are relations and the classification of relations.

From a mathematical perspective, relation is a subset of the Cartesian product; from a computational perspective, relation is a two-dimensional table; from a logical perspective, relation is more than a binary predicate. [CCH11]

The formal classification of relations is more complex. According to the form we have many relation classifications, such as binary relations and multi-relations; grammatical relations, semantic relations and pragmatic relations; explicit relations and implicit relations. In accordance to the environment, the relation classifications are: web entity relations and plain text entity relations. From the pattern point of view of the relations, we have: pre-defined relations and non-pre-defined relations. Recent developments and research in the area normally use binary relations, grammatical relations, explicit relations, web relations and pre-defined relations. [CCH11]

For the entity relation extraction there are many techniques that are normally used. Some of the main ones are generally described in the next subsections.

2.1.1 Knowledge-Based Entity Relation Extractions

This method of extraction uses linguistic knowledge. Before the implementation of extraction, it constructs a pattern set based on words, speech or semantic, which are then stored in the database. During the relation extraction, the pretreated sentence fragment will try to match with the patterns in the pattern set. If there is a successful match we can conclude that this sentence fragment has the corresponding relationship property of the pattern. [CCH11]

Using the knowledge-based entity relation extraction method, the most difficult step is the construction of relation patterns. Initially, the construction of relation patterns depended on linguists, they analyse the related corpus to the extraction task in depth, using the existing linguistic achievements, enumerating every possible expression of relationship and finally constructing the relation pattern by hand. This method makes the period of constructing the pattern too long, and makes the application's cost too high. In reality this method is very difficult to realize. To solve this problem, several scholars have proposed different solutions. Douglas E. [AHB⁺95] proposed *FASTUS* extraction system

in MUC-6 evaluation contest, express a variety of domain-dependent rules in a extensible, common mode through the introduction of the "macro" concept. Roman Yangarber [YGTH00] proposed *Proteus* extraction system in MUC-7, the pattern constructing method of relations extraction in this system is based on sample generalization. [CCH11]

2.1.2 Social Network-Based Entity Relation Extractions

Numerous studies found that the nature of social networks is heterogeneous. It is not a large number of identical nodes randomly connected, but rather a combination of many types of nodes. There are more connections between nodes of the same type. Different node types have fewer connections between them. So the researchers defined these connections between similar nodes, represented by sub-graphs of the same social network, of network groups or communities, shown in Figure 2.1.

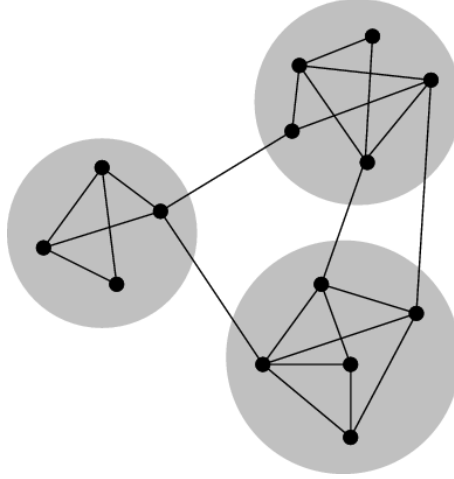


Figure 2.1: Network Community Structure

These sub-groups of nodes that have similar characteristics, according to the above characteristics of communities, can determine the semantic relations in the named entity structures and its similarity in network structure. [SCR03]

2.1.3 Other Methods

We also have methods that are based on machine-learning techniques. Feature-Based Entity Relation Extractions, as an example, which is a method that does not need to write knowledge rules by linguistic experts, only needs many samples used as training data, constructing a classifier with a variety of learning method, expressing the relation as a multi-dimensional vector. [CCH11]

Kernel-based methods can make use of many different forms of data organizations to express entity relations. while calculating the distance between the entities, it can use

kernel function other than the inner product of eigenvectors. Compared with the vector based method, the advantage of Kernel-based method is that it can express entity relation more flexibly, and colligate multi-disciplinary knowledge and information through the kernel function mapping. Zelenko proposed a machine learning method based on kernel functions for the relation extraction [ZG05]. He firstly defined the kernel functions based on shallow parse expressions on the text, and designed an efficient dynamic programming algorithm to calculate the value of kernel function. Secondly, used the support vector machine (SVM) and voting perceptions algorithms to achieve information extraction, the experiments showed that the kernel method has very good performance results. [CCH11]

Skounakis extracted three types of dual entity relations from the scientific literature with the model of HHMM [RY02]. Dan Roth proposed to identify the entity and the entity relation in the sentence with the means of probabilistic framework, and fully considered of interdependence between the entity and the entity relationship. [CCH11]

2.2 Entity Relation Extraction Applications

Entity Relation Extraction methods provide systems with entity relation information in an automatic way. Many systems need entity relation information and many times entity relation extraction methods to be able to fulfil their purposes.

The main purpose is to gather and organise in a data base the biographical information gathered through entity relation extraction so that other systems like Voxx can use it to help in semantic understanding of online information.

In this section we will do an overview of some of those systems and some examples of them will be described.

2.2.1 Automatic Biography Generators

Automatic Biography Generators are systems that in some way, as the name implies, gather biographical information from the World Wide Web in an automated fashion.

Most of these systems just gather the information with the purpose of generating biographical templates for human use.

Some of the most relevant projects in this area are projects like the ArtEquAkt project and the Web-Based Detection of Music Band Members and Line-Up project which we will describe next.

2.2.1.1 ArtEquAkt

The Artequakt project aims to implement a system that searches the Web and extracts knowledge about artists, automatically producing tailored biographies of artists.

The need for automatic knowledge harvesting tools is quickly increasing as the amount of knowledge available and spread across the Web has never been so large. Annotations on the Semantic Web could facilitate acquiring such knowledge, but annotations are rare and in the near future will not probably be rich or detailed enough to cover all the knowledge contained in these documents. Hence advanced knowledge services may require tools able to search and extract specific knowledge from the Web, guided by a domain conceptualization (ontology) that details what type of knowledge to harvest.

The Artequakt system deals with three main problems:

- Many Information Extraction (IE) systems rely on predefined templates and pattern-based extraction rules or machine learning techniques in order to identify and extract entities within text documents. Ontologies can provide domain knowledge in the form of concepts and relationships. Linking ontologies to IE systems could provide richer knowledge guidance about what information to extract, the types of relationships to look for, and how to present the extracted information.
- There are many IE systems that enable the recognition of entities within documents (e.g. 'Renoir' is a 'Person', '25 Feb 1841' is a 'Date'). However, such information is incomplete and of little value without acquiring the relation between these entities (e.g. 'Renoir' was born on '25 Feb 1841'). Extracting such relations automatically is difficult, but crucial to complete the acquisition of knowledge fragments and ontology population (building the knowledge base).
- When analyzing documents and extracting information, it is inevitable that duplicated and contradictory information will be extracted. Handling such information is challenging for automatic extraction and ontology population approaches.

The Artequakt project aims to implement a system that searches the Web and extracts knowledge about artists, based on an ontology describing that domain, and stores it in a knowledge base to be used for automatically producing tailored biographies of artists.

The figure 2.2 of page 12 illustrates Artequakt's architecture which comprises of three key areas.

The first concerns the knowledge extraction tools used to extract factual information items from documents and pass them to the ontology server.

The second key area is the information management and storage. The information is stored by the ontology server and consolidated into a knowledge base which can be queried via an inference engine.

The final area is the narrative generation. The Artequakt server takes requests from a reader via a simple Web interface. The reader request will include an artist and the style of biography to be generated (chronology, summary, fact sheet, etc.). The server uses story

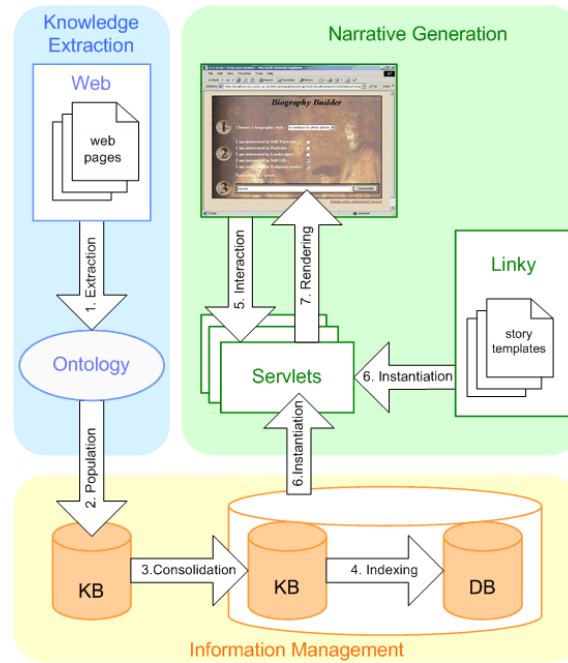


Figure 2.2: ArtEquAkt's Architecture

templates to render a narrative from the information stored in the knowledge base using a combination of original text fragments and natural language generation.

The first stage the Artequakt project consists of developing an ontology for the domain of artists and paintings. The main part of this ontology was constructed from selected sections in the CIDOC Conceptual Reference Model (CRM) ontology.

The ontology informs the extraction tool of the type of knowledge to search for and extract. An information extraction tool was developed and applied that automatically populates the ontology with information extracts from online documents. The information extraction tool makes use of an ontology, coupled with a general-purpose lexical database, WordNet and an entity-recognizer, GATE as guidance tools for identifying knowledge fragments consisting not just of entities, but also the relationships between them. Automatic term expansion is used to increase the scope of text analysis to cover syntactic patterns that imprecisely match our definitions.

The extracted information is stored in a knowledge base and analyzed for duplications and inconsistencies. A variety of heuristics and knowledge comparison and term expansion methods were used for this purpose. This included the use of simple geographical relations from WordNet to consolidate any place information; e.g. places of birth or death. Temporal information was also consolidated concerning the precision and consistency.

Narrative construction tools were developed in the Artequakt project to query the knowledge base through an ontology server to search and retrieve relevant facts or textual paragraphs and generate a specific biography. The automatic generation of tailored

biographies is done focusing in two areas. Firstly, providing biographies for artists where there is sparse information available, distributed across the Web. This may mean constructing text from basic factual information gleaned, or combining text from a number of sources with differing interests in the artist. Secondly, the Artequakt project aims to provide biographies that are tailored to the particular interests and requirements of a given reader. These might range from rough stereotyping such as "A biography suitable for a child" to specific reader interests such as "I'm interested in the artist's use of colour in their oil paintings". [SWPS07]

2.2.1.2 Web-Based Detection of Music Band Members and Line-Up

The Web-Based Detection of Music Band Members and Line-Up project is intended to automatically detect band members and instrumentation using World Wide Web content mining techniques.

To this end, this project combines a named entity detection method with rule-based linguistic text analysis.

Automatic extraction of textual information about music artists can be used, for example, to enrich music information systems, for automatic biography generation, to build relationship networks, or to define similarity measures between artists, a key concept in music information retrieval.

The Web-Based Detection of Music Band Members and Line-Up project is an approach to finding the members of a given music band and the respective instruments they play. In this project, since it still is in a preliminary stage, the instrument detection was restricted to the standard line-up of most Rock bands, i.e. it only checks for singer(s), guitarist(s), bassist(s), drummer(s), and keyboardist(s).

Basically, at this stage this project comprises four steps: web retrieval, named entity detection, rule-based linguistic analysis, and rule selection.

Concerning the web retrieval step, given a band name B, the Web-Based Detection of Music Band Members and Line-Up project uses Google to retrieve the URLs of the 100 top-ranked web pages, whose content is then retrieved via wget.

Trying to restrict the query results to those web pages that actually address the music band under consideration, the system adds domain-specific keywords to the query, which yields the following four query schemes:

- "B" + music (abbreviated as M in the following)
- "B" + music + review (MR)
- "B" + music + members (MM)
- "B" + line-up + music (LUM)

Discarding all mark-up tags, the system eventually obtains a plain text representation of each web page.

This is an interesting approach and very pertinent in comparison with my own thesis, because, as it will be explained further in this document, this kind of web queries is used in my web information retrieval systems.

Concerning the named entity detection in The Web-Based Detection of Music Band Members and Line-Up project a quite simple approach is followed. First, it extracts all 2-, 3-, and 4-grams from the plain text representation of the web pages.

This assumes no artist name is longer than four single names. Subsequently, some basic filtering is performed. It excludes those N-grams whose substrings contain only one character and retains only those N-grams whose tokens all have their first letter in upper case and all remaining letters in lower case.

Finally, it uses the *iSpell English Word Lists* [kev] to filter out those N-grams which contain at least one substring that is a common speech word. The remaining N-grams are regarded as potential band members.

Having determined the potential band members, it performs a simple rule-based linguistic analysis to obtain the actual instrument of each member. It defines the following rules and applies them on the potential band members:

1. M plays the I
2. M who plays the I
3. $R M$
4. M is the R
5. M , the R
6. $M (I)$
7. $M (R)$

In these rules, M is the potential band member, I is the instrument, and R is the role M plays within the band (singer, guitarist, bassist, drummer, keyboardist). For I and R , we use a synonym lists to cope with the use of multiple terms for the same concept (e.g. percussion and drums). We further count on how many of the web pages each rule applies for each M and I (or R).

These counts are document frequencies (DF) since they indicate, for example, that on 24 web pages Ralf Scheepers is said to be the singer of the band Primal Fear according to rule 6 (on 6 pages according to rule 3, and so on). To reduce uncertain information, we filter out those rules whose DF is below a threshold expressed as a fraction of the DF of

the highest scored rule (according to the DF score of all applying rules for the band under consideration).

Finally, for every instrument, the rule with the highest DF is selected and the respective (member, instrument)-pair is predicted. [SWPS07]

2.2.2 Automatic Question Answering Systems

An Automatic Question Answering System is a platform where a user can input a certain question in natural language and receive the supposed answer.

The Question Answering (QA) task has received a great deal of attention from the Computational Linguistics research community in the last few years. The definition of the task, however, is generally restricted to answering factoid questions: questions for which a complete answer can be given in 50 bytes or less, which is roughly a few words. Even with this limitation in place, factoid question answering is by no means an easy task. The challenges posed by answering factoid questions have been addressed using a large variety of techniques, such as question parsing, question-type determination, Word-Net exploitation, Web exploitation, noisy-channel transformations, semantic analysis, and inferencing. [SB04]

This kind of systems are related to this thesis project because, obviously, being able to understand synonyms allows a Question Answering System to understand and relate a biggest range of information to the entity in analysis.

For instance, if in a text the same public personality is addressed by his name or by his position on a certain institution. We can have access to more information if we take both into consideration. We are then able to understand the relation between names and positions/jobs.

Without noticing we are solving part of one of the mechanisms for lexical cohesion in the Portuguese language. The Anaphora that is related with the use of biographical synonyms.

This procedure is manifested in the substitution of one word (or longer term) for a term which, in the text, is equivalent (which serves as a synonym). [APAL10]

“**José Sócrates** disse esta sexta-feira a estudantes de Língua Portuguesa, num liceu em Paris, França que sente orgulho em servir o País "em momento de dificuldade". 'Nunca senti que o povo português me tivesse virado as costas', declarou ainda o **primeiro-ministro português**, que recordou as duas vitórias eleitorais que o levaram à chefia do Governo.”

The anaphora is thus a recovery - total or partial - of the referent of words earlier in the text. In this case, the interpretation of the anaphoric term (the one which refers to a previous) depends on the one of its antecedent. [APAL10]

In the above text, *primeiro-ministro português* is the anaphora of *José Sócrates*. And we only identify the referent of *primeiro-ministro português* (the reality represented by this term) if we take into account the meaning of it, which is the same meaning of its antecedent, *José Sócrates*.

Most of the effort to develop QA systems have to do with making them capable of responding questions about people, they need to have the capacity to collect data about people. Therefore, the important and related part of QA systems is precisely the fact that they answer questions about persons. The automatic generation of public personality's biographies can be seen as a problem of successive QA:

- Who is X?
- What is X profession?
- What position did X have last year?
- Etc...

Some of the most relevant projects in the area of Question Answering for the Portuguese language are projects like the XisQuê project and the RAPOSA project which will be described after doing a brief overview over an international Question Answering system as an example, Wolfram|Alfa.

2.2.2.1 Wolfram|Alfa

Wolfram|Alpha is one the more ambitious and well known Question Answering Systems. It's long-term goal is to make all systematic knowledge immediately computable and accessible.

The main objectives of Wolfram|Alpha project is to automatically obtain all objective data; implement every known model, method, and algorithm; and make it possible to compute whatever can be computed about anything. [Wol] The finality of Wolfram|Alpha project is to provide a single source that can be used by everyone for obtaining answers to factual queries. [Wol]

Basically Wolfram|Alpha aims to provide expert-level knowledge and capabilities to the broadest possible range of people. And the expertise areas span through all professions and education levels. [Wol]

It is still an ongoing project, but clearly entity relation extraction must be one of the features implemented by Wolfram|Alpha if the system is to ever be able to cover the broader number of person information and relations between themselves available.

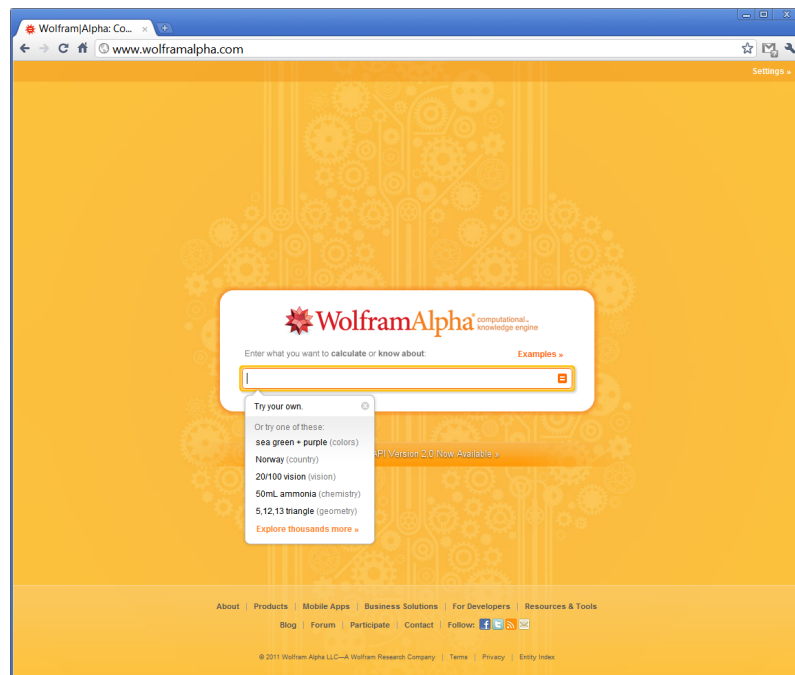


Figure 2.3: WolframAlpha's home page

2.2.2.2 XisQuê

XisQuê project is supported by a QA system developed to comply with the following major design features.

The admissible inputs are well-formed questions from Portuguese (e.g. *Quem assassinou John Kennedy?*).

The XisQuê system provides a real-time output.

The answers are searched in documents retrieved on the fly from the Web. The documents are obtained in the Portuguese web, that is the collection of documents written in Portuguese and available on-line.

The questions may address issues from any subject domain.

The answers returned are excerpts of the retrieved documents without additional processing, that is, the answer is not in natural language; the outputs are excerpts where the answer might be.

At the system's heart lies the QA infrastructure, which is responsible for handling the basic non-linguistic functionality. Its architecture follows what has become a quite standard configuration that has been explored and perfected in similar QA systems for other natural languages:

- **Question Processing.**

This phase involves three tasks:

- Detection of the expected semantic type of the admissible answers;
- Gathering of relevant keywords;
- Extraction of the main verb and major supporting NP of the input question.

- **Document Retrieval.**

In this phase, the system acts as a client of search engines (viz. Ask, Google, MSN Live and Yahoo!), submitting the list of keywords obtained in previous phase and retrieving relevant documents.

- **Answer Extraction.**

The last phase includes two tasks performed over the retrieved documents:

- The sentences most likely containing an admissible answer are selected;
- Candidate answers are extracted from the selected sentences. XisQuê delivers up to 5 candidate answers (termed “short answers”) together with the sentences from which they were extracted (“long answers”).

It may happen that for some answers only “long answers” are provided. [BRSS08]

On top of this infrastructure, the natural language driven modules were implemented by using state-of-the-art shallow processing tools developed at the NLX-Group [nlx]. They include tools for sentence and token segmentation, POS annotation, morphological analysis, lemmatization and named entity recognition, specifically designed to cope with the Portuguese language. [BRSS08] I will describe these tools on the Text Parser section.

2.2.2.3 RAPOSA prototype

RAPOSA (FOX) is a prototype question answering (QA) system for Portuguese that is being developed in Faculdade de Engenharia da Universidade do Porto as a subsidiary of a larger project that aims at developing wide-scope semantic analysis tools for Portuguese. The RAPOSA project was started mainly because the question answering task provides a good application scenario for validating the capabilities of our semantic analysis tools, and also for guiding its future developments.

The version of RAPOSA that participated in the QA@CLEF06 track is still very simple and is able to address only a very limited type of questions, mainly simple factoid questions that involve people, locations, dates and quantities.

Contrary to other question answering systems for Portuguese that make use of extensive linguistic resources or deep parsing techniques, RAPOSA uses shallow parsing techniques and relies on the semantic annotation produced by our named entity recognition (NER) system SIEMÊS (which I will describe in the text parser section), one of the key components of the suite of analysis tools being currently developed. As we will

State of the Art

The screenshot shows the XisQuê website interface. At the top, the logo 'XisQuê' is in orange, followed by the text 'Resposta a perguntas na Web portuguesa'. To the right, there is a search bar with the placeholder text 'Introduza uma pergunta:'. Below the search bar are two buttons: 'Perguntar' and 'Limpar'. The main content area displays a question: 'Pergunta: Quando morreu Elvis Presley?'. Below the question are five answers, each starting with 'Resposta #1:' through 'Resposta #5:'. Each answer includes a short text snippet and a link labeled 'documento fonte'.

XisQuê Introduza uma pergunta:

Resposta a perguntas na Web portuguesa

Pergunta: Quando morreu Elvis Presley?

Resposta #1: 50
Elvis Aaron Presley, ou simplesmente Elvis Presley (East Tupelo, Mississippi, 8 de Janeiro de 1935 — Memphis, Tennessee, 16 de Agosto de 1977) foi um famoso cantor, músico, ator e dançarino denominado o Rei do Rock, também conhecido pela alcunha de Elvis The Pelvis, apelido pelo qual ficou conhecido na década de 50 por sua maneira extravagante e ousada de dançar.
[documento fonte](#)

Resposta #2:
Elvis Presley: Elvis não morreu? Elvis Presley
[documento fonte](#)

Resposta #3:
Nesta quinta-feira (16), completam-se 30 anos que o cantor e compositor norte-americano Elvis Presley morreu de insuficiência cardíaca em sua mansão, Graceland, na cidade de Memphis, Estados Unidos.
[documento fonte](#)

Resposta #4:
Hoje, há 30 anos, morreu Elvis Presley, o Rei do Rock.
[documento fonte](#)

Resposta #5:
Elvis Christmas photos - <http://www.last.fm/music/Elvis+Presley/+images/55579793> / <http://www.last.fm/music/Elvis+Presley/+images/55579743> / <http://www.last.fm/music/Elvis+Presley/+images/55579757> / <http://www.last.fm/music/Elvis+Presley/+images/55579753> And simply one good photo - <http://www.last.fm/music/Elvis+Presley/+images/55580287>
[documento fonte](#)

© Todos os direitos reservados.

Figure 2.4: Example of XisQuê utilization

explain more thoroughly in the next section, SIEMÊS is used to tag a list of text snippets where candidate answers is believed to be found, extracted from the answer collection. For the type of questions being currently addressed, RAPOSA assumes that the correct answer is one of the entities tagged by SIEMÊS, and its job is thus to select the right one(s). [Sar06]

SIEMÊS is also used during the question parsing stage in order to identify relevant entities that may exist in the question.

As a result of its dependency to SIEMÊS, RAPOSA is currently limited to answer questions that involve factoids and simple definitions, although the number of different entities that SIEMÊS is able to tag is relatively high (more than 100). Therefore, RAPOSA is not able to answer list questions and definition questions that involve an explanation sentence. SIEMÊS does not yet resolve co-reference and anaphoric references, which is also a severe limitation when addressing questions whose answer cannot be explicitly found in the scope of one sentence.

The architecture of RAPOSA follows the standard approach, and consists of a pipeline of four blocks, each one responsible for a different stage in the question answering problem:

- **Question Parser.**

- Receives a question in raw text;
- Identifies the type of question, its arguments, possible restrictions and other relevant keywords, and transforms it to a canonical form. The admissible types of answer for the question are also identified at this stage.

- **Snippet Extractor.**

This block is responsible for retrieving snippets of texts (currently sentences) from the answer collection, using the information present in the canonical form of the question produced, by the previous block. The Snippet Extractor may return several text snippets.

- **Candidate Generator.**

After tagging the previously found snippets with SIEMÊS, this block tries to find candidate answer. Candidate answers are restricted to the set of admissible types found by the Question Parser. Several candidate answers may be found.

- **Answer Selector.**

This block selects one answer from the list of candidates found by the previous block. At the moment selection is being made based on redundancy, i.e., on the number of supporting snippets for each candidate.

Note that SIEMÊS is used in two of these blocks, namely in the Question Parser and in the Candidate Generator block. All the four blocks are still in a very preliminary stage of development. [Sar06]

2.2.3 Expert Finding Systems

The discovery of individual experts in a certain area, knowledge bases created by them or communities of expertise in a rapid way is an essential element of organizational effectiveness. [May06]

Expert Finding Systems allow this process of discovery to be easily accomplished. These systems need to map entity relations between persons (experts) and their areas of knowledge as well as their contact information. For this to be accomplished entity relation extraction processes are needed.

Typically the enterprises or organizations in need to find possible employees with expertise in a certain area, have little or no means of finding them other than by recommendations from other persons, entities or organizations. Also, busy experts normally do not have time to maintain adequate descriptions of their continuously changing specialized skills and areas of expertise.

Past experiences with this kind of "skills" databases indicates that they are quickly outdated and difficult to maintain,. [May06] What is required is an ability to support the following functions:

- *Identify* - Expert's name and/or large collections of properties / parameters (email, documents, briefings among others);
- *Classify* - Assess the type and level of expertise of individuals using multiple sources of information;
- *Rank* - Produce a rank order of experts on particular areas;
- *Recommend* - Given a the particular needs of the searcher, return an ordered list of experts in accordance to their rank level.

This kind of systems need to use entity relation extraction techniques to obtain the list of persons associated to their expertise areas in an automated fashion.

In this section we will describe the most relevant expert finding systems that use this kind of automated way of acquiring the expert's information and associations to their expertise areas.

2.2.3.1 MITRE's Expert Finder

MITRE's Expert Finder uses information mining techniques and activities on MITRE's corporate Intranet related to experts to provide all information acquired on each expert in an intuitive fashion to end users. An illustration of the initially created prototype in action is available in Figure 2.5. A user is trying to find data mining experts in The MITRE Corporation in this example. Searching for the expression "data mining", the system ranks all employees by the number of mentions about it and its statistical association with the employee name either in corporate communications (newsletters) or based on what they have published in their resume or document folder (a shared, indexed information space). [May06]

In MITRE's corporate employee database, the mention's frequency of each employee are used to rank them, pointing to sources in which they appear.

In Figure 2.5 we can see the first deployed MITRE prototype.

In Figure 2.5 we can see the simple user search using a simple keyword interface. In the example, a user searches for "expert finding" and is returned the top ranked experts in accordance to parameters like public documents, contributions, project time charges and so on. These parameters are shown below each expert profile. [May06]

State of the Art



Figure 2.5: MITRE's Prototype Expert Finder "Data Mining" Example [May06]

2.2.3.2 TACIT ActiveNet

TACIT ActiveNet's primary strength is the automated processing of expert's email and other elements they produce such as documents or briefings as well as descriptive material about themselves such as resumes or web pages. To complement this main feature TACIT ActiveNet also enables experts to self declare expertise.

The level of expertise of a particular individual on a particular topic is automatically ranked in TACIT. This is done by extracting frequencies of noun phrases from unstructured text as well as the date/time of their appearance. Thus while not classifying extracted phrases into semantic entities such as people, organizations, or locations, TACIT extracts and synthesizes linguistic units into "topics". For example, TACIT automatically detects the difference between "clinical trials" and "criminal trials" by observing that these two phrases belong to different uses. [May06]

We can see on Figure 2.6 the TACIT interface showing an expert public profile.

User profiles are automatically created. Users profiles have both a public section as well as a private section. This private profile will match queries of expert seekers but not reveal its private information to seekers unless users release it following an alert of the interest. The languages supported for source tracking by TACIT are English, French and German. [May06]

TACIT users can search for phrases, but the system does not perform full natural language processing of the query. TACIT supports keyword complemented with boolean operators (AND, OR, NOT) in its queries. [May06]

State of the Art

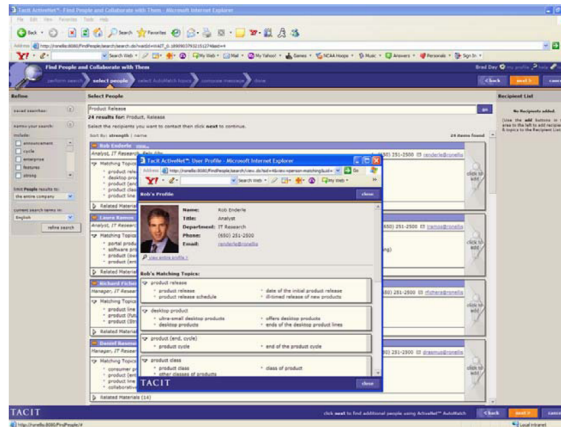


Figure 2.6: TACIT Active Net [May06]

2.2.3.3 AskMe

AskMe project unfortunately was already shut-down. It used to automatically processes a range of sources including documents, email, and external publications. Expertise keywords could also be manually added in the user profiles by each user.

One of the main AskMe features included the distinction between Auto profiling and Dynamic Profiling. Auto profiling is based on the mining of keywords either directly through AskMe or through a third party repository (e.g. email or document repository) to automatically identify expertise. Dynamic Profiling is the identification of expertise based upon the solutions end users provide and documents the authors provide either directly into AskMe or in a third party repository. As an example consider when an author publishes a document on a particular subject, when other users perform queries that match those documents, AskMe will identify not only the content of the document as a source of information but the author's profile will appear as a potential knowledge provider on the subject as well.

Expertise validation in AskMe was based on the type of information sources and expert qualifications/certifications that were present in the submitted document's metadata. AskMe also had rich Question Answering features. AskMe supported English and all major European languages but currently does not support characters for Chinese or Japanese. [May06]

AskMe's web based interface allowed users to search using natural language, Boolean, or fields based search like drop down boxes containing different ontologies. The search could be done using many parameters to define the end user's special needs through categories like skills, experience, project history, certifications, an so on. A list of experts or related documents was returned when the search was finally processed. [May06]

2.2.4 People Search Systems

The amount of information we have access these days has radically increased due to the internet. This increase of available information has led to a renewed interest in a broad range of Information Retrieval related areas that go beyond standard document retrieval.

Entity retrieval has gained some of this new attention. Entities are not represented directly (as retrievable units such as documents), and we need to identify them "indirectly" through occurrences in documents. [Bal07] In this section we focus on one particular type of entity: people.

Many commercial systems recognized this need for people search. These systems offer ways for finding individuals or properties them. Some of these searches may include locating classmates and old friends, finding partners for date and romance, white and yellow pages, etc.

As Krisztian Balog proposes, there are two information access tasks for an enterprise / organization setting: people finding, which is concerned with the retrieval of individuals that meet some criteria, and people profiling, which is about characterizing a specific person. Both tasks are explored along two main axes: topical and social. [Bal07]

In an enterprise setting, a key criterion by which people are selected and characterized is their level of expertise with respect to some topic. These kind of systems are better described in section 2.2.3.

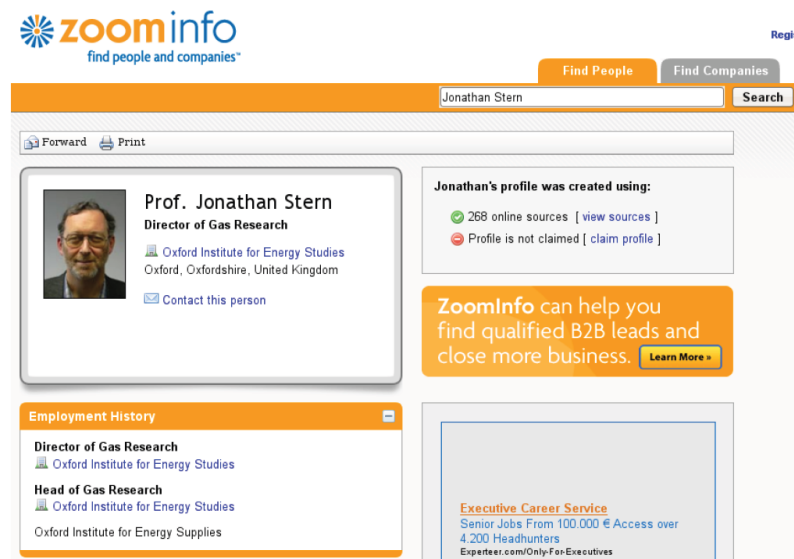
People Search Systems need to map persons to their possible names, role, professions, contact information and other artefacts among many other possible relations. For this to be accomplished entity relation extraction processes are needed.

In this section we will describe some People Search Systems that use this kind of automated way of finding relevant information about people.

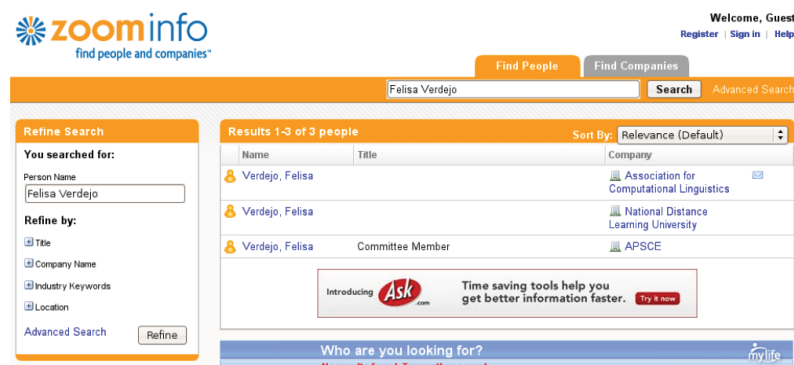
2.2.4.1 ZoomInfo

ZoomInfo was launched in 2005 as a Web People Search service. This service focuses on business related people. Submitting a query to Zoominfo results in a list of people profiles, each one containing information extracted from various web pages in which the person is mentioned as can be seen in Figure 2.7. [P⁺09]

ZoomInfo management does not publish details on the methods they use to solve ambiguity. Whichever method is used, it is relatively easy to find errors in the documents grouping and information extraction results. Searching for the name of "Felisa Verdejo" as an example test seen in Figure 2.8 from A. Picón's PhD Thesis, "*Web people search*" [P⁺09] the result obtained are three profiles that actually belong to the same person. Each profile relate the person to a different organisation. The first is related to the UNED University, another to the Asia-Pacific Society for Computers in Education and, finally, a

Figure 2.7: ZoomInfo person's profile example [P⁺09]

third one to the Association for Computational Linguistics. The search service was not able to link all this information to the same person in this particular test case. [P⁺09]

Figure 2.8: ZoomInfo's people search example [P⁺09]

2.2.4.2 Spock

Spock, a recent company, launched in 2007 a new people search service. This service first extracts people information from general pages on the Web which are then combined with information from structured sources like Wikipedia, IMDB, ESPN, LinkedIN, Hi5, Myspaces, Friendster, Facebook, Youtube, Flickr, etc. It is not difficult to find, as seen in the ZoomInfo project, examples of wrong document groupings in Spock. We can see an example test search for information about professor *Dekang Lin* spread across four different profiles in Figure 2.9 from A. Picón's PhD Thesis, "Web people search" [P⁺09].

Spock offered a 50000 dollars prize, one year after starting their people search service, to a team that could automatically solve the ambiguity of people names with the highest accuracy. The prize was awarded to team of six researchers from Germany’s Bauhaus University Weimar. [P⁺09]

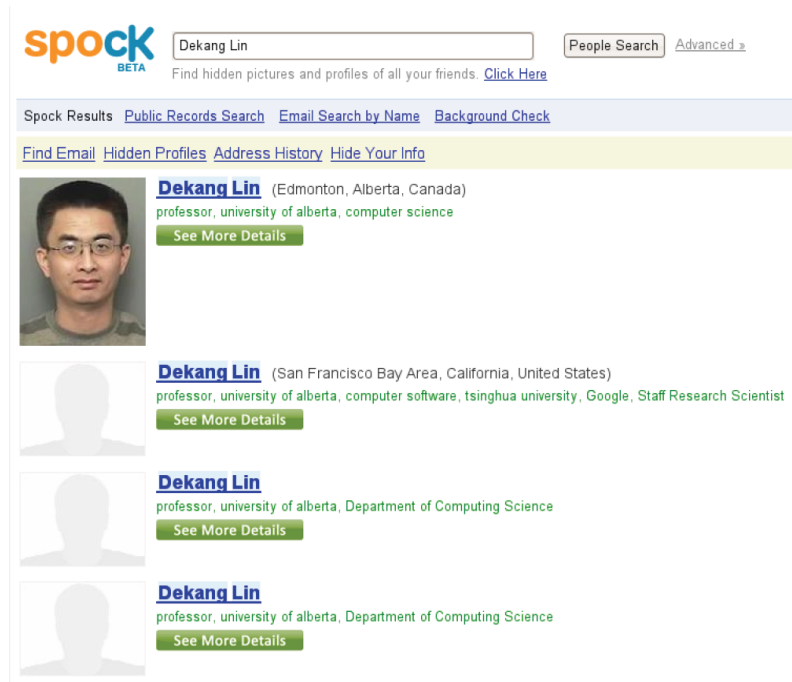


Figure 2.9: Spock’s people search example [P⁺09]

2.3 Similar Projects

Many projects are similar to this thesis project, directly or indirectly. In this section we will describe some systems that have, in some way, goals directly related to this thesis prototype main goals.

2.3.1 Freebase

Freebase project is actually the main inspiration for the development of this thesis project.

Mainly Freebase is a scalable, graph-shaped database of structured general human knowledge. It is based on a large community of contributors that insert this human knowledge in the form of know facts and entity artefacts. Freebase provides a graphical user interface that allows public read and write access. Researchers in the area can through an HTTP-based graph-query API obtain all information available on Freebase. [BCT07]

Freebase has information of approximately 20 million Topics or Entities. In Freebase, most of the available topics are associated with one or more types of entities such as

State of the Art

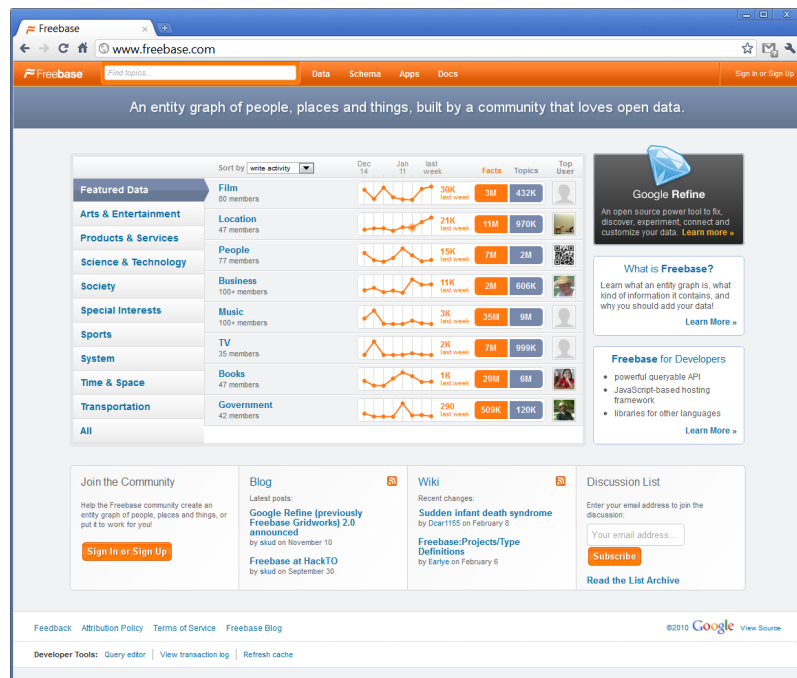


Figure 2.10: Freebase's home page

people, places, books, films, etc. Additional properties like "date of birth" for a person or latitude and longitude for a location can be also added. [fre] All entities can be related to other entities in many ways.

Freebase's data is the result of the contributions of its end users.

Freebase's API allows queries to be performed against its database data. This API also allows the end user to contribute with information to Freebase. There are Libraries available for many languages. [fre]

The biggest difference between this thesis project and Freebase is the fact that to accomplish the task of populating our ontology we do not have the means nor the community based effort Freebase has, so for our Portuguese language solution a semi-automatic way of acquiring data needs to be implemented. All information must be gathered in an automatic way and the validation of all information needs to be done by a human operators, thus the way is semi-automatic and not fully automatic.

2.3.2 KnowItAll

KnowItAll is a system that extracts facts, concepts, and relationships from the web and stores the acquired information in a structured way in an extensible ontology. [ECD⁺04]

KnowItAll creates text extraction rules for each class (entity) and relation in its ontology.

Each KnowItAll module runs as an independent thread. Communication between the implemented modules is accomplished through asynchronous messaging processes. [ECD⁺04]

KnowItAll's main modules are: [ECD⁺04]

- **Extractor** - a set of extraction rules for each entity and relation are instantiated;
- **Search Engine Interface** - queries based on its extraction rules are automatically formulated. Each query is composed of the keywords in the rule. KnowItAll makes use of up to 12 search engines including Google, Alta Vista, Fast between others to perform the formulated queries in order to extract relevant information;
- **Assessor** - search engine results computed statistics for each query are used to assess the veracity of the *Extractor*'s conjectures;
- **Database** - KnowItAll stores all the extracted information in a structured database that replicates the ontology decisions.

KnowItAll's first task is to begin a bootstrap learning phase, where a set of extraction rules for each entity or relation are instantiated. After, the KnowItAll's Extractor begins finding instances from the web that match the predefined set of rules. Finally, the KnowItAll's Assessor assigns the computed probability to each instance. [ECD⁺04]

KnowItAll' sequence of events is very similar to the one this thesis project needs to implement, although the entities and relations that we are going to consider are more specific than the ones considered in KnowItAll. The bootstrapping philosophy of KnowItAll's system is very close to the one this thesis project intends to use.

2.4 Summary

As we saw, the entity relation extraction is much larger subject than it seems at first sight. In this chapter we talked about many projects that directly or indirectly need entity relation extraction processes to function.

We talked about applications that were developed in the Automatic Biography Generators context, which is the closest to this thesis. We also talked about the Automatic Question Answer Systems, which also may allow to better test this project. An overview of Expert Finding Systems and People Search Systems is also done.

These kind of applications all play their part on bringing the natural language development a step forward. This step forward will allow us to one day make Natural Language Communication between machines and human beings a reality.

At this moment in time, much is in need to be done in this particular type of systems, principally concerning the Portuguese language.

Chapter 3

System Specification

This chapter contains the specification of the new public entity information extraction and ontology system, which fulfils the goals previously identified.

The first section contains an overview of the system, presenting a description of the desired behaviour and architecture. Secondly we present a description of the required modules as well as the main features that should be implemented with a brief description of each one. After that we perform an analysis of the use cases that need to be addressed. Finally, the global architecture of the public entity information extraction and ontology system is thoroughly described.

3.1 Overview of the New System

According to the aims of this project explained in sections [1.3](#) and [1.4](#), and the needs specified in the previous chapter, the main objective of this thesis project is to create a system that can automatically populate an ontology with biographical information of public entities.

The system should also allow human validation of information, thus becoming a semi-automatic system where human community efforts plays an important role so we have defined three distinct components:

- **Database** - where all entity information and its relations are stored;
- **Automatic Information Extraction System** - a text-mining system that must be able to populate our ontology and interact with the user validations to choose amongst different paths of entity discovery;
- **Graphical User Interface** - A complete interface where all the information contained in our ontology can be viewed, edited and validated.

3.1.1 Database

As mentioned before, the information gathered automatically by the Information Extraction System, or added manually through the Graphical User Interface to the system, needs to be mapped and stored in a database. But even before approaching that problem, first we need to define what kind of information is going to be stored in it.

The main public entities that we want to obtain and store in our system are Persons and Organizations:

Person

For each Person, the database must contain:

- **Name** - the main name of the person in question;
- **Nicknames** - other possible names for the same person;
- **Roles/Ergonyms** - each entry of this type makes the connection between the person and the organization the role belongs to, so it is important to keep a time reference to allow the same position to be mapped to different persons in distinct time spans;
- **Record of Relations** - between this particular entity and others regarding a particular position the entity occupied;
- **Photo** - this photo must be obtained from a reliable source in order to be sure of its veracity;
- **Sources** - all sources where the information was gathered must be kept;
- **Other** - all other fields that might be considered important during implementation.

Organization

For each Organization, the database must contain:

- **Name** - the main name of the organization in question;
- **Other Names** - other possible names for the same organization;
- **Jobs/Positions** - each entry of this type makes the connection between the organization and the person the position belongs to, so it is important to keep a time reference to allow the same role to be mapped to different persons in distinct time spans;
- **Record of Relations** - between this particular organization and others regarding a particular time span;

System Specification

- **Logo** - this logo must be obtained from a reliable source in order to be sure of its veracity;
- **Sources** - all sources where the information was gathered must be kept;
- **Other** - all other fields that might be considered important during implementation.

Role

For each Role, the database must contain:

- **Name** - the main name of the role in question;
- **Other Names** - other possible names for the same role;
- **Person** - each entry of this type makes the connection between a person and the role;
- **Organization** - each entry of this type makes the connection between an organization and the role;
- **Sources** - all sources where the information was gathered must be kept;
- **Other** - all other fields that might be considered important during implementation.

Since all information must be validated by an human actor and to be sure the validations are correctly made, we must implement a user authentication system. For that all user information must also be stored in our DB.

User

For each User, the database must contain:

- **Name** - the main name of the user in question;
- **User Name** - name utilized to perform login;
- **Password** - password associated with user name;
- **Active** - flag that lets the system know if the user is or is not allowed to perform login;
- **Other** - all other fields that might be considered important during implementation.

Since all entries in the DB will have a certainty level associated, we also need to store it in the DB. Each certainty entry also must store information about the element from which it was discovered in order to allow us to draw the discovery history of the Automatic Information Extraction System and to propagate certainty through discovered elements.

The certainty entries and the propagation process will be further explained in section [3.1.2.3](#).

Certainty

For each Certainty entry, the database must contain:

- **Identification** - identification number of the element to which the certainty entry is associated;
- **Type** - Flag that represents the type of element to which the certainty entry is associated;
- **From Who Identification** - identification number of the element from which the element in question was discovered;
- **From Who Type** - Flag that represents the type of element from which the element in question was discovered;
- **Number of Hits** - number of times the element appeared in the search process;
- **Certainty** - certainty percentage that the Automatic Information Extraction System has in relation to the element found, according to the method certainty and the number of hits ;
- **Other** - all other fields that might be considered important during implementation.

Now that the information needed is defined we determine how to obtain all this information in a reliable way in order to assure its veracity.

3.1.2 Automatic Information Extraction System

To obtain the information referred in chapter [3.1.1](#), an application based on a scripting language is used to crawl the WWW or a local DB.

To acquire all this data with a certain degree of veracity and discernment we need to mind the sources of this information. To better know which sources and resources are available and how to use them, see chapter [3.1.2.1](#). This process should also be automated as in a loop, in a way that even with a certain limited initial information the maximum

information can be acquired. The level of certainty of any information that is acquired automatically must be calculated and stored. All information gathered automatically from a previously found entry will have its certainty level influenced by the certainty level of the first. This topic is further explained in chapter 3.1.2.3.

To better explain the sequence of events in this Automatic Information Extraction System (AIES) the diagram 3.1 tries to illustrate a simple iteration.

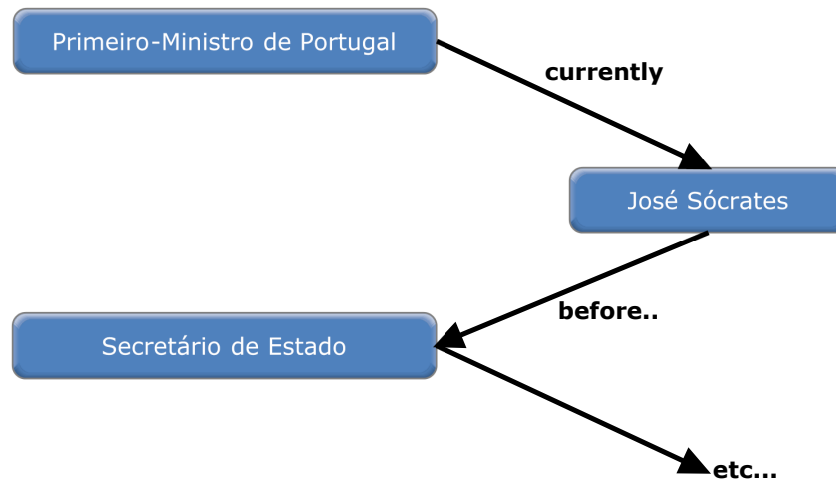


Figure 3.1: Diagram that illustrates the sequence of search events

The techniques that are used to obtain this information from each source of information are described in 3.1.2.2.

3.1.2.1 Sources of online information

Choosing the most adequate sources of information is crucial to obtain data with some credibility. Some of the most credible sources of information supply an application programming interface that allows the crawler to obtain structured results. In this section we describe some of these sources and the techniques that allow us to obtain the structured data.

Wikipedia/DBpedia

To obtain specific information out of Wikipedia, we can use DBpedia.

DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. It allows information retrieval applications to make sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data. [dbp] Through DBpedia we can get specific parts of articles, like abstracts and titles, amongst others. This is extremely useful to enhance entity profiles.

Sapo

The Sapo Search Engine allow us to obtain JavaScript Object Notation Files (JSON) with the search results in an organized fashion, making it easy to choose the hyperlink we want to pursue according to the search goals.

Google

To obtain search results from the Google search engine, we can use the HTTP GET method. This method requests a representation of the specified resource, and is typically used in this kind of crawler applications.

Yahoo

Yahoo provides an Application Programming Interface (API) to allow the retrieval of information, just like Sapo.

3.1.2.2 Query Techniques

The hardest part of this thesis project is to figure how to make productive and effective queries to search engines in order to obtain the needed information. The main goal is to be able to automatically follow a logical sequence of queries in order to build a path of answers and get pertinent information.

Person Information Extraction Query Techniques

Given a person's name N we could make these Portuguese queries to the search engines (the following are examples):

- $N + \text{"é o"}$
- $N + \text{"fez"}$
- $N + \text{"disse"}$
- $N + \text{"ocupa o cargo de"}$

With these queries we could obtain some links where the person's ergonyms, roles and consequent relations with Organizations could be discriminated. We can try to define the ergonym/role instead of the person's name, if we do not know any in order to obtain information about persons that already applied for that specific job/occupation.

Given a ergonym/role name R , we could make these Portuguese queries to the search engines (the following are mere examples):

- "o" + R + "disse"
- "o actual" + R + "é o"
- R + "participou"
- "com a presença do" + R

Another possibility is the union of the person's name and its ergonym/role, making it possible to estimate the normal position of both in a sentence and predict were to find either one if missing. For example:

- "o" + R + ", " + N + ", disse"
- N + ", no cargo de" + R + ", fez"

To obtain specific relation's information between a person and a certain entity we can simulate parts of sentences. These will eventually force the results to return the name of the entity that, in fact, may have a relation with the person.

For the person we already know we can consider: R and N for its role and name respectively and for the second entity, the entity we want to discover that has a relation with the first, we can consider $R2$ and $N2$. For example:

- "o" + R + "em colaboração com" + $R2$
- N + "e" + $N2$ + "participaram no"

Organization Information Extraction Query Techniques

Given an organization's name N we could make these Portuguese queries to search engines (the following are examples):

- N + "é uma organização que"
- N + "fez"
- N + "criou"
- N + "permitiu"

With these queries we could obtain some links where the roles that the organization has, and consequent relations with Persons that occupy them could be discriminated. We can try to define roles instead of the organization's name, in case we do not know any, to try to obtain information about organizations that have that specific role.

Given a role's name R we could make these Portuguese queries to search engines (the following are examples):

- "o" + R + "pertence à"
- "o actual" + R + "do"
- R + "é um cargo da"

Another possibility is the union of both the organization's name and one role that might belong to it, making it possible to suppose the normal position of both in a sentence and predict were to find either one if missing. For example:

- "o" + R + "do" + N + "disse"
- N + " tem como" + R + "o"

To obtain specific relation's information between an organization and a certain entity we can simulate parts of sentences that will eventually force the results to return the name of the entity that, in fact, may have a relation with the organization.

For the organization we already know we can consider: R and N for one of its roles and its name respectively and for the second entity, the entity we want to discover that has a relationship with the first, we can consider $N2$ for it's name, and try to find matches for the same role. For example:

- "o" + R + "da" + N + "actualmente é o" + $N2$
- N + "contratou" + $N2$ + "para desempenhar o papel de" + R

These are all examples; they may change considering the needs along the project development. As explained before, this is the difficult part of the project: finding patterns in the text, always minding the target texts and the desired type of information to be extracted.

3.1.2.3 Certainty of Acquired Information

Although all the data is already filtered during the acquisition process, the main obstacle in the veracity of the information is the WWW itself. In general, all information has its degree of imperfections, since it was acquired from human input. Sometimes only because of spelling mistakes. One must also have in mind problems like duplicated names.

All information extracted automatically has a level of certainty associated to it that needs to be stored in the project's DB. This certainty is calculated according to the certainty of the extraction method, previously calculated, and the actual number of times that that entry occurred on the WWW or local DB.

The main objective of keeping the certainty level of an entry is to allow the operator to have some indications that may help him to validate the information. It is also used to propagate certainty and to define the end of the search cycle. To propagate certainty, all information gathered automatically from a certain entry will have their certainty level influenced by the certainty level of it. When the certainty of a found element is below 50% (even if only because the propagation of the previous certainties over it), the search cycle stops pursuing this search tree branch and goes to another that can still be searched.

To better illustrate the certainty sequence of events, please analyse the certainty tree picture 3.2.

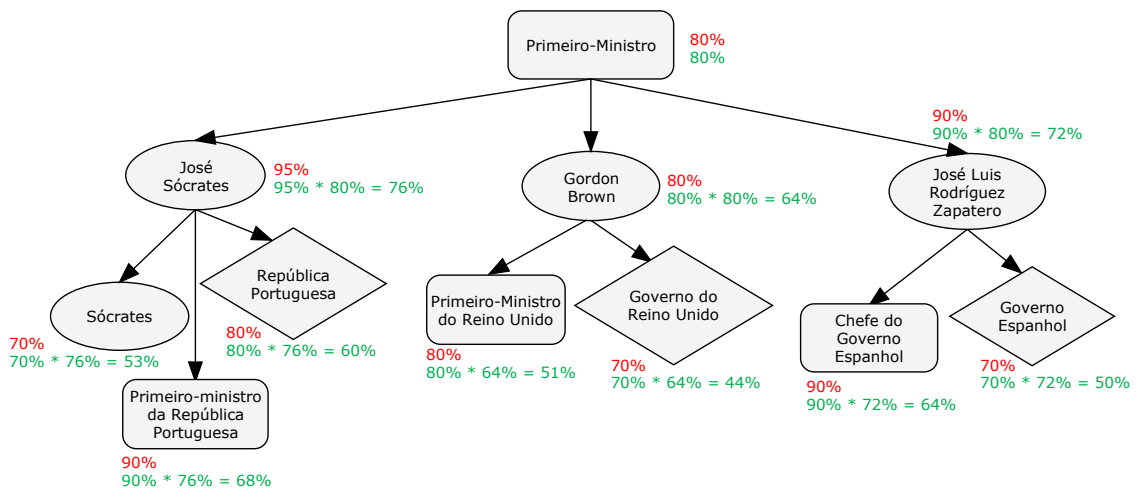


Figure 3.2: Certainty diagram illustrating sequence of discovered elements

In this figure we can see the certainty being updated. In red we have the *method's certainty*. This certainty is calculated taking in consideration the fixed method's certainty that was calculated by manually analysing a small sample of results and the number of times the element in question appeared in the search results. In green we can see the *final certainty* being calculated by the multiplication of the actual *method's certainty* and the *final certainty* of the element from which the actual element was discovered. In figure 3.2 all elements have their specific representation, *Persons* are represented by an oval icon, *Organizations* are represented by a diamond icon and finally *Roles* are represented by a rounded rectangle.

Despite all considerations, it is very difficult to avoid incorrect or imprecise information. To overcome this fact, an interface will be developed allowing an human actor to validate and correct all the entries. The Graphical User Interface specification is described in chapter 3.1.3. When the human actor actually validates an entry, its certainty will pass to 100% and consequently all entries found after the referred entry will see their certainty increased accordingly.

The figure 3.3 tries to describe the certainty update event.

System Specification

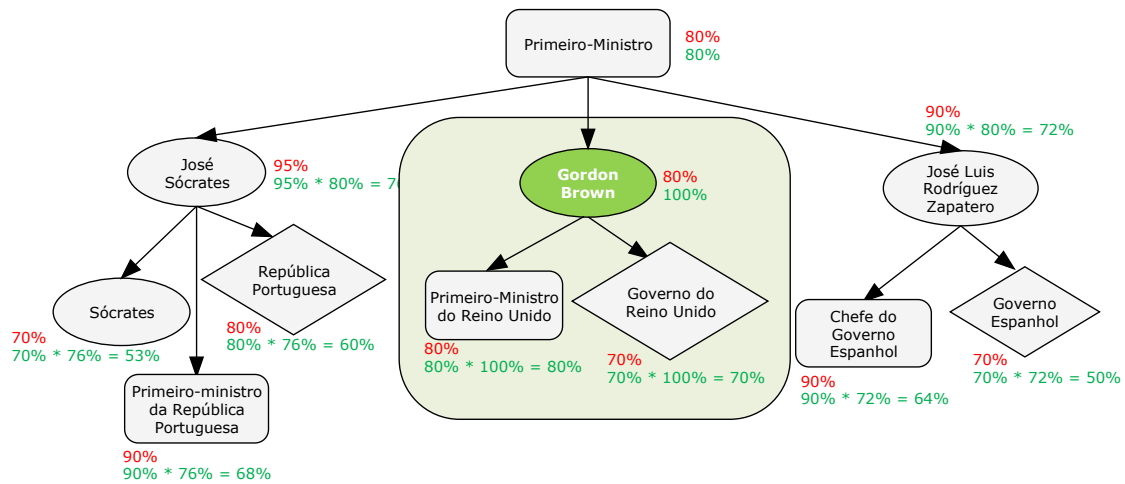


Figure 3.3: Certainty update event diagram

As can be seen, *Gordon Brown* was validated by a human operator. Its certainty is automatically updated to 100%. Consequently all elements that were discovered from *Gordon Brown* had their certainty updated accordingly.

3.1.3 Graphical User Interface

The GUI of this project is a web interface that allows the human actor to execute many actions described in Table 3.1.

To avoid mis-usage of this interface a login system needs to be implemented. As an extra an interactive graph showing all the connections between entities should also be developed to aid the visualization of the information on the database. The main final result is just the final database with all the biographical information of public entities to then be included in other projects like the Voxx project.

3.2 Use Cases

This subsection presents the main use cases of the new public entity information extraction and ontology system. As can be seen in figure 3.4, they were grouped in three main usage groups:

- Administration Access
- Operation Access
- Regular Access

Associated to each usage group there is a specific user type. Each of these user types operates inside its respective usage group. The user types are:

System Specification

Create	new User profile new Person new Organization new Role
Bind	existing Roles to Persons existing Roles to Organizations
Unbind	Person's Roles Organization's Roles
List	all Users registered in the system all Persons on the DB all Organizations on the DB all Roles on the DB
Visualize	a User's profile a Person's profile a Organization's profile a Role's profile all relations between Persons and Roles all relations between Organizations and Roles search certainty trees of each entry (Graph illustrating de discovery sequence) an interactive graph showing all the connections between entities
Edit	all User's information all Person's information all Organization's information all Role's information
Delete	a User a Person an Organization a Role
Activate	or deactivate a User to allow him or not to use the system
Validate	each atomic information of a Person each atomic information of a Organization each atomic information of a Role
Control	the information extraction system

Table 3.1: Graphical User Interface Specification

System Specification

- Administrator User Type
- Operator User Type
- Normal User Type
- Unauthorized User

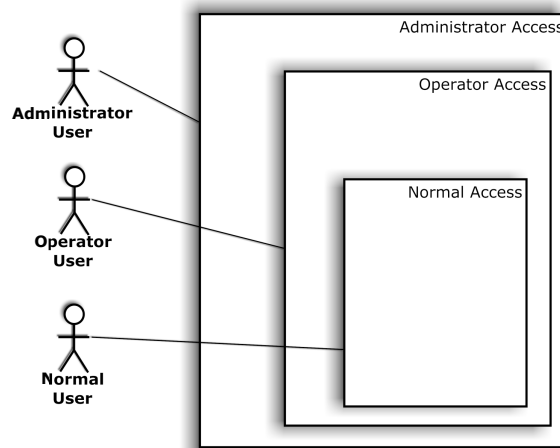


Figure 3.4: Usage Groups Diagram

Each usage group specifies the different ways the interface can be accessed. As can be seen, they are all contained inside each other. This means that the *Administrator User* can see and do everything an *Operator User* can and more, and likewise the *Operator User* can see and do everything a *Normal User* can do and more. The *Unauthorized User* can not access or do anything, only login to become one of the other users. The *Administrator User*, *Operator User* and *Normal User* all belong to the *Authorized User* class as seen in the User Inheritance Diagram 3.5.

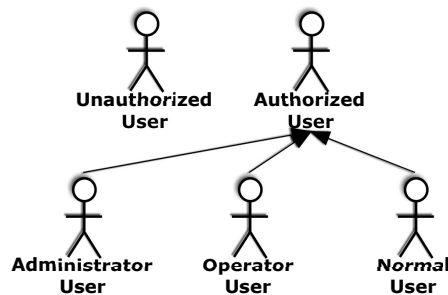


Figure 3.5: User Inheritance Diagram

3.2.1 Regular Access

In this section the main *Normal User* use cases are described. The GUI login system allows an *Unauthorised User* to become one of the *Authorized User* types referred in the User Inheritance Diagram 3.5. The login process is described in the Authorization Process Use Case Diagram 3.6.

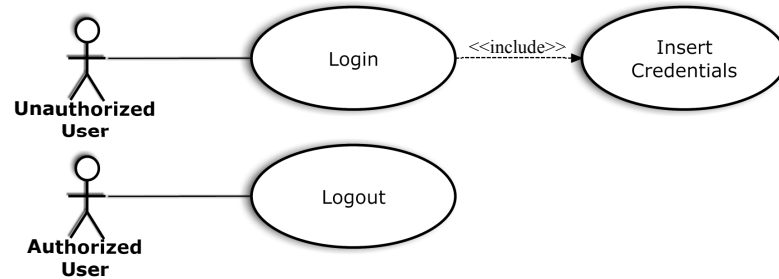


Figure 3.6: Authorization Process Use Case Diagram

When a user is logged in as a *Normal User*, he only gets viewing access permissions of all the system. The main viewing use cases for a *Normal User* are described in the diagram 3.7.

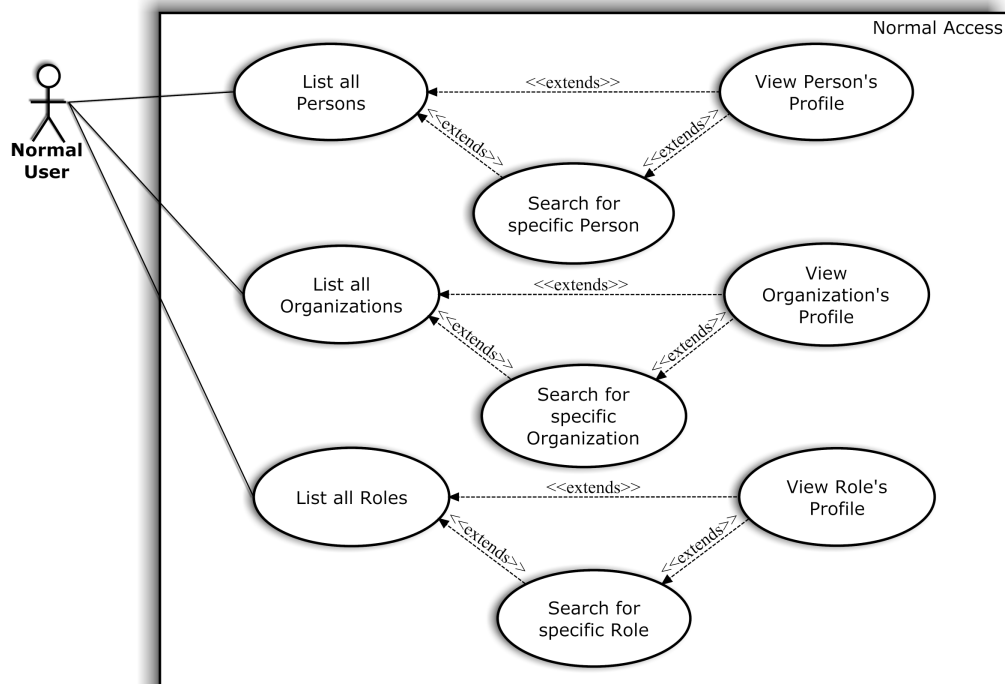


Figure 3.7: Normal User View Access Diagram

When a *Normal User* is already viewing a profile (*Person*, *Organization* or *Role*), he can view all information associated to it, including certainty and number of hits. Each field of information has a validation symbol associated to it. This symbol can be a red cross if the field in analysis has not yet been validated by a human operator. Accordingly, the symbol can be a green tick if the field in analysis was already validated by a human operator. As a viewing accessibility option, the user can also hide or show validation symbols to ease the viewing of the profile. When viewing a profile of any element in the ontology the user can also view the Information History Graph. This can be seen in the *Normal User* view of a *Person / Organization / Role* profile diagram 3.8. This graph contains the elements from which the element in analysis was discovered and the elements that were discovered from it. When seeing a *Person*'s profile or an *Organization*'s profile the user can also see a list of role's associated to them and click on them to see their profile and the *Organization/Person* associated respectively. This can be seen in diagram 3.8. When seeing a *Role*'s profile the user can click on the *Person* and *Organization* associated to it to go to their's profile. This can be seen in diagram 3.8.

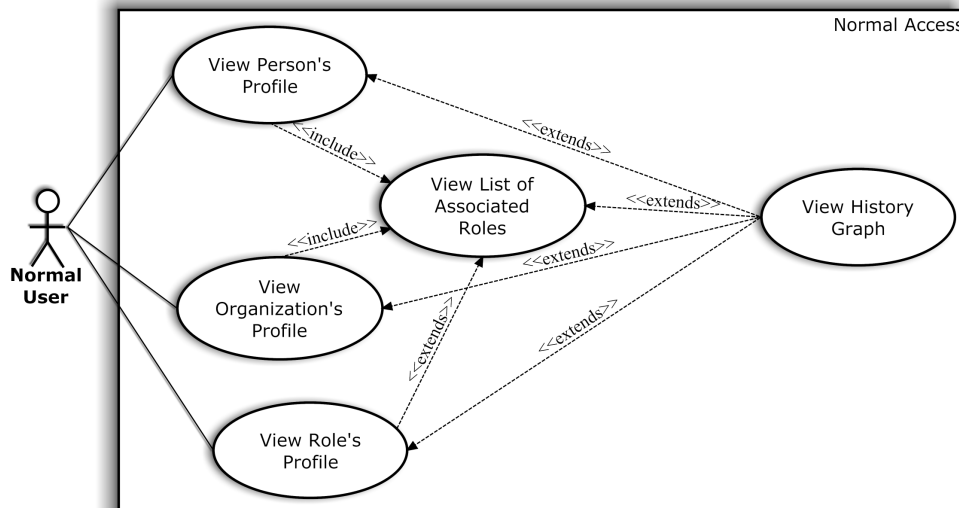


Figure 3.8: Normal User View of a Person / Organization / Role profile

Every *Authorized User* can also edit its own profile. In the profile editing page, the user may change his name and password. The user name can not be changed in order to avoid problems with duplicate user names. The profile editing can be seen in diagram 3.9

3.2.2 Operation Access

In this section the main *Operator User* use cases are described. The *Operator User* has all the *Normal user* viewing use cases and their respective editing and adding use cases.

System Specification

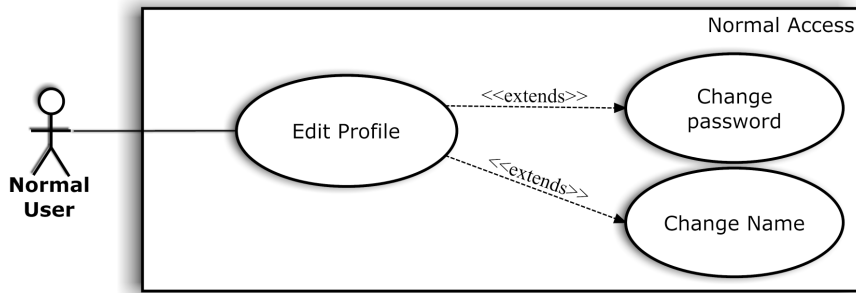


Figure 3.9: Normal User edit of own profile

When the *Operator User* views the list of *Persons*, *Organizations* and *Roles*, besides being able to search and see each profile, he can add new *Persons*, *Organizations* and *Roles* respectively. This can be seen on diagram 3.10.

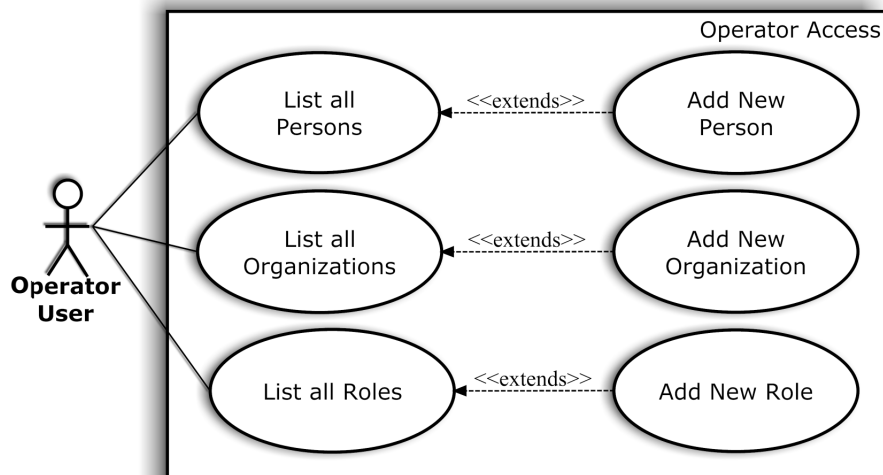


Figure 3.10: Operator User adding use cases

Viewing any profile (of *Persons*, *Organizations* and *Roles*), the *Operator User* can edit all information and validate it. This can be viewed in the diagram 3.11.

When viewing any profile the *Operator User* can also delete it. This can be seen in diagram 3.12. When deleting a *Person*, *Organization* or *Role*, all information related to it must be deleted, including the certainty entry on the DB, all nicknames and sources. When deleting a *Person* or a *Organization* all roles associated must be unbinded so that they become free for other *Persons* or *Organizations* to use. At *Person's* and *Organization's* profiles the user needs to also be able to bind or unbind any existing roles. New roles can also be created and automatically associated to the *Person / Organization* in question. Please analyse diagram 3.12 for the better understanding. When viewing a *Role's* profile

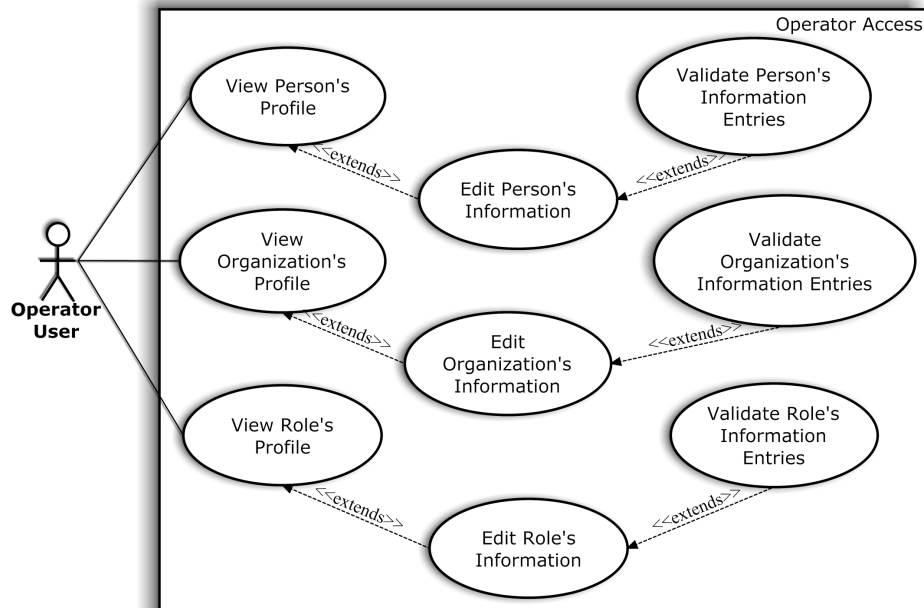


Figure 3.11: Operator User editing use cases

the *Operator User* may change the *Person* that is associated, simply by selecting another one from the list. The same kind of change can be done for the *Organization* associated to allow a *Role* to have the correct associations mapped to it. The *Operator User* must be also able to only unbind the *Person* or the *Organization* without having to associate the *Role* to a new one.

Another feature that needs to be implemented in each profile page is the ability to edit or add new nicknames / known names and edit or add related links or references. This feature allows the identification of the same *Person*, *Organization* and *Role* by their different names in many contexts. This feature use case can be seen in diagram 3.13.

3.2.3 Administration Access

In this section the main *Administrator User* use cases are described. The *Administrator User* has all the *Operator user* viewing and editing use cases adding complementary control over the information extraction system and user management. Regarding the user management, besides being able to do the use cases described in 3.9, the *Administrator User* is able to list all users or search for a specific one. These use cases are described in figure 3.14. The *Administrator User* has access to all *User* profiles with viewing and editing permissions. Besides the normal editing of user profile seen in 3.9, the *Administrator User* can activate or deactivate a user. A deactivated user can perform login but can not access any information on the system until his situation becomes normalized and he

System Specification

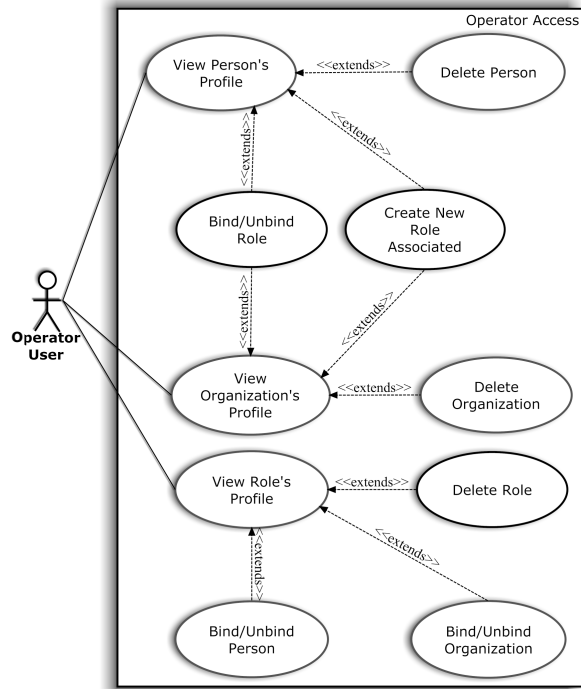


Figure 3.12: Operator User bind, unbind role and delete use cases

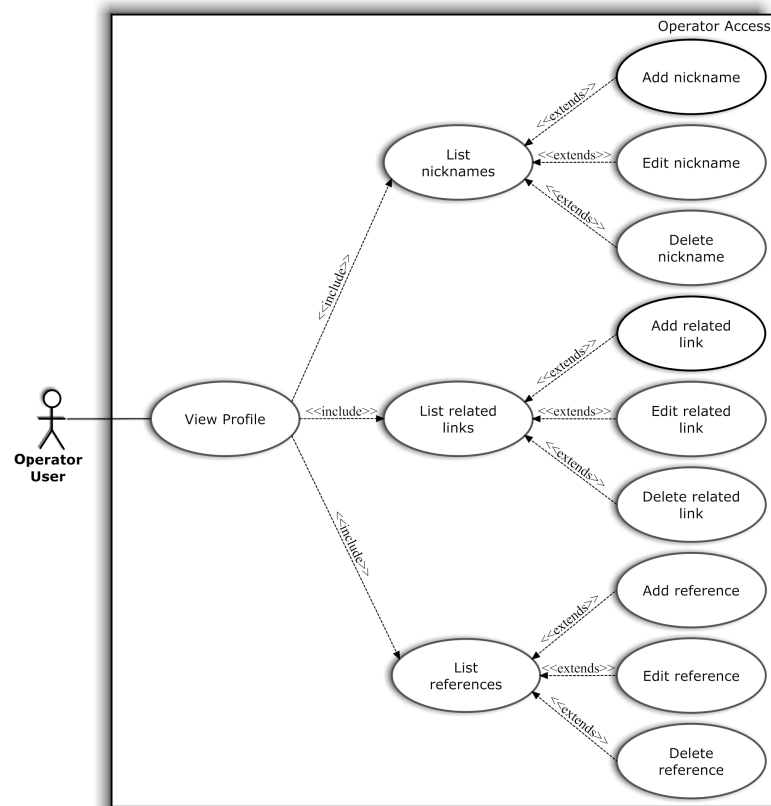


Figure 3.13: Editing nicknames, related links and references use cases

System Specification

is activated by a *Administrator User*. The *Administrator User* is also the only kind of user that can delete any other user. These use cases are illustrated in figure 3.15.

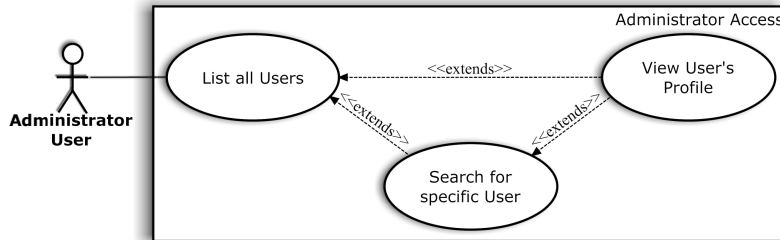


Figure 3.14: List and search all users use cases

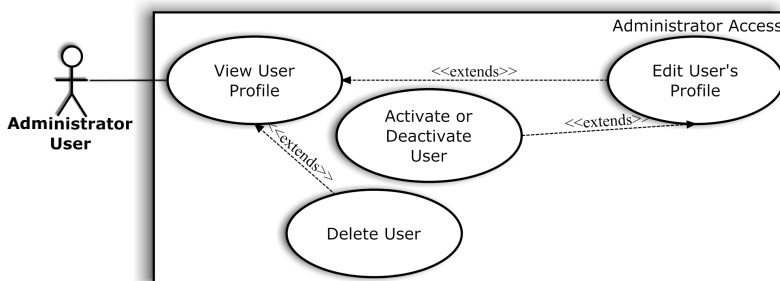


Figure 3.15: Administrator's user management use cases

Concerning the control over the information extraction system, the ideal interface would allow the *Administrator User* to define which course the public entity search system should take. A good way of doing that would be defining the name from which the search engine should start. Another interesting feature would be to allow the *Administrator User* to view in real time all logs of each search branch allowing him to know the course and problems of the entity search system. These use cases can be analysed in diagram 3.16.

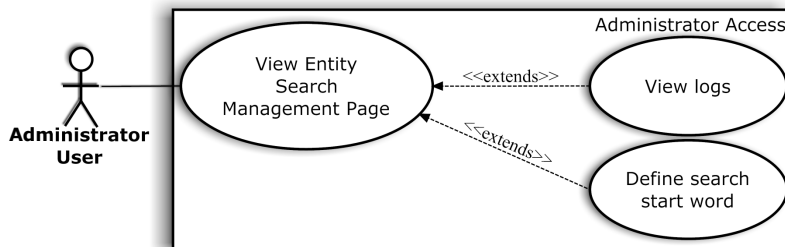


Figure 3.16: Administrator's search system management use cases

3.3 Main Desired Features

The identification of the required features was based on the overview of the specification of the new system made on section 3.1 and on the use cases presented in section 3.2. Each feature has an associated priority (high, medium or low) and an id.

Since the DB exists only to support the GUI and the AIES and has no special feature directly associated to it, in this section we will only speak of the features that need to be implemented in the GUI and the AIES.

3.3.1 Automatic Information Extraction System

As can be observed in table 3.2, the system should support the extraction of person's names with high priority, the extraction of organization's names with medium priority and the extraction of role's names with high priority.

ID	Feature	Priority
F1.1	Extraction of Person's Names	High
F1.2	Extraction of Role's Names	High
F1.3	Extraction of Organization's Names	Medium

Table 3.2: Information extraction feature list

To be able to see the connections between *Persons*, *Roles* and *Organizations*, the system needs to be able to map them together. The table 3.3 contains all the mapping types that should be implemented. Map *Persons* to *Roles* with high priority, map *Organizations* to *Roles* with medium priority, map *Organizations* to *Organizations* with low priority, map *Persons* to *Persons* with low priority and map *Roles* to *Roles* with low priority.

ID	Feature	Priority
F2.1	Map Persons to Roles	High
F2.2	Map Organizations to Roles	Medium
F2.3	Map Persons to Persons	Low
F2.4	Map Organizations to Organizations	Low
F2.5	Map Roles to Roles	Low

Table 3.3: Relation mapping feature list

To complement the acquired information the system should be able to extract other details about the entities it discovers. Additional information from Wikipedia or other sources should also be used to complete each entity profile.

As can be seen in table 3.4, the system should obtain *Person's* photos, nicknames, abstract and article link with medium priority. *Organization's* logo, known names, abstract and article link should also be obtained with medium priority. The same happens

for *Role*'s abstract, article link and time span. Last but not least, with low priority, the system should obtain *Organization*'s headquarters and type; identify *Person*'s sex and finally obtain other relevant information about all these entities.

ID	Feature	Priority
F3.1	Obtain Person's photo	Medium
F3.2	Obtain Person's nicknames	Medium
F3.3	Obtain Person's abstract	Medium
F3.4	Obtain Person's article link	Medium
F3.5	Obtain Organization's logo	Medium
F3.6	Obtain other Organization's known names	Medium
F3.7	Obtain Organization's abstract	Medium
F3.8	Obtain Organization's article link	Medium
F3.9	Obtain Role's abstract	Medium
F3.10	Obtain Role's article link	Medium
F3.11	Obtain Role's time span	Medium
F3.12	Obtain Organization's headquarters address	Low
F3.13	Identify Organization's type (Tag HAREM)	Low
F3.14	Identify sex of Person	Low
F3.15	Obtain other relevant informations about each Entity	Low

Table 3.4: Complementary information extraction feature list

3.3.2 Graphical User Interface

For the Graphical User Interface the main features to be implemented are described in this section.

As can be analysed in table 3.5, the system should allow the view of persons list with high priority, the view of organizations list with high priority and the view of roles list with high priority.

ID	Feature	Priority
F4.1	View List of Persons	High
F4.2	View List of Roles	High
F4.3	View List of Organizations	High

Table 3.5: View lists feature list

In table 3.6 we can see that the system should allow the view a person's profile with high priority, the view of an organization's profile with high priority and the view of a role's profile with high priority.

Since the system should allow editing tasks on the DB, the table 3.7 shows that the system must supply a way to edit persons with medium priority, edit organizations with medium priority and the edit roles list with medium priority.

ID	Feature	Priority
F5.1	View Person's Profiles	High
F5.2	View Role's Profiles	High
F5.3	View Organization's Profiles	High

Table 3.6: View specific profiles feature list

Knowing that the system must have an authentication system to prevent misuse of it, the table 3.8 shows us that a login system should be implemented with medium priority. Additionally, to complement the authorization and user system, a list of system users should be implemented with low priority, as well as a view of user profile and an edit profile feature with low priority.

Additional to all these features, some extra features should be considered, these features are not key features although of very interesting use and can be seen on table 3.9. The extra features are the visualization of the certainty graph of each entity with medium priority, the entity relation's graph with low priority and finally the Automatic Information Extraction System's page which has the main goal of controlling the referred system with medium priority.

3.4 Global Architecture

This section introduces the system's architecture by the means of diagrams and a small description of each part of the overall architecture. Considering all the proposed features, the overall architecture of the system is presented in figure 3.17.

The new system is composed by three main components:

- **GUI** - The Graphical User Interface component allows the clients to interact with all the rest of the system in an easy way. All information should be available from it to the user. All interactions user-system should be implemented in this component.
- **DB** - The Database component main goal is to support the ontology defined and the user authentication system. This kind of support involves storing all extracted information and also storing all user details.

ID	Feature	Priority
F6.1	Edit Persons	Medium
F6.2	Edit Roles	Medium
F6.3	Edit Organizations	Medium

Table 3.7: Edit specific entities feature list

System Specification

ID	Feature	Priority
F7.1	Login System	Medium
F7.2	View list of Users	Low
F7.3	View User's profile	Low
F7.4	Edit Users	Low

Table 3.8: Authentication system feature list

ID	Feature	Priority
F8.1	View entity certainty graph	Medium
F8.2	AIES control page	Medium
F8.3	View entity relations graph	Low

Table 3.9: Graph view feature list

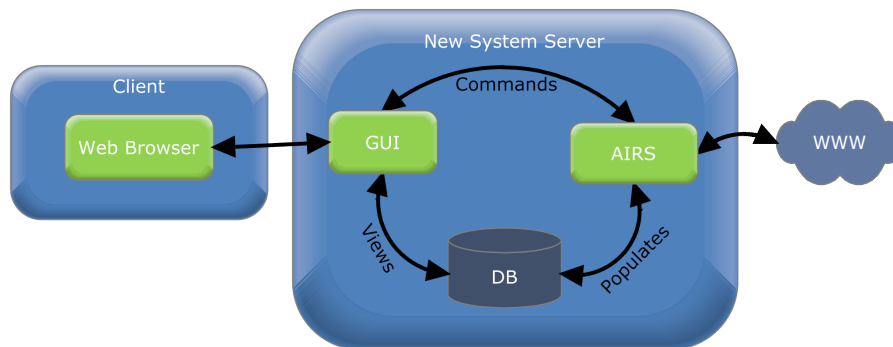


Figure 3.17: Global Project's Architecture

- **AIES** - The Automatic Information Extraction System component is responsible for acquiring all elements from various WWW sources and populate the Database with it.

The interactions between components are mainly:

- **Client-System** - Clients interact with the system through the GUI with a normal web browser.
- **GUI-AIES** - Any kind of the information extraction system control should be accomplished through the GUI that directly issues commands to the AIES.
- **GUI-DB** - The GUI uses the information stored in the DB to generate views of profiles and entity lists. All editing tasks in the GUI should submit the results to the DB too.
- **AIES-DB** - All AIES results should be submitted to the DB to populate the ontology. When the AIES needs to consult certainty information or verify already existing elements to then update them the queries should be made to the DB accordingly.
- **AIES-WWW** - To fulfil the AIES need of data many sources of information are needed. In this case they are represented by the World Wide Web. The AIES performs queries to the WWW and obtains raw results that need to be treated in order to then store in the DB.

3.4.1 Technologies

In terms of technologies to be used in this project, no limitations were imposed.

- The data base could be implemented in MySQL
- Scripting Languages for the information extraction: Perl [[per](#)], Python [[pyt](#)]
- Interface: Catalyst [[cat](#)], Django [[dja](#)], Javascript, Ajax

This technology decisions are mere possibilities, during the implementation of the project they may change.

3.5 Summary

This chapter presented the proposed architecture for the Automatic Collection of Biographical Data on the Web for Public Personalities System. A description of the main desired use cases, features and requirements was given.

System Specification

Chapter 4

System Implementation

A prototype of the new Automatic Collection of Biographical Data on the Web for Public Personalities System was implemented using the specification of the previous chapter as guideline.

The new system code-name is OntoExtract (OE).

The technologies chosen to implement the prototype were:

- **Database** - MySQL, since we have more experience with it and because it is used in most Sapo Labs projects, including Voxx.
- **Automatic Information Extraction System** - Perl [\[per\]](#), mainly because its performance with regular expressions when analysing huge text databases is very good and also because Sapo Labs researchers use it and could provide help with development.
- **Graphical User Interface** - Catalyst (Perl's Web Framework) [\[cat\]](#), because it's a Model View Controller written in Perl. Since the AIES was implemented in Perl, the best choice for module integration between the GUI and AIES is Catalyst. Javascript was also used for validation effects and support of extra features such as the pop-up windows with the graph images of entity discovery history.

This chapter presents the most relevant implementation details. It includes:

- Global explanation of sequence of events in OE;
- Most Relevant Database and Ontology decisions;
- Most Relevant Automatic Information Extraction System algorithms and heuristics;
- Most Relevant Automatic Information Retrieval functions;

- Implemented Graphical User Interface features.

In this chapter a top-down approach is used to describe the Onto Extract implementation details.

4.1 Database and Ontology decisions

This section contains all the database relevant decisions. Before that we need to refer that OE's implementation focused on using the Voxx's News database to obtain data for entity extraction. The process of extracting entities from journalistic sources will be explained in detail in section 4.2.

Considering that the main goal of this thesis project was to focus on the information extraction itself and not on the ontology, the database design is simple and was iteratively simplified from the beginning of the project. The final DB design contains 7 tables:

- **Person** - For each person in the database we store its *name*, *gender*, *title* (example: "Doutor"), *picture*, *country*, *address*, *birthday*, *death-day*, *criminal record*, *abstract*, *article link*, *tag HAREM* (which is a tag according to the rules defined in the *HAREM* initiative for recognition of entities mentioned in Portuguese) [MS08] and *notes*. Additionally some extra information is also kept, specifically an *incomplete flag* (represents the state of completeness of the person's profile), a *last modified* parameter (shows the last time the person's profile was modified) and a boolean field for each other entry (except the notes entry) that keeps track of the information that was already validated.
- **Organization** - For each organization in the database we store its *name*, *logo*, *country*, *head quarters address*, *foundation day*, *end day*, *slogan*, *abstract*, *article link*, *tag HAREM* and *notes*. Additionally some extra information is also kept such as an *incomplete flag* (represents the state of completeness of the organization's profile), a *last modified* parameter (shows the last time the organization's profile was modified) and a boolean field for each other entry (except the notes entry) that keeps track of the information that was already validated.
- **Role** - For each role in the database we store its *name*, *start date*, *end date*, *abstract*, *article link* and most importantly the associated *person's id* and *organization's id*. Additionally some extra information is also kept like an *incomplete flag* (represents the state of completeness of the organization's profile), a *last modified* parameter (shows the last time the organization's profile was modified) and a boolean field for each other entry (except the notes and the person's or organization's id entry) that keeps track of the information that was already validated.

- **Nickname** - For each nickname / known name in the database we store its *name* (meaning the record of the name discovered), *element's id* and *element's type flag*. Additionally a *valid name flag* that keeps track of the name's validity.
- **Related URL** - For each related url / reference in the database we store its *link*, *element's id*, *element's type flag* and a *type flag* that distinguishes if the entry is a related link or a reference. Additionally a *valid link flag* that keeps track of the name's validity.
- **Certainty** - For each certainty in the database we store its *method's certainty*, *final certainty* (due to certainty propagation), *number of hits* (number of pattern matches for the element in the news DB), *element's id* and *element's type flag* (person, role, organization or nickname). To be able to generate a certainty tree and link elements that are found to the elements that allowed these to be found we need to keep the *from element's id* and the *from element's type flag*.
- **User** - For each user in the database we store its *complete name*, *user name*, *password*, *role* (defines the type of access to the OE interface: administrator, operator and normal user), *e-mail*, *notes* and *status* (specificity if user is or not active and thus able or not to login the system). To be able to control user activity, a *last access* field is included.
- **News** - A copy of the Voxx's news table is included in the OE's DB for text mining purposes. The relevant fields that can be described here are the *content* field, where the actual news can be found and the *page url*, where the news links are kept.

Most of these tables are connected in some way, their connections can be analysed generally in the diagram [4.1](#).

As can be seen on diagram [4.1](#) there are several specific relations that are worth explaining:

- **Person-Role** - Each *Person* may have many *Roles* associated and each *Role* can only have one *Person* at a time. So if many *Persons* have the same *Role* name like "*primeiro-ministro português*" over time, a duplicate *Role* is created with different start and end date.
- **Organization-Role** - Each *Organization* may have many *Roles* associated and each *Role* can only have one *Organization* at a time. So if many *Organizations* have the same *Role* name like "*presidente-executivo*" or the same *Organization* has many mandates for the same *Role* over time, a duplicate *Role* is created with different start and end date.

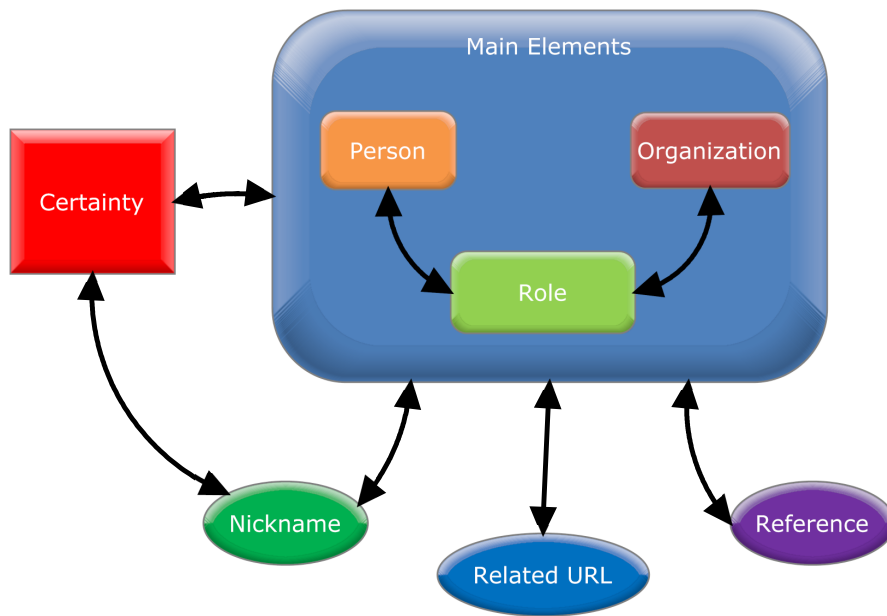


Figure 4.1: Generic representation of the Database connections

- **Main Elements-Nickname** - Every *Main Element* can have several *Nicknames* associated, a *Nickname* can only have one *Main Element* at a time associated to it. If several *Main Elements* have the same *Nickname* associated, a duplicate is created.
- **Main Elements-Related URL** - Every *Main Element* can have several *Related URLs* associated, a *Related URL* can only have one *Main Element* at a time associated to it. If several *Main Elements* have the same *Related URL* associated, a duplicate is created.
- **Main Elements-Reference** - Every *Main Element* can have several *References* associated, a *Reference* can only have one *Main Element* at a time associated to it. If several *Main Elements* have the same *Reference* associated, a duplicate is created.
- **Main Elements / Nickname-Certainty** - Every *Main Element* and *Nickname* always have one certainty entry directly associated. Since the *Certainty* table keeps track of the elements from which every element was discovered, every *Main Element* and *Nickname* may appear many times as the source from which other *Main Elements* and *Nicknames* were discovered.

4.2 Biographical Information Extraction

In this section the most relevant AIES algorithms and heuristics are described.

Although the specification of all sources is explained in subsection 3.1.2.1, the main source of information (from where the entities are identified in OE) is the news database used in *Voxx*, supplied by *Sapo Labs*. Additional information is then acquired from other sources to complete each entity profile.

4.2.1 Match Patterns

All entities are identified in the news content through journalistic text patterns. The use of appositive nouns as a way to define who is the *Person* associated to the *Role* is greatly used in journalistic text. OE uses patterns that include appositive nouns to identify elements. Two extraction patterns were considered in the extraction process using the appositive nouns and the use of regular expressions that traverse the results. The patterns are:

- **Fine** - a refined pattern that matches fewer times on the content and captures more precise results;
- **Rough** - a less refined pattern that matches with much of the content of analysed news but captures a considerable amount of noisy data.

The *Fine* pattern regular expression used in OE's extraction process can be observed in figure 4.2.

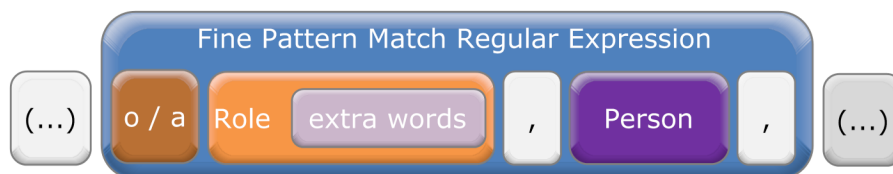


Figure 4.2: Fine Pattern Regular Expression Diagram

In order to properly understand the regular expression diagram 4.2 we need to mention that *Role* represents the general *roles* that the AIES wants to search for (like: "*presidente*", "*ministro*", "*administrador*", "*director*") in order to acquire persons related to it. The *extra words* stand for role defining words and possibly words that allow us to obtain the name of associated *Organizations*. One possible example of this kind of sentences can be seen in Figure 4.3.

In this sentence, with the *Fine* pattern, we can extract three things: the *person's gender*, the *person's role* and the *person's name*. The gender can be captured by the "*o*" a the beginning of the sentence, because in Portuguese, whenever the role is preceded by "*o*" it means that the person associated is a man, and when the role is preceded by "*a*" it means the person associated is a woman. The role can be represented by the words between "*o*" or "*a*" at the beginning of the match and the first comma (which defines the beginning of



Figure 4.3: Example of Fine Pattern Match Diagram

the appositive nouns). It is important to mention that the *extra words* inside the *Role* capture only allow the role's extra defining information to have 50 characters at a maximum. This constraint avoids matching sentences that are "false" appositive structures (sentences that fit in grammatically but do not comply with the appositive structure because words between commas do not define any previous role). Finally the person's name can be acquired by extracting the words between the two commas in the match. Basically when the AIES tries to acquire a match for a new known role with the *Fine* pattern, for instance the role "*presidente*", the *Role* space becomes "*presidente*", as can be seen in the following example of a regular expression 4.4.

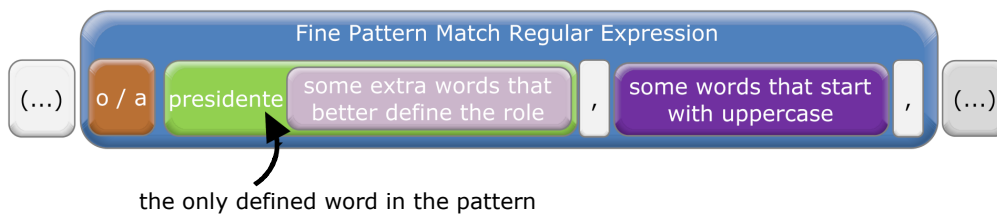


Figure 4.4: Example of Fine Pattern Match Diagram for Role "presidente"

Here the "*o / a*" would capture the gender of the person, the "*presidente + some extra words that better define the role*" would capture the specific role of the person and finally the "*words that start with uppercase*" would capture the person's name. An example of a match for regular expression example 4.4 could be seen in Figure 4.5.

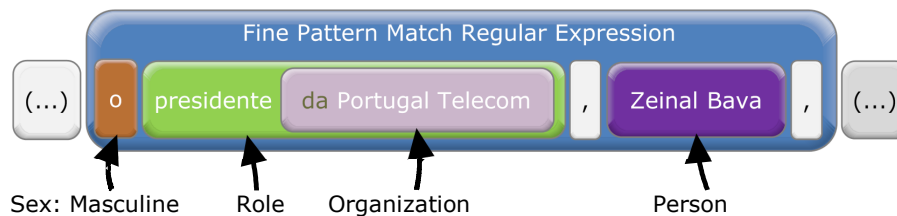


Figure 4.5: Example of result for Fine Pattern Match Diagram for Role "presidente"

In this particular match, we can even extract the *Organization's* name to which the role belongs to. From the identified *Role*, *presidente da Portugal Telecom*, the *Organization's* name can be extracted, in this case, *Portugal Telecom*.

Several cleaning functions were implemented namely a function to clean *Person's names*, one to clean *Organization's names* and one to clean *Role's names*. Each of these functions tries to remove, as much as possible, the noisy data captured along with the information that interests OE. Additionally a function that tries to identify *Organizations* from the *Role* captured was implemented. For further reading on the subject, please go to subsection 4.2.3.

The *Rough* pattern's regular expression used in OE's extraction process can be analysed in regular expression 4.6.

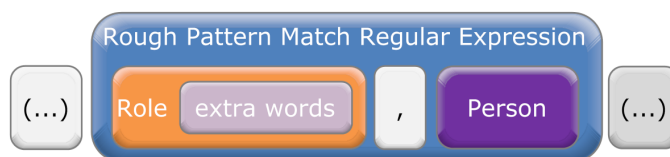


Figure 4.6: Rough Pattern Regular Expression

As can be seen in figure 4.6 the *Rough* pattern is not as specific as the *Fine* pattern, allowing the system to obtain a much bigger result set (with less accurate results). In this pattern, the gender is only identified if there is a "o" or a "a" before the *Role*; the pattern does not force that to occur. This pattern also does not force the limit of the role *extra defining words* to a maximum of fifty characters, thus allowing to match "false" appositive nouns. Lastly, this pattern does not force the appositive nouns to appear between two commas, they can appear between two commas or a comma and a full stop.

Valid match results for the *Fine* and *Rough* patterns are the same. Although the *Rough* method has a lot more matches, the percentage of valid results for the *Fine* method is much better than for the *Rough* method, as we can see comparing by tables 4.1 and 4.2.

For each pattern previously defined, manual tests were done to calculate certainty percentages of the information extracted by each one. Each one was analysed by counting the number of correct matches for several different *roles*. The five *roles* used for the evaluation of method certainty were chosen randomly. With more tests the method's certainty evaluation became more precise, but since all evaluations had to be done manually, it was not practicable to test more. As can be seen in tables 4.1 and 4.2 the difference in number of matches is substantial as well as the difference in correct names captured associated to the *roles* specified. Nonetheless it is interesting to observe that although the *rough* pattern obtained more incorrect matches, the number of correct results are still considerably more when compared to the *fine* pattern. The *rough* pattern obtained one hundred and eighty one more matches in total results and one hundred and six more correct results in relation to the *fine* pattern, but the difference between them in error percentage is of twenty one percent, which means that the *rough* pattern has almost three times the error percentage in results obtained when compared with the *fine* pattern. Feel free to compare them

in Tables 4.1 and 4.2.

Roles Tested	Names Discovered	Correct Names	Precision Percentage
"primeiro-ministro"	48	44	91
"presidente da república"	24	19	79
"presidente do PSD"	2	2	100
"presidente da PT"	2	2	100
"treinador do Benfica"	5	5	100
Total	81	72	88

Table 4.1: Fine Pattern person's name identification test

Roles Tested	Names Discovered	Correct Names	Precision Percentage
"primeiro-ministro"	146	101	69
"presidente da república"	71	48	67
"presidente do PSD"	28	12	42
"presidente da PT"	3	3	100
"treinador do Benfica"	14	14	100
Total	262	178	67

Table 4.2: Rough Pattern person's name identification test

For this experimental validation the total *precision percentage* of these tests were considered as an estimation of the defined *method's certainty* for each pattern. So in the calculation of each discovered element's *final certainty*, the *method's certainty* considered is sixty seven percent if the element was discovered using the *rough* pattern, and eighty eight percent if it was discovered using the *fine* pattern. Further reading about the process of certainty calculation can be done in subsection 4.2.5.

4.2.2 Name Extraction

In this subsection the implemented name extraction overall process is described. In the next sections each relevant part of the name extraction process is described with further detail.

After a match on the news, the process of acquiring the desired information, in this case the names of the entities that exist in every match, begins. Every match represents a relation, because in every match at least a *person's* name and a *ergonym/role's* name is specified; sometimes even the *organization's* name can be extracted. If after the process of acquiring every part's names in this relation we do not have at least a *person's* name and an *ergonym/role's* name or an *organization's* name and a *role's* name, the match is annulled. Only a match that can map a relation is considered. Having only a *person*, *organization* or *role's* name is not relevant if it is not related to other names. The *person's* gender and *person's* / *role's* raw name can be extracted directly from the regular expressions that

match with the news, to obtain *organization*'s names a second process of extraction needs to occur from the extracted *role*'s raw name. The implemented solution for this problem was to find the words "da", "das", "de", "do", "dos" in the raw *role*'s name. When one is found all the words that come after it will be considered a possible *organization*'s raw name that needs to be cleaned with the implemented *Organization's Name Cleaning Function* that is described in subsection 4.2.3.2. The process of extracting *organization*'s raw names is better described in Figure 4.7.



Figure 4.7: Organization's name extraction regular expression

After all available raw names are extracted, they are sent to their respective cleaning functions. Nevertheless before starting the name cleaning process a small check is done to ascertain the validity of the match. This validation consists in verifying if the next bullet points occur.

- The person's raw name:
 - has more than three characters - a complete name with only three characters does not contain enough distinguishing information to associate it to a certain person;
 - starts with a capital letter;
 - ends with a word that is capitalized.
- The role's raw name:
 - has more than three characters - a complete role with only three characters is not specific enough to be of use.

If these minimum requirements are not fulfilled then the match is ignored and the news match analysis proceeds to the next match. If instead they are fulfilled, then the name cleaning process starts. The name extraction process can be generally analysed in the diagram of sequence of events present in Figure 4.8.

After obtaining all the cleaned names, the nickname identifying process starts. In this process all *person*'s names are compared and if they are very similar, they are considered to be *nicknames* for the same *person*, the one with the bigger number of matches on the news is considered the main name. Consequently the others are considered the *person*'s *nicknames*. The same happens to *organization*'s names and *role*'s names. Further reading

System Implementation

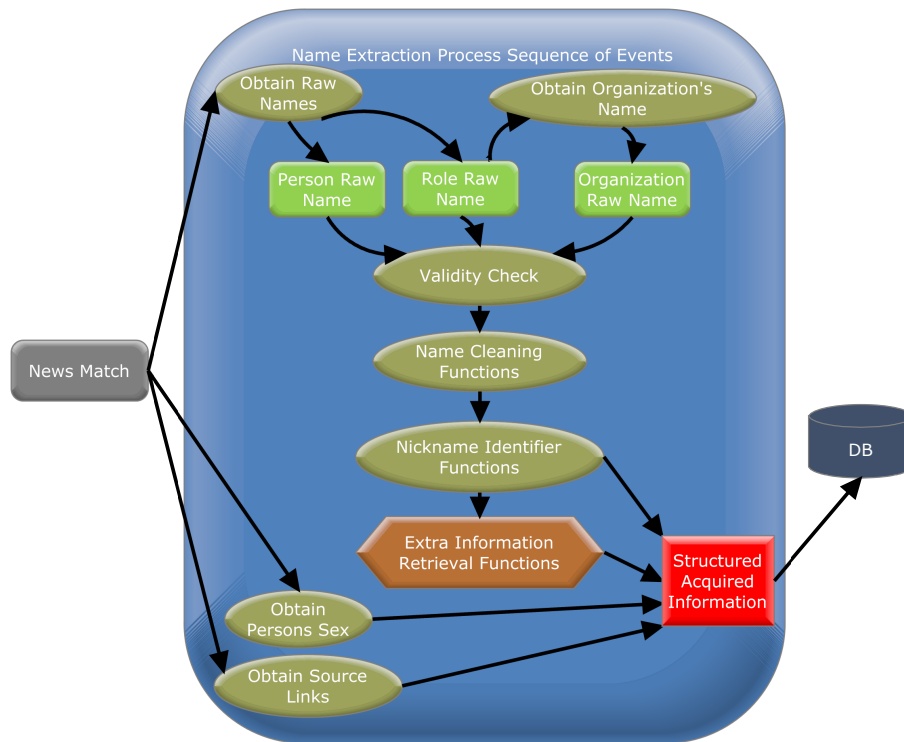


Figure 4.8: Entity name extraction process general sequence of events

on this subject is available in subsection 4.2.4. The extracted *person's gender* in each match is kept, in the end the gender with more hits for that person is the one that will be considered for DB storing purposes. Additionally every news source link (which is one of the entries in the news table on the DB) is kept, so that every *person* in the system has a list of *related URLs* with all the links where any information about it was found. The same happens for every *role* and *organization*. All sources of information are always kept. After all entity names are discovered, the information retrieval process begins to find additional information about each one. This process consists in acquiring images, abstracts and article links for each entity in DBpedia dumps of Wikipedia. The information retrieval process is better explained in subsection 4.3. Finally, after all information is gathered the DB storing process begins.

For debugging effects and to better control the entity's names extraction process, a log system was implemented to view the entities being found for each *role / ergonym* entry.

As can be seen in Log File 4.1 example, all *person / organization / role's* names found are already separated with their nicknames.

4.2.3 Name Cleaning Functions

Sometimes, in journalistic text, the appositive nouns structure is not respected, and instead of having a *person's / organization's / role's* name we may end up with their names

System Implementation

```
Waiting MySQL answer to query: presidente...
Received MySQL answer to query: presidente!
Beginning name extraction process...
```

```
----> Persons Found:
```

```
=====
Barack Obama => 958
Barak Obama => 5
Barack Obama => 1
Barack Obamam => 1
=====
Cavaco Silva => 953
Aníbal Cavaco Silva => 605
AníbalCavaco Silva => 3
Aníbal CavacoSilva => 1
=====
Manuela Ferreira Leite => 431
Manuela Fereira Leite => 1
Manuela^M Ferreira Leite => 1
=====
António Costa => 378
António => 93
António Sá da Costa => 7
=====
Durão Barroso => 334
José Manuel Durão Barroso => 97
Durão Barros => 3
DurãoBarroso => 1
=====
Nicolas Sarkozy => 290
Nicholas Sarkozy => 6
Nicólas Sarkozy => 1
Nicolas Sarcozy => 1
Nicolas Sarzozy => 1
=====
Hugo Chávez => 269
(...)
```

Log File 4.1: OntoExtract's entity extraction log file example for role *presidente*

followed or preceded by extra information, or we may even have no name at all.

To avoid this kind of problems, we implemented some "cleaning" methods. They are used to obtain a clean name or none at all, assuring that the name stored in the DB is actually a *person / organization / role*.

4.2.3.1 Person's Name Cleaning Function

To evaluate if the captured name is actually a *Person's* name, first we created lists of words that are not commonly found in *Person's* names. The lists created in order to verify if a person's name is real are described in table 4.3.

List of:	Nr. of entries	Description
Conjugated Verbs	191720	List of 4000 verbs conjugated that can not match in a name
Communication Verbs	111	List of the most used special communication verbs
Nationalities	171	List of the most common nationalities, to avoid matching cases like: " <i>o espanhol Zapatero</i> "
Countries	318	List of most of the countries / states
Cities	155	List of most relevant cities in the World and most of the Portuguese cities
Organizations Types	18	Most common organization types that cannot match inside a person's name like: " <i>comité</i> ", " <i>governo</i> ", " <i>clube</i> "
Extra Terms	36	Most commonly used journalistic terms that cannot match a person's name
Total	192529	-

Table 4.3: Lists of names that cannot appear in a *Person* name

The lists previously created are loaded from separate files at the beginning of the extraction process to generate a global hash of words that can not match a person's name. The person's cleaning function accepts as input the extracted raw *person* name from the news match. If the raw *person's* name has only one word and it matches one of the words in the lists, the *person's* name is immediately discarded and the match is cancelled. If the *person's* name has more than one word, a cycle of regular expressions is applied until the result is just that *person's* name. Basically, this first-stage cycle of name-cleaning regular expressions determines the location of the unwanted word in the raw *person's* name and tries to remove it accordingly. The regular expressions used in this cycle can be analysed in Figure 4.9.

Analysing Figure 4.9 we can see in green color what is kept after this first stage cycle of a *person's* name cleaning. If the *person's* raw name has any of the words stored in the global list of words that should not match a person's name, the cleaning cycle begins. If the *Unwanted Word* is at the end or the beginning of the raw name, it is removed. In

System Implementation



Figure 4.9: Regular expressions used in first stage *Person's* name cleaning

case the *Unwanted Word* is in the middle of the raw *person's* name, then the cleaning cycle keeps the person's raw name until the appearance of the referred word. After each iteration, the cleaning cycle shown in figure 4.9 tries to find new matches until a still raw *person's* name without unwanted words is acquired in the end of the cycle. After the initial cleaning cycle, the resulting name might not be a *person's* name and so a new cleaning and verification needs to be done to ensure it is a person's name.

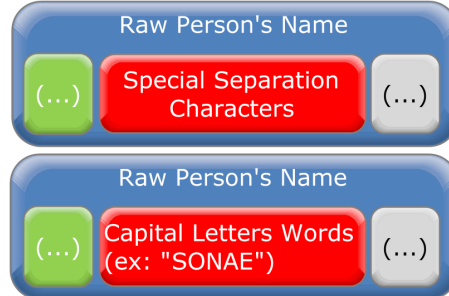


Figure 4.10: Second stage *Person's* name cleaning regular expressions

Analysing Figure 4.10 we can see in green color what is kept after this second stage cycle. These cleaning tasks are mainly used to detect existence of numbers or special separation characters in the *person's* name, such as opening / closing brackets, opening / closing square brackets, quotation marks, inverted commas, semicolons, colons and dashes. Another verification performed in this stage is the removal of words that are only composed of capital letters. This is done by keeping only the raw *person's* name until the first capital letter word appears.

At this point the raw person's name is already striped of most common noisy data that normally appears in journalistic text. To conclude the person's name cleaning, a final regular expression described in Figure 4.11 is applied.

System Implementation

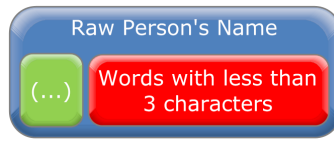


Figure 4.11: Final stage *Person's* name cleaning final verifications

Analysing Figure 4.11 we can see in green what is kept after this final stage cycle of *person's* name cleaning. The filtering verifies if the last word of a name is a very small word. If so, the regular expression discards it. This verification deals with cases where the final word is a word like "*do*", "*da*", "*de*", which are valid words inside a *person's* name but not at the end. An example of this specific case is shown in diagram 4.12, where it becomes clear that it is common to find this kind of words in the final *person's* names given the previous regular expressions.

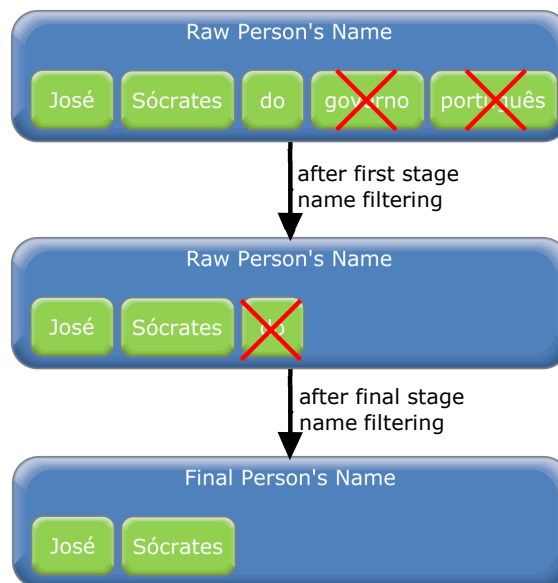
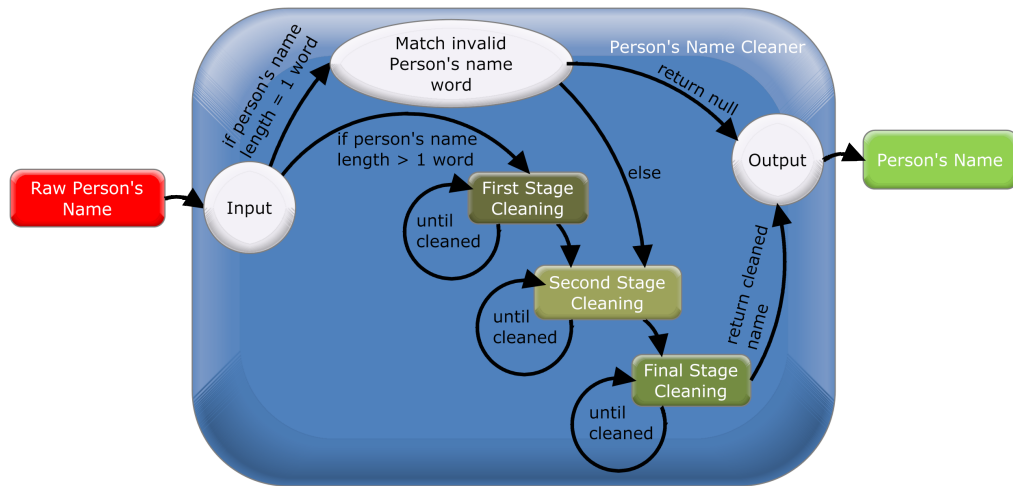


Figure 4.12: Sequence of filtering stages in name cleaning

To better understand the whole *person's* name cleaning process, we can see the sequence of events in the *person's* name cleaning function in diagram 4.13. Note that in Figure 4.13 the *First Stage Cleaning*, *Second Stage Cleaning* and *Final Stage Cleaning* refer to the ones explained in this subsection, and their regular expressions are the ones being checked at each iteration; until no regular expression matches at each stage, it continues to iterate.

Figure 4.13: *Person's name cleaner* sequence of events

4.2.3.2 Organization's Name Cleaning Function

In order to evaluate if the captured name is actually an *Organization's* name, we generated lists of words that are not commonly found in *Organizations's* names. The list used to verify if the organization's name is real is described in Table 4.4.

List of:	Nr. of entries	Description
Communication Verbs	111	List of most used special communication verbs
Total	111	-

Table 4.4: Lists of names that can not appear in an *Organization's* name

This list created is loaded at the beginning of the extraction process, to generate a global hash of words that should not match an organization's name. The organization's cleaning function has accepts the extracted raw *organization* name from the news match as input. Unlike the *person's* name cleaner function, if the raw *organization's* name globally matches one of the words in the lists, the *organization's* name is immediately discarded and the match is cancelled. If the *organization's* name does not match any of the invalid words, some cleaning tasks need to occur to verify if there are not any numbers or special separation characters in the *organizations's* name, like opening / closing brackets, opening / closing square brackets, quotation marks, inverted commas, semicolons, colons and dashes. This first *organization's* name cleaning stage is very similar to the second stage in the *person's* name cleaning process described in Figure 4.10. They differ in that the special separation characters may appear at the beginning or the end of the name, and the capital letters verification does not exist because many *organizations's* names are precisely composed of capital letters, mostly because they stand for initials of other words. This first stage can be analysed in Figure 4.14.

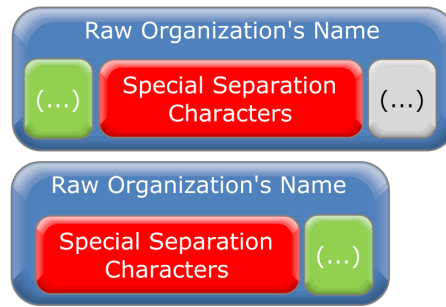


Figure 4.14: First stage *Organization's* name cleaning regular expressions

Analysing Figure 4.14 we can see in green what is kept after this first stage cycle of *organization's* name cleaning. The second and final stage of *organization's* names filtering is very similar to the final verification of *person's* names, and checks if the last word of a name is a very small word. The difference between this and the person's name cleaning final stage is that this one verifies if the last word is composed by only one character like "à", "a", "o", and the person's final cleaning stage verifies if the last word is smaller than three characters, mainly because many organization's names finish with a two character word, for instance "grupo SL", "SGPS SA".

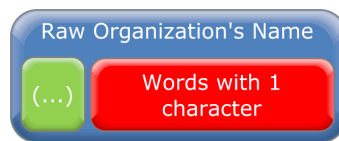


Figure 4.15: Final stage *Organization's* name cleaning final verifications

Analysing Figure 4.15 we can see in green what is kept after this final stage cycle of *organization's* name cleaning.

Summarizing this function's description, we can see in diagram 4.16 the sequence of events in the *organization's* name cleaning function. Note that in Figure 4.16 the *First Stage Cleaning* and the *Final Stage Cleaning* refer to the one's explained in this subsection, and their regular expressions are the one's being verified at each iteration; until no regular expression matches in each stage, it continues to iterate.

4.2.3.3 Role's Name Cleaning Function

To be able to evaluate if the captured name is actually an *Role's* name first created lists of words that are not commonly found in *Role's* names. The only list used to verify if the role's name is real is the same used for the purpose of cleaning organization's names described next in Table 4.4. The list created is loaded at the beginning of the extraction process to generate a global hash of words that must not match an role's name. The role's

System Implementation

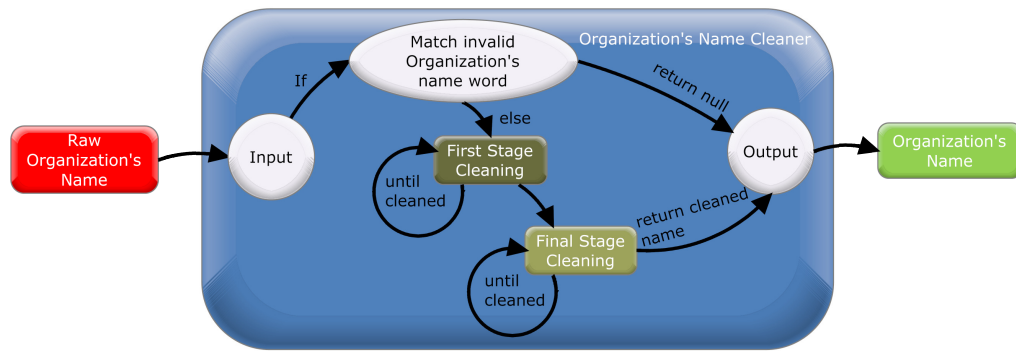


Figure 4.16: *Organization's* name cleaner sequence of events

cleaning function accepts as input the extracted raw *role* name from the news match. Just like the *organization's* name cleaner function, if the raw *role's* name globally matches one of the words in the lists, the *role's* name is immediately discarded and the match is cancelled. If the *role's* name does not match any of the listed invalid words, some cleaning tasks need to occur to check if there are no numbers or special separation characters in the *role's* name, like opening / closing brackets, opening / closing square brackets, quotation marks, inverted commas, semicolons, colons and dashes. This first *role's* name cleaning stage is very similar to the first stage in the *organization's* name cleaning process described in Figure 4.14; the difference is that the special separation characters may only appear in the middle of the name.

The capital letters verification, just like in the *organization's* name cleaning process, is not performed. This is due to the fact that all *organization's* names are extracted from within a *role's* name. This results in the fact that the *role's* name cleaning process needs to have the same name cleaning restrictions as the *organization's* name cleaning process. This first stage can be analysed in Figure 4.17.



Figure 4.17: First stage *Role's* name cleaning regular expressions

Analysing Figure 4.17, we can see in green what is kept after this first stage cycle of *role's* name cleaning.

The second and final stage of *role's* names filtering is exactly like the final stage of the *organization's* name cleaning process 4.15. It consists in checking if the last word of a name is a very small word. To better illustrate this final stage please analyse Figure 4.18.

System Implementation

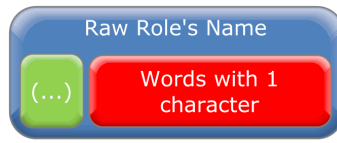


Figure 4.18: Final stage *Role's* name cleaning final verifications

Analysing Figure 4.18 we can see in green what is kept after this final stage cycle of *role's* name cleaning.

To summarize this function's description, we can see the sequence of events in the *role's* name cleaning function in diagram 4.19.

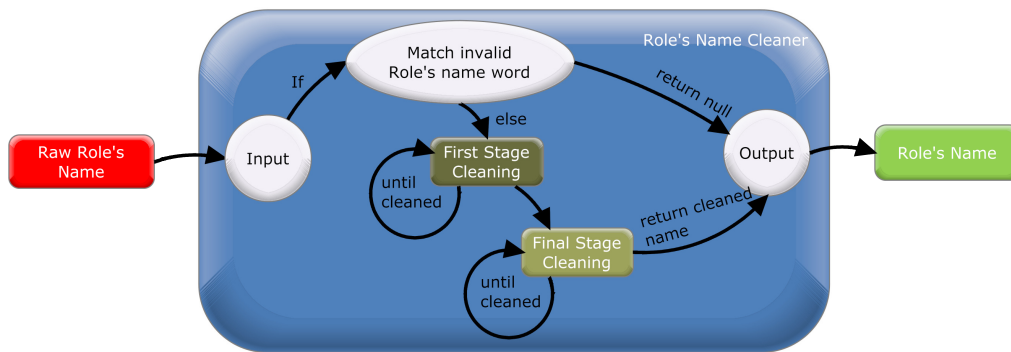


Figure 4.19: *Role's* name cleaner sequence of events

4.2.4 Name Similarity Functions

In this subsection, we describe the name similarity functions. They are necessary to infer if various different names are referring to the same entity.

The main name similarity function is the *nickname identifier*. Its implementation includes a modified version of the *Levenshtein Distance Algorithm* [lev] as a base distance calculator between two words.

The inputs for the *nickname identifier function* are:

- **accepted difference percentage** - value below which two names are considered similar enough to represent the same entity;
- **list of names and their hit counts** - list of all names acquired from a name extraction method's result and their respective number of appearances in news matches.

First, we sort the list of names according to the number of times they appear in the news matches. This sorting process is of crucial importance to start the nickname identifier process with names that are most probably the main real names, being the most mentioned in the news. After choosing the name with the highest number of occurrences

in the news matches, it is converted in an array with all its composing words, and is deleted from the list of all found names. The next step compares each of its composing words to the other names in the result list. This is accomplished with three main comparison methods, depending on the name length in terms of word count.

Designating the name we want to find nicknames as $N1$, and each other name that will be compared as $N2$, the next Algorithm Pseudo-Code 4.1 illustrates the three methods according to the number of words in $N1$ and $N2$.

- $N1$ has more than two words:
 - $N2$ has at least two words:
 - * Final Distance = $(65\% * \text{distance between last name of } N1 \text{ and } N2) + (35\% * \text{distance between first name of } N1 \text{ and } N2)$
 - * If the Final Distance calculated is still greater than 25%, then Final Distance = $(65\% * \text{distance between last name of } N1 \text{ and } N2) + (35\% * \text{distance between the penultimate name of } N1 \text{ and } N2)$
 - $N2$ has only one word:
 - * Final Distance = distance between last name of $N1$ and $N2$
 - * If the Final Distance calculated is still greater than 25%, then Final Distance = distance between first name of $N1$ and $N2$
- $N1$ has exactly two words:
 - $N2$ has at least two words:
 - * Final Distance = $(55\% * \text{distance between last name of } N1 \text{ and } N2) + (45\% * \text{distance between first name of } N1 \text{ and } N2)$
 - $N2$ has only one word:
 - * Final Distance = distance between last name of $N1$ and $N2$
 - * If the Final Distance calculated is still greater than 25%, then Final Distance = distance between first name of $N1$ and $N2$
- $N1$ has only one word:
 - Final Distance = distance between last name of $N1$ and $N2$

Algorithm Pseudo-Code 4.1: Nickname Identifier Algorithm Pseudo-Code

As can be seen in Algorithm 4.1, name comparison is performed taking in consideration the importance of each word in the name. The importance percentage decided for each was decided based on experimental tests made during the project implementation.

As previously mentioned, each distance calculation specified in Algorithm 4.1 was made through a modified version of the *Levenshtein Distance Algorithm*, which calculates the relative difference between two names. The *Levenshtein distance* between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. [lev] Note that if a name is fully contained in the other, the distance between the smaller one to the other is 0 and the distance of the bigger one in comparison to the smaller is the number of extra characters. That's why the distance calculations done in

the *nickname identifier* algorithm are always done comparing one of the relevant names of *N1* against the *N2* whole name.

The *nickname identifier* algorithm showed in Algorithm 4.1 only considers two names similar enough if the input percentage is higher than the one obtained in the combined distance calculation of the names *N1* and *N2*.

The first step is to determine how many words the names *N1* and *N2* have. If *N1* has more than two words and *N2* has at least two words, the final distance between the two is calculated considering the distance between the *N1*'s last name and *N2*'s full name, and between the *N1*'s first name and *N2*'s full name. The percentage given to the two comparisons in order to obtain a combined distance result is of sixty-five percent and thirty-five percent, respectively. If after that, *N1* and *N2* combined distance result is still different enough to overcome twenty five percent, the final distance is re-calculated, this time using the penultimate name of *N1* instead of the first. The penultimate name is used because many public personalities are called by their two last names – a good example of this is "*José Luís Rodríguez Zapatero*" only called "*Rodríguez Zapatero*".

If *N1* has more than two words and *N2* has only one word, the final distance between the two is calculated only considering the distance between the *N1*'s last name and *N2*'s name. If after that, *N1* and *N2* combined distance result is still different enough to overcome twenty five percent, the final distance is re-calculated and this time using the first name of *N1* instead of the last. This is useful to merge names like *Barack Obama* and *Barack*.

If *N1* has exactly two words and *N2* has at least two words, the final distance between the two is calculated considering the distance between the *N1*'s last name and *N2*'s full name, and between the *N1*'s first name and *N2*'s full name. The percentage given to the two comparisons to obtain a combined distance result is of fifty-five percent and forty-five percent respectively.

If *N1* has exactly two words and *N2* has only one word, the final distance between the two is calculated only considering the distance between the *N1*'s last name and *N2*'s name. If after that *N1* and *N2* combined distance result is still different enough to overcome the barrier of twenty five percent, the final distance is re-calculated and this time using the first name of *N1* instead of the last.

Finally if *N1* has only one word, the final distance between the two is calculated only considering the distance between the *N1*'s only name and *N2*'s name. A example of this case is for instance comparing "*Mourinho*" with the name "*José Mourinho*".

When similar names are found, they are all kept in a list together with their respective number of hits. They are also removed from the original list of all names so that they are not matched again in the next iteration. The process endures until the original list of all found names becomes empty.

At the end of the nickname identifier process the function returns a list of all compiled lists of similar names.

The input percentage of difference accepted between two names needs to be adjusted for each entity type. Table 4.5 shows the percentage used for each one of the main entities that need to have similar names associated to them. These percentages were also improved iteratively along the projects development, through experimental tests. Some justifications for the percentages given to each entity comparison are available in the *Description* column of Table 4.5.

Entity Type	Accepted Difference	Description
Person	15%	The easiest name comparison. Usually, famous persons are called by their last name or a relatively unique combination of names.
Organization	2%	Very hard to ensure that nicknames really belong to the same Organization; there are many similar acronyms.
Role	0.5%	Even harder, due to many similar organizations acronyms and role descriptions.

Table 4.5: Percentage of accepted differences for each entity type

To better illustrate the nickname identifier sequence of events, please analyse the diagram presented in Figure 4.20.

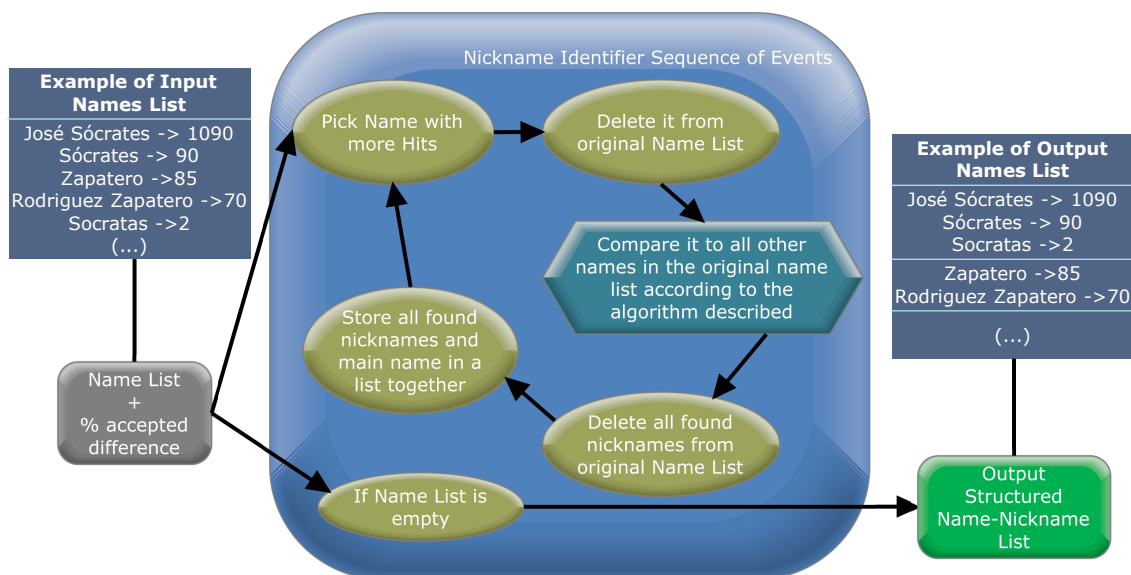


Figure 4.20: Nickname identifier sequence of events

4.2.5 Certainty Calculation

In this section the process of certainty calculation specified in subsection 3.1.2.3 is described, as well as some techniques to augment the certainty of an discovered entity.

An element has always two types of certainties that are kept in OE associated with it:

- **Method's Certainty** - The certainty of the pattern match calculated when the element was found in the news;
- **Final Certainty** - The *method's certainty* times the *final certainty* of the element from which the referring element was discovered.

Every certainty that is calculated with the AIES never exceeds ninety-five percent. The extra 5% are reserved for human validations, allowing to differentiate information that was already validated by an human operator.

To calculate the *method's certainty* two parameters are taken in consideration:

- **Pattern's Fixed Certainty** - Certainty previously calculated through the use of experimental tests in order to ascertain the percentage of correct results. The ones used in OE can be analysed in Tables 4.1 and 4.2;
- **Number of Matches in the News** - The number of times the entity's name appeared in the news with one pattern match.

The *number of matches in the news* are used to increment the certainty of the method to a maximum of 95%. This increment should be done according to the meaning that the number of appearances in the news pattern matches has. For instance, if a certain entity name appears only once, the certainty of this entity associated to the respective role match, is not significant. Instead, if a certain name appears three thousand times associated to a certain role, then the level of certainty that both belong together is almost absolute.

To this effect, a top barrier, in number of times that the name of the entity needs to appear in the news to ensure its veracity, was defined. This top number was arbitrarily defined as 50, because when an entity's name appears 50 times associated to a certain role, the possibility of being a wrong name or relation is very remote.

A gradual evolution needed to be implemented in a way that the resulting certainty represents a reliable approximation to the real meaning the number of hits has. To solve this problem, we propose the usage of a logarithmic function, since it evolves in a way that the difference between 5 or 10 hits is greater in significance of importance than the difference between 25 and 30 hits.

When a certain entity is found 5 times it's certainty is already relatively high because it means that the same relation entity-role occurred in 5 different news. This certainty evolves until reaching a point where the certainty level is almost absolute. Since the top

barrier defined is 50, when an entity is found 50 times associated to a certain role or more, the sum to the *method's certainty* must be the exact the difference between it and 95%, to assure that the maximum certainty of everything that is automatically extracted does not overcome the established limit. The implemented solution was based on the logarithm of base 50, and can be analysed in Figure 4.21.

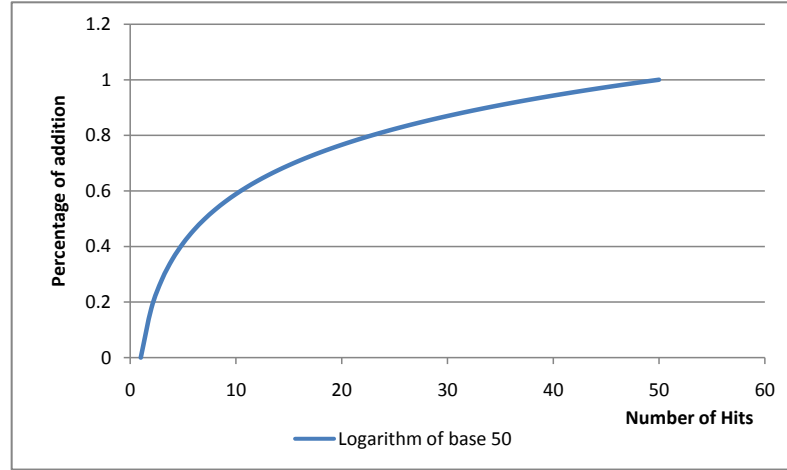


Figure 4.21: Logarithm of base fifty used to calculate increase in certainty

This logarithm maps a possible approach to the addition problem. When the *number of hits* is 1, the logarithmic function is 0, which means the addition is not performed. When it is 50 the logarithmic function is 1 which means that the total difference between the *method's certainty* and 95% will be added to the *method's certainty*. This is just one of the possible solutions and the chosen for this experimental validation.

For a *fixed pattern's certainty*, PC , and a number of matches in the news NH , the final *method's certainty* calculation algorithm can be analysed in Algorithm 4.2.

- **Method's Certainty** = $PC + \log_{50}(NH) * (100 - PC)$
- If (**Method's Certainty** > 95) then **Method's Certainty** = 95
- Return **Method's Certainty**

Algorithm Pseudo-Code 4.2: Certainty Calculation Algorithm Pseudo-Code

As an example, imagine that the name "*Michael Jordan*" was found 12 times in the news associated to an already known role, "*jogador de basquetebol*". The role "*jogador de basquetebol*" was previously found with a final certainty of 78%. Suppose that the pattern that allowed us to find the name "*Michael Jordan*" with the role "*jogador de basquetebol*" has a fixed certainty of 67%. The *method's certainty* and *final certainty* calculation of the person "*Michael Jordan*" is described in Algorithm 4.3.

A way to improve a *method's certainty* is to combine various match patterns in the news in order to try to obtain the same information, using the example described above,

- $Nr_Hits = 12$
- $Role_certainty = 78\%$
- $Pattern_certainty = 67\%$
- $Method_certainty = Pattern_certainty + \log_{50}(Nr_Hits) * (100 - Pattern_certainty) = 67 + \log_{50}(32) * (100 - 67) = 87\%$
- $Final_certainty = Method_certainty * Role_certainty = 87\% * 78\% = 68\%$

Algorithm Pseudo-Code 4.3: Certainty Calculation Algorithm Pseudo-Code

we could try to use a new pattern with the name "*Michael Jordan*" to try to find the same role instead of only using the pattern with the role "*jogador de basquetebol*". If there is a match for this new pattern, then the *method's error certainty* of both could be multiplied, improving the *method's certainty* of the entity. If the name "*Michael Jordan*" was found associated to the role "*jogador de basquetebol*" with a *method's certainty* of 50% with the new pattern match, then the *method's error certainty* of the name "*Michael Jordan*" is going to be updated from 13% to 13% times 50%, which is approximately 7%. This improves the method's certainty from 87% to 93%.

4.3 Biographical Information Retrieval

In this section we describe the implemented Information Retrieval functions. All implemented information retrieval functions developed use the DBpedia dumps as source of information. They were developed to complement the information extracted from the news. Three functions were developed:

- **Abstract Retrieval Function** - obtain the Abstract for a given entity name;
- **Article Link Retrieval Function** - obtain an Wikipedi Article Link for a given entity name;
- **Photo Link Retrieval Function** - obtain a Photo for a given entity name.

The three correspondent DBpedia dumps were processed in order to have only one entity name and the respective links / abstract in each line.

When, during the information extraction process, the main name is already defined, each of the above referred functions is called to attempt retrieval of the Abstract's / Article Links / Photo Links match for the name in question. no soft-matching is done when there is not a direct match between the name and any name in the DBpedia dumps. The first nickname for the same entity (the one with more number of hits after the main name) is used instead, in order to try to match again. This process repeats itself until there is a match or no more nicknames are available. The information retrieval system is described briefly in Figure [4.22](#).

System Implementation

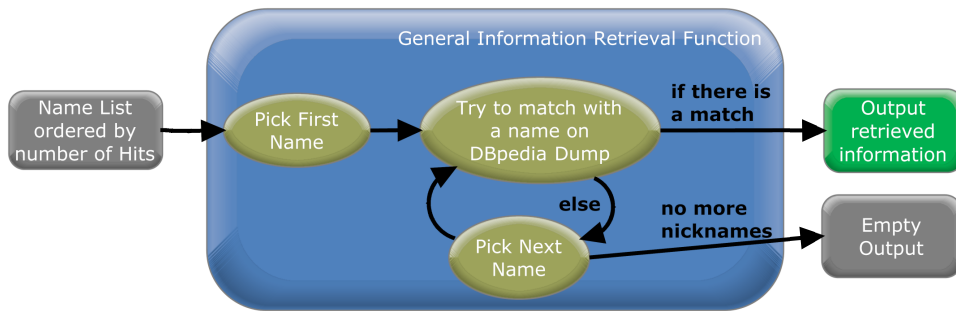


Figure 4.22: Generic Information Retrieval Function Diagram

Note that the dumps used are of version 3.5.1 which was launched in March 2010 and some links are already broken. Another problem is that although the abstract and the article link dumps are available for the Portuguese Wikipedia, the photo links dump is only available for the English Wikipedia. This means that most Portuguese public entities do not match with names in the photo links dump, either because they do not exist or because their names were translated to English and do not match, for instance the name "*Fernão de Magalhães*" exists in the English DBpedia dump as "*Ferdinand Magellan*".

4.4 Graphical User Interface

This section describes the graphical user interface and its implemented features. The graphical user interface was implemented in *Catalyst*, a Model / View / Controller framework for *Perl*.

The main window of OntoExtract's GUI is shown in Figure 4.23. Here, an unauthorized user must login to become one of the three authenticated users: *Normal User*, *Operator User* or *Administrator User*.

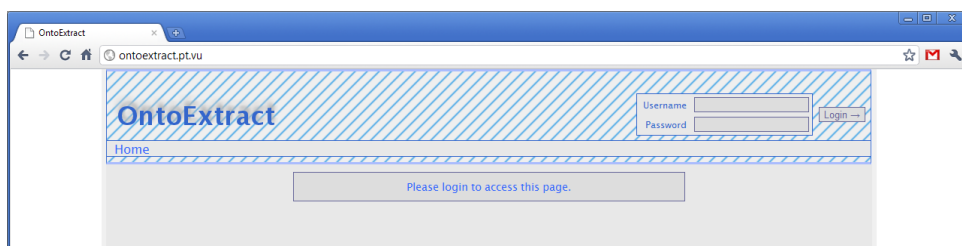


Figure 4.23: Main OntoExtract's GUI page

The only available option in this page is to login using the *user name* and *password* credentials that will allow the user to browse inside OntoExtract. After logging in, according to the user type, many options become available. The common features available are *Person List*, *Organization List*, *Role List* and *Edit Profile*. In the *Person List*, a list

of available person's profiles on OntoExtract appears. Each page of the list contains ten persons at a time, and the list of persons has a few general information available to help choose the right person to browse: *Photo*, *Name*, *Country*, *Birthday* and *Abstract*. Additionally in this page, a *Normal User* can search for parts of person's names, and OE shows a list of all the matches. In Figure 4.24 we can see a result page for a search for names with the word "Bill" in the *Person List*.



Figure 4.24: OntoExtract's *Person List* page after searching for name "Bill"

The *Organization List* and the *Role List* are similar and have the same features as the *Person List*. Each row is clickable in these types of listing pages, taking the user to the corresponding entity's profile. Figure 4.25 shows us a *person's* profile, in this case "Bill Gates", that was opened clicking on his name from the page represented in Figure 4.24.

On top we have the person's photo on the right and the *person's* discovered information on the left, in this case the *gender*, and the *abstract*. Note the green ticks next to the top name, gender, abstract and photo. They represent an operator validation of the information. The red crosses in the *Related Link* field represent the sources of information (news) that were not yet validated by a human operator. In the person's specific information field we can also find some extra information: the completeness of the profile (in this case, there are fields missing so the profile is incomplete), a time-stamp for the last modification event, the certainty level since the person was already validated (in this case

System Implementation

Figure 4.25: OntoExtract *Person's Profile* page, in this case of "Bill Gates"

it is 100%) and the frequency this person appeared on the news (in this case 21 times). We also have an *Other Names* field where all nicknames associated with the person in question are shown. In this case no nickname was associated to "Bill Gates", but in Figure 4.26 we can see "Teixeira dos Santos" profile, where we can see that the name "Teixeira dos Santos" was recognized as a possible other name that refers to the same person in the Other Names section.

As can be seen in both profile pages there is also a *References* field where all references used in the process of completing a profile appear. The field for this experimental validation is dedicated to human operators. Also in the person's profile, there is a *Ergonyms / Roles* field where we can see all the roles and respective organizations that were found for the specific person. All roles and organizations are clickable, allowing the user to browse to their profiles. In this field, we can also see the certainty and number of hits associated to each role.

The button "Show" for each role, nickname and person profile (next to the main name on top) shows a pop-up window where the "history" of the discovery process is shown. This pop-up window shows a graph, generated on demand, that illustrates which names were discovered from the current name, and from which name the current name was found. To implement this feature we used the Perl's GraphViz extension. The Figure 4.27 shows us some examples of this pop-up window.

System Implementation

OntoEXTRACT > PERSON > TEIXEIRA DOS SANTOS

Sex: Masculino

Abstract: Fernando Teixeira dos Santos é o actual Ministro das Finanças (desde Outubro de 2009)

Completeness: Perfil Incompleto

Last Modified: 9 day(s) ago

Certainty: 100%

Frequency: 889 Hits

Other Names - (6-6 of 6)

Name	%	Hits	History
Teixeira dos Santos	88	1	Show

Ergonyms / Roles - (1-5 of 11)

Organization	%	Hits	History
ministro alemão das Finanças	88	1	Show
ministro belga das Finanças	88	1	Show
ministro da Economia	95	1	Show
ministro da Economia e Inovação	95	2	Show
ministro da Economia e Inovação	93	6	Show

Related Link - (1-5 of 33)

Link	Accessed
http://dn.sapo.pt/inf/...?content_id=1518510	10 day(s)
http://www.informacao...	10 day(s)

References - (1-0 of 0)

Figure 4.26: OntoExtract *Person's Profile* page, in this case of "Teixeira dos Santos"

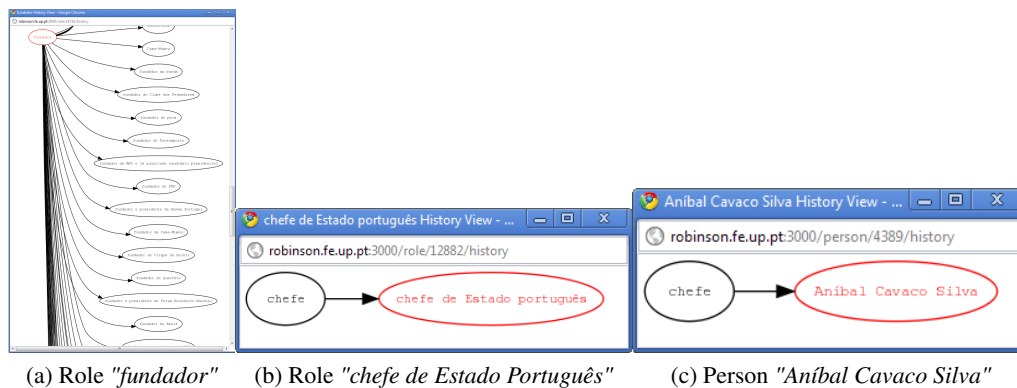


Figure 4.27: OntoExtract's pop-up windows for some history graphs

When the logged user is of type *Administrator* or *Operator*, some extra features become available. In the entities profile page on top, on the right hand of the name, we can see two new buttons: *Edit* and *Delete*. On top of the profile, some new options become available when clicking the *Edit* button, as can be seen in Figure 4.28.

The screenshot shows the 'OntoExtract > PERSON > AARON SORKIN' profile page. At the top, there are buttons for 'Show History View', 'Edit', 'Delete', and 'Hide/Show Validation'. The main form is divided into two columns. The left column contains fields for 'name', 'sex', 'title', 'picture', 'country', 'address', 'birthday', 'deathday', 'criminalrecord', 'abstract', 'articlelink', 'tag', 'notes', and 'incompleteflag'. Each field has a corresponding 'Valid' checkbox. The right column contains sections for 'Add Other Names', 'Add Related URL/Reference', 'Edit Related URLs', 'Edit References', 'Bind Existing Unbinded Role', 'Create New Role Organization', and 'Unbind/Delete Role'. Each section has its own set of validation checkboxes and buttons. At the bottom right, there is a small photo of Aaron Sorkin. At the bottom left, there is a summary section showing 'Sex: Masculino', 'Completeness: Perfil Incompleto', and 'Last Modified: 9 day(s) ago'.

Figure 4.28: OntoExtract *Person's Profile* page example with edit options available

Many editing options are available in this *Edit View*. Note that every validation is performed in the respective validation boxes next to each information entry.

Another feature that was implemented in the validation process was a log system for debugging and tracking purposes. This log system allows the administrator to track every validation and when it was done. In Log File 4.2 we can see part of one of these logs.

```
[2011/01/15 - 17:44] - validation of Person: José Maria Aznar
[2011/01/15 - 17:44] - validation of Role: ex-ministro espanhol
[2011/01/15 - 17:47] - validation of Person: Vladimir Putin
[2011/01/15 - 17:47] - validation of Role: Presidente russo
[2011/01/15 - 17:51] - validation of Person: Bono
[2011/01/15 - 17:51] - validation of Role: vocalista dos U2
[2011/01/15 - 18:31] - validation of Person: Diego Armando Maradona
[2011/01/15 - 18:31] - validation of Role: seleccionador argentino de futebol
```

Log File 4.2: Operator User validations log file example

4.5 Implemented Features

In this section, the implemented features list referred in subsection 3.3 is analysed to get a general overview of the state of implementation of the main features.

Each Table from subsection 3.3 was replicated with an extra column for the implementation status.

Looking at the implementation status we can say that 30 of 39 features defined in subsection 3.3 were implemented. We developed all nine high priority features. 17 medium priority features were implemented, in 19. Finally, of 11 low priority features, 4 were totally implemented.

4.5.1 Automatic Information Extraction / Retrieval System

Regarding the implementation status of the main features, all were completely implemented as can be seen on Table 4.6. So *Person's*, *Role's* and *Organization's* Names are extracted, although some improvements could be done to better clean *Organization's* names.

ID	Feature	Priority	Implemented?
F1.1	Extraction of Person's Names	High	Yes
F1.2	Extraction of Role's Names	High	Yes
F1.3	Extraction of Organization's Names	Medium	Yes

Table 4.6: Information extraction feature list implementation status

Concerning the relation mapping features between the entities, referred in Table 4.7, only the two main cases were implemented. These allow OntoExtract to map *Persons* to *Roles* and *Organizations* to *Roles*.

ID	Feature	Priority	Implemented?
F2.1	Map Persons to Roles	High	Yes
F2.2	Map Organizations to Roles	Medium	Yes
F2.3	Map Persons to Persons	Low	No
F2.4	Map Organizations to Organizations	Low	No
F2.5	Map Roles to Roles	Low	No

Table 4.7: Relation mapping feature list implementation status

The Complementary information extraction / retrieval main features referred in Table 4.8 were generally implemented. Only obtaining the *Role's* time spans, defined with medium priority, was not implemented. Of the low priority features in this area, only identifying the *Person's* gender was implemented.

ID	Feature	Priority	Implemented?
F3.1	Obtain Person's photo	Medium	Yes
F3.2	Obtain Person's nicknames	Medium	Yes
F3.3	Obtain Person's abstract	Medium	Yes
F3.4	Obtain Person's article link	Medium	Yes
F3.5	Obtain Organization's logo	Medium	Yes
F3.6	Obtain other Organization's known names	Medium	Yes
F3.7	Obtain Organization's abstract	Medium	Yes
F3.8	Obtain Organization's article link	Medium	Yes
F3.9	Obtain Role's abstract	Medium	Yes
F3.10	Obtain Role's article link	Medium	Yes
F3.11	Obtain Role's time span	Medium	No
F3.12	Obtain Organization's headquarters address	Low	No
F3.13	Identify Organization's type (Tag HAREM)	Low	No
F3.14	Identify sex of Person	Low	Yes
F3.15	Obtain other relevant informations about each Entity	Low	Partial

Table 4.8: Complementary information extraction / retrieval feature list implementation status

4.5.2 Graphical User Interface

Regarding the implementation status of the OntoExtract's GUI entity lists, all were completely implemented as can be seen on Table 4.9. So *Person's*, *Role's* and *Organization's* Lists are available to any authenticated user through the GUI.

ID	Feature	Priority	Implemented?
F4.1	View List of Persons	High	Yes
F4.2	View List of Roles	High	Yes
F4.3	View List of Organizations	High	Yes

Table 4.9: View lists feature list implementation status

The entity profile viewing features were also all implemented. So it is possible to view any of the main entities profiles: *Person*, *Organization* and *Role* profiles, as shown on Table 4.10.

ID	Feature	Priority	Implemented?
F5.1	View Person's Profiles	High	Yes
F5.2	View Role's Profiles	High	Yes
F5.3	View Organization's Profiles	High	Yes

Table 4.10: View specific profiles feature list implementation status

Edition features on all types of entities were also totally implemented as can be seen on Table 4.11. Editing Persons, Roles or Organizations is hence allowed by the OE GUI System.

System Implementation

ID	Feature	Priority	Implemented?
F6.1	Edit Persons	Medium	Yes
F6.2	Edit Roles	Medium	Yes
F6.3	Edit Organizations	Medium	Yes

Table 4.11: Edit specific entities feature list implementation status

OE's Authentication system is also fully operational and with role types for users. As can be seen on Table 4.12, all User main defined features were fully implemented.

ID	Feature	Priority	Implemented?
F7.1	Login System	Medium	Yes
F7.2	View list of Users	Low	Yes
F7.3	View User's profile	Low	Yes
F7.4	Edit Users	Low	Yes

Table 4.12: Authentication system feature list implementation status

Finally, of the extra viewing defined features only the entity's certainty graph viewing was implemented, as can be seen on Table 4.13

ID	Feature	Priority	Implemented?
F8.1	View entity certainty graph	Medium	Yes
F8.2	AIES control page	Medium	No
F8.3	View entity relations graph	Low	No

Table 4.13: Graph view feature list implementation status

Many extra features were implemented like the Log System. The main features described on the System Specification chapter 3.3 were improved during the system's implementation with some extra features.

4.6 Summary

This chapter presented the system's implementation of the proposed architecture for the Automatic Collection of Biographical Data on the Web for Public Personalities System referred in the previous chapter. This new system's name is OntoExtract.

A description of the main implemented features, focusing on the algorithms and heuristics used to solve all biographical entity extraction and retrieval problems is done. Finally, the list of main features defined in the previous chapter is completed with information related to the completeness of each feature.

Summarizing, of the thirty-nine main features defined in subsection 3.3 thirty were implemented. In a total of nine high priority features, all were developed. Seventeen

System Implementation

medium priority features were implemented of nineteen. Finally, of eleven low priority features only four were totally implemented.

System Implementation

Chapter 5

Tests and Results

This chapter presents an evaluation of the results achieved by the implemented prototype according to previously defined system specification, which was installed and configured in a test server running on a computer at LIACC (Artificial Intelligence and Computer Science Laboratory). In order to evaluate the OntoExtract project, two types of tests were simultaneously performed. Each tester validated the information automatically gathered through the AIES as an operator user. The tester is also a logged operator user and can assess the usability of the graphical user interface.

5.1 System Module Evaluation

In this section a description of all considered module evaluation techniques is done. In order to evaluate OntoExtract, we sent several invitations to students at *Faculdade de Engenharia da Universidade do Porto*. Thirty beta testers accepted to participate in a test at *Sapo labs*. A login as an operator user was created for each tester. Each had to validate twenty persons, ten persons from the *Lusophone* community and ten foreign persons, using the graphical user interface.

For the purpose of these tests, OntoExtract's DB was cleaned and the AIES was set to begin acquiring information of *Persons / Roles / Organizations* from the inputs of one hundred and seventy-two basic *roles* and two hundred and fifty-four *ergonyms*. Examples of some *roles / ergonyms* used can be seen on Table 5.1.

To evaluate the validity of the information, five evaluation parameters were considered for each tested *person*:

- **Name** - Verify if a person's name is correct or if some spelling error exists;
- **Gender** - Verify if the person's gender was correctly determined;

Ergonyms	Roles
actor	adjunto
advogado	administrador
agrônomo	almirante
apresentador	autarca
argumentista	bastonário
arquiteto	bispo
arquivista	cabeça-de-lista
artista	candidato
astronauta	cardeal-patriarca
astrônomo	chanceler
atleta	chefe
autor	comandante
bailarino	comissário
bombeiro	consultor
caixoteiro	convidado
cantor	coordenador
carpinteiro	correspondente
(...)	(...)

Table 5.1: Examples of *Roles* and *Ergonyms* used for finding *Persons* / *Roles* / *Organizations* with the defined news patterns

- **Photo** - See if the person's profile picture that was retrieved from DBpedia corresponds to the person in analysis;
- **Role / Ergonym** - Analyse only the role that has the biggest number of hits and verify if in fact the person ever held the position in question;
- **Organization** - In case the role / ergonym analysed has an organization associated, verify if it is a valid one.

To evaluate the usability of OntoExtract's graphical user interface, each beta tester had to answer to a series of opinion questions that comply with the *System Usability Scale* (SUS) [Bro96]. This is a simple ten-item *Likert Scale* [Mar07] giving a global view of individual assessments of usability. SUS yields a single number representing a composite measure of the overall usability of the system being studied.

The complete survey is divided in four sections (Available in Appendix A):

- **User Identification** - Several questions about gender, name and age;
- **Usability Questions** - To assess the usability of the graphical user interface the SUS was used, all ten SUS questions were translated;

- **Validation Results** - In this section the beta testers had to introduce the aggregated data of their validations (for instance how many Portuguese persons they had validated);
- **Suggestions** - For extra evaluation, some optional fields for suggestions and extra comments were added.

This survey was to be taken after the validation of all required persons, and thus already having thoroughly used the interface.

Each tester also had a paper and a pen where they had a table that they had to fill with the validated names. For each person they had a column, that they should tick if correct, for correct *name*, correct *gender*, valid *photograph*, valid *role / ergonym* (with greater number of hits), correct *organization* associated to the referred *role / ergonym*. This was useful for handling the differences between system and real validations, and for an easier counting of aggregated information submitted to the final survey.

5.2 Results

For the purpose of these validation tests the OE's DB was cleaned and the AIES script run with a base set of *ergonyms* and *roles*. No second iteration was done, meaning that the names found with this initial set of *roles / ergonyms* were not used for further search in the news for other names. The total number of found entities with this initial set of *roles / ergonyms* can be consulted in Table 5.2.

Entities	Number of Elements
Persons	6562
Organizations	3872
Total Roles	17775
Different Roles	9496

Table 5.2: Number of entities found in the start *roles / ergonyms*

As can be seen, the entity *Role* has been separated into two categories, the *Different Roles* and the *Total Roles*. This is because several *persons* have the same *role* or at least have a *role* with the same name causing some *roles* to be duplicated.

The 30 testers that did the test had an average age of 21 and most were men. The older tester had 27 years and the younger 18. In 30 testers 26 were men and 4 were women. This means that 87% of the testers were men and only 13% women. In Figure 5.1 we can see some moments of the tests.

In this section all acquired information from the tests performed is shown.



Figure 5.1: OntoExtract's testing moments at *Sapo Labs*

5.2.1 Graphical User Interface

We used the SUS scale to evaluate the GUI. In Figure 5.2 we can see the answers to the ten SUS questions of usability.

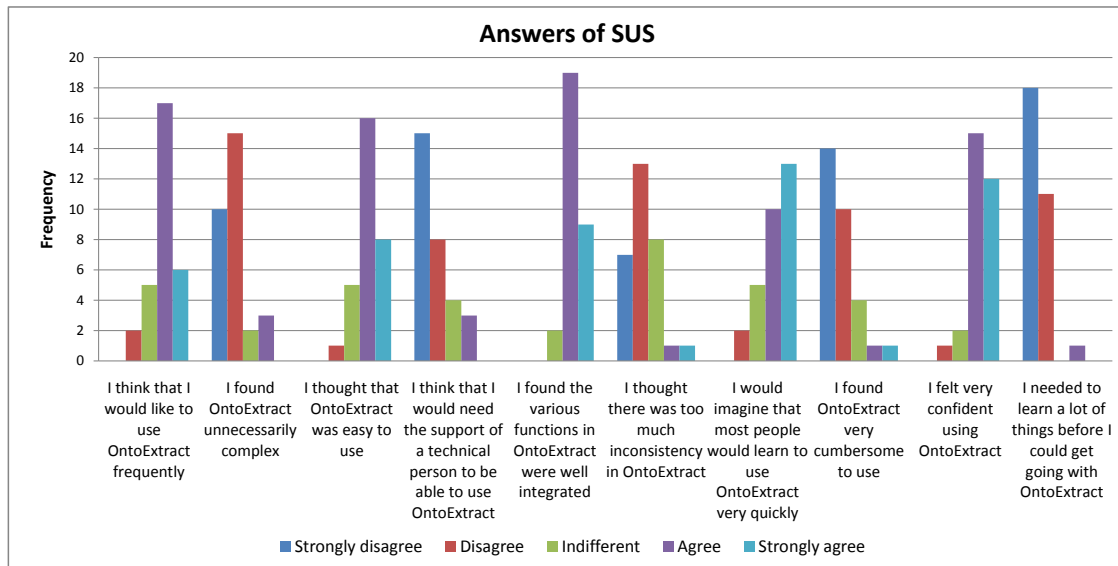


Figure 5.2: Distribution of answers to the SUS questions

The SUS scale score is a measure that can be easily interpreted. The value varies from 0 to 100 and if the value is near 100, it means that the individual considered the instrument extremely effective, efficient and satisfactory. In this experiment the result of the total score has a mean of 78.25 and a standard deviation of 12.72. The median is 80.0, the mode 82.5, the minimum is 47.5 and the maximum is 100.

5.2.1.1 Relevant Values Analysis

The fact that the mean value of the SUS test for usability is of 78.25 and the SUS scale is between 0 and 100, means that the interface has achieved its purpose. The Usability level

Tests and Results

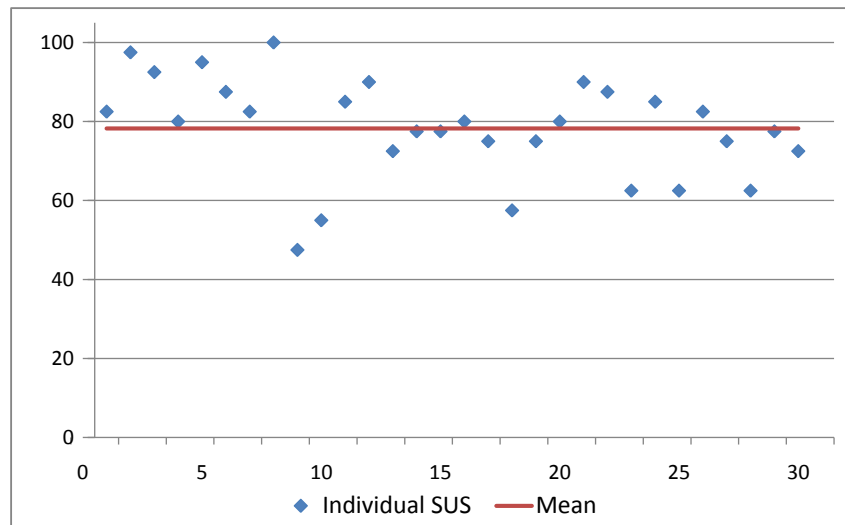


Figure 5.3: SUS results for each tester

of 78.25 percent is very acceptable. To calculate the SUS score, first we have to sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For questions 1,3,5,7 and 9 the score contribution is the scale position minus 1. For questions 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiplying the sum of the scores by 2.5 we obtain the overall value of System Usability.

To better analyse the SUS results, instead of considering all the answers we removed the three best and the three worst responses and verified that the mean was almost the same. This way most outliers were discarded. In Figure 5.4 we have the worst SUS evaluation after the removal of the outliers with an overall value of 62.5 in 100. In this Figure we can see each item's response values that were taken in consideration in the calculation of the overall SUS value.

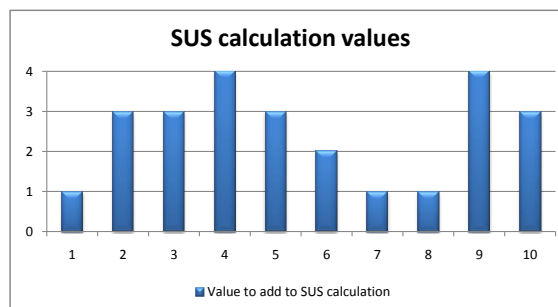


Figure 5.4: Worst SUS evaluation values for each question

Three questions were rated the worst: question 1 (*"I think that I would like to use OntoExtract frequently"*), question 7 (*"I would imagine that most people would learn to use OntoExtract very quickly"*) and question 8 (*"I found OntoExtract very cumbersome"*).

to use"). In fact, the act of validating a person's entry involved a large set of repetitive actions, turning the interface a bit boring to use after a while. This fact also affect, the will to frequently use OE and the difficulty of using OE. The more complex and boring the validation process is, the more difficult it is to teach people in a quick way to use OE. This is a point where the Graphical User Interface could be greatly improved. This analysis was only of one SUS response so may change for each tester.

5.2.2 Information extraction

To evaluate the information extraction system, AIES, each beta tester, had to validate 20 persons: 10 persons from the *Lusophone* community and 10 foreign persons. For these tests 30 beta testers attended, making a total of 600 person validations. Since the total number of found persons in OntoExtract is 6562, the number of tester validations represent a percentage of approximately 9.14% of the total number of persons in the system.

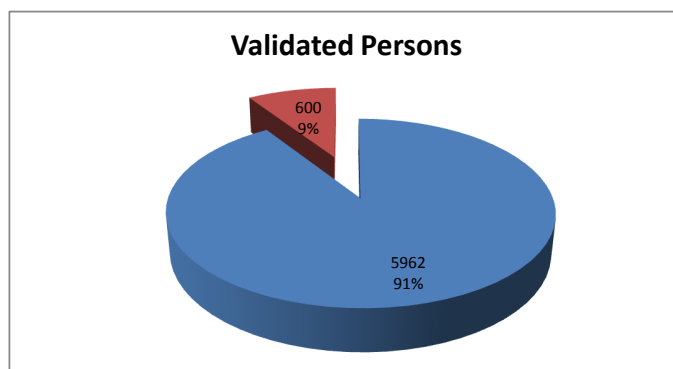


Figure 5.5: Percentage of validated *Persons* in OntoExtract after the tests

As can be seen in Figure 5.5 the number of validated *Persons* in OntoExtract after the tests, despite being a large number of validations, is still a minor part of the whole system. Nonetheless, it is a good representation of the global system.

The aggregated responses of all testers can be analysed in Table 5.3. In this table a general count of all parameters evaluated for each Person is displayed.

5.2.2.1 Relevant Values Analysis

There were significant changes between the *Portuguese Persons* validations and *Foreign Persons* validations, mainly in captured *Photos* and *Organization* identification. Also a small difference between correct *roles / ergonyms* found exists, but not significant.

The difference in Correct Photos found between the Portuguese Persons and the foreign is of about 97 and in correct Organizations found of 52. The difference is really

	Portuguese	Foreign	Total
Number of Evaluated Personalities	300	300	600
Correct Names	296	294	590
Correct Gender	293	293	586
Correct Photo	9	106	115
Role / Ergonym Correct	248	260	508
Correct Organization	131	183	314

Table 5.3: Aggregated Total Validation Results

significant. The difference in the number of acquired correct photos, or in most cases, acquiring photos at all is mainly due to some retrieval limitations the DBpedia dumps have. Further explanation is done in subsection 5.2.2.2.

The really intriguing difference is between the number of correct *Organizations* detected, because we have found better results for *foreign persons* and the sources of all acquired *Organization's names* are news written in Portuguese. This might be due to the fact that when Portuguese journalists refer to a foreign *Person*, it is because he is already well-known. Since the personalities are already well-known, probably the *Organization* they belong to is also well-known or is the reason they are so well-known. On the other hand they are rarely referred and are not that "familiar" to Portuguese people, making it uncommon of them to have nicknames or the same level of detail and amount of information about them on the news. Although because of this, less information about them is going to be available, it also allows for a more precise extraction because the information available is actually more concise.

Nonetheless, the main objective of OntoExtract is to obtain the *roles / ergonyms* associated to the *Person* they belong to, and in that area, OE has very good results. In 600 studied cases 508 *roles / ergonyms* were correct. The difference between the Portuguese and the foreign persons role identification is small and insignificant. Both attain a good extraction level as can be seen on Figure 5.6.

It is interesting to also note that the gender identification through the news has a great level of accuracy, in 600 person validations 586 had their gender correctly identified.

The name cleaning functions also worked very well in the *Person's name* cleaning, because almost none of the names were incorrect or with extra information. In 600 names 590 are correct and without any extra nouns or verbs.

To better understand the data in Table 5.3 please analyse Figure 5.6 where *Portuguese* persons validation and *Foreign* persons validations are directly compared.

5.2.2.2 Error Analysis

As noted in the last subsection: there is a very big difference between the number of photos found for *Portuguese persons* and the number of photos found for *foreign persons*.

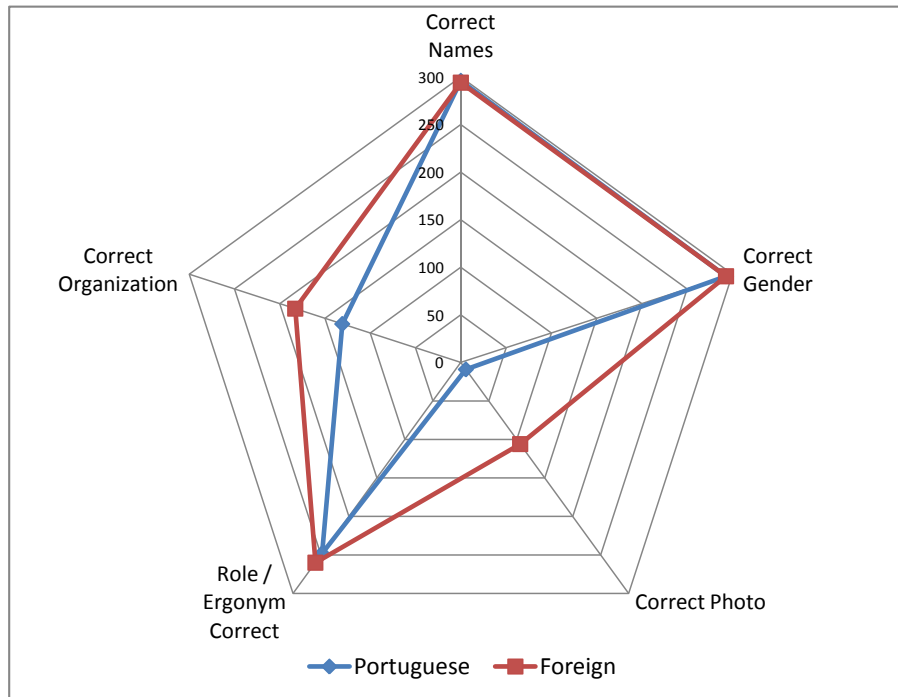


Figure 5.6: Graph comparing *Portuguese* and *Foreign* persons validations

This is mainly due to the fact that the DBpedia dump used for the retrieval of the photos is in English because there is no DBpedia dump for photos in Portuguese presently. The fact that the dump is in English brings many obvious disadvantages, one of which is that the Portuguese entities have their names normally translated to English and in some cases this disables the match with the name found and the whole process of photo retrieval ends without success.

Another problem is that, since the DBpedia dumps are intended to be downloaded, and need to stay relatively small, the dumps are normally of a small portion of the Wikipedia, and only the most important and relevant articles are taken in consideration for dumping. Unfortunately, the relevant articles for the English Wikipedia normally are not the same for the Portuguese Wikipedia and the real important Portuguese entities are not considered, and thus their photo retrieval through the DBpedia dump becomes difficult.

5.3 Summary

This chapter presented an analysis of the results obtained through the experimental tests.

These tests were made on *Sapo Labs* and were based on human validation of found Persons on the news. For them thirty testers complied to the invitation and actively participated in the validation process. Each one had to validate 20 persons, 10 Portuguese and 10 foreign.

Tests and Results

To evaluate the system's usability the set of SUS's ten questions was used.

Tests and Results

Chapter 6

Conclusions and Future Work

6.1 Work Summary

The main goal of this dissertation was to specify and develop a prototype of the system that could collect biographical data on the web for public personalities in an semi-automatic way. This system, besides being used for consulting and searching purposes, it can also be used for giving support to intelligent search engines, support journalistic tools that identify people in the news, support citation analysis tools like *Sapo Voxx*, and so on. In order to achieve the proposed goals we performed a study of similar systems implementing related features, like *Freebase* [fre] and *KnowItAll* [ECD⁺04]. A comprehensive study of many other relation extraction systems was needed in order to be able to understand the problem and discover viable solutions. The main motivation for this dissertation project is the fact that we do not have the community nor the resources projects that, for example, *Freebase* has, and so, to implement a similar system for the Portuguese language, a semi-automatic system was necessary because a totally manual system is unviable due to the lack of possible users.

Following the goals definition and the study of the state of the art, the specification of the new system, *OntoExtract*, took place. The new system has three main components: a database storing all relations and entity information; an automatic information extraction system able to extract new relations along with entity names and their binding role; and a graphical user interface allowing the manual validation of information by an human operator. Additionally, a biographical information retrieval system is required to acquire extra information like article abstracts from wikipedia, article links and photos. The global architecture of *OntoExtract* was defined taking in consideration all these features.

Having the complete system specification, a prototype was developed with codename *OntoExtract*. Most of the main specified features were implemented, specifically 30 of

39. All 9 high priority features were developed. 17 medium priority features were implemented, of 19. Finally, four of eleven low priority features were implemented.

Several tests took place to evaluate OntoExtract. For these tests 30 persons volunteered and actively participated in them. Each volunteer had to act as an operator and validate 20 persons: 10 of the Lusophone community countries and 10 of other foreign countries. Each test included the validation of a *Person*'s name, gender and acquired photo, as well as the associated *Role / Ergonym* that occurred more times in the news. In case the *Role* had an *Organization*, the operator had to assess if it was a valid *Organization*. The results of these tests revealed that the *Person*'s names associated with their *Roles / Ergonyms* are extracted from the news with good accuracy. Regarding the *Organization*'s names, the extraction process could be improved but generally they are acquired correctly when discovered. The person's gender identification is also performed with very good accuracy. The photos are retrieved with higher efficiency for foreign personalities; generally they correspond to the Person in question when retrieved. The retrieval results of this project depend greatly on the sources, in this case DBpedia, as explained in chapter 5.

To conclude, OntoExtract implements most of the proposed features and stands as a functional prototype that validates the proposed solution for the problem raised: To be able to, in a semi-automatic way, obtain biographical information of public entities.

6.2 Future Work

Although we successfully developed a complete system specification and a functional prototype that implements most of the proposed features, there are still further steps that need to be followed in order to obtain a more robust system:

- Provide OntoExtract with extraction and retrieval functions capable of understanding the time span of *Roles / Ergonyms*;
- Provide OntoExtract with extraction and retrieval functions that allow it to acquire birthdays and dates of death for each *Person*;
- Provide OntoExtract with extraction and retrieval functions that allow it to acquire foundation day and end day for each *Organization*;
- Add extra news sources or even allow the system to automatically find news in Portuguese on the WWW to allow the crawling of other available sources of information;
- Improve the GUI for better user experience;

Conclusions and Future Work

- Create an extra component / module that could serve as an web-service to allow other applications to easily query for informations;
- Improve *Organization* and *Role* name cleaning functions;
- Improve *Organization* and *Role* nickname identifying functions;
- Create more information retrieval functions capable of obtaining extra information about each entity.

Conclusions and Future Work

References

- [AHB⁺95] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, David Martin, Karen Myers, and Mabry Tyson. Sri international fastus system: Muc-6 test results and analysis. In *Proceedings of the 6th conference on Message understanding*, MUC6 '95, pages 237–248, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.
- [APAL10] M.Olga Azeredo, M.Isabel Pinto, and Maria do Carmo Azeredo Lopes. *Da Comunicação à Expressão*. Lisboa Editora, 1st ed. edition, 2010.
- [Bal07] Krisztian Balog. People search in the enterprise. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 916–916, New York, NY, USA, 2007. ACM.
- [BCT07] K. Bollacker, R. Cook, and P. Tufts. Freebase: A shared database of structured general human knowledge. In *Proceedings of the national conference on Artificial Intelligence*, volume 22, page 1962. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.
- [Bro96] J. Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.
- [BRSS08] A. Branco, L. Rodrigues, J. Silva, and S. Silveira. XisQuê: an online QA service for portuguese. *Computational Processing of the Portuguese Language*, page 232–235, 2008.
- [cat] Catalyst - web framework. <http://www.catalystframework.org/>.
- [CCH11] Xian-Yi Cheng, Xiao-hong Chen, and Jin Hua. The overview of entity relation extraction methods. In Ran Chen, editor, *Intelligent Computing and Information Science*, volume 134 of *Communications in Computer and Information Science*, pages 749–754. Springer Berlin Heidelberg, 2011.
- [dbp] wiki.dbpedia.org : About. <http://dbpedia.org/About>.
- [dja] Django | the web framework for perfectionists with deadlines. <http://www.djangoproject.com/>.
- [ECD⁺04] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander

REFERENCES

- Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 100–110, New York, NY, USA, 2004. ACM.
- [fre] What is freebase? - freebase. http://wiki.freebase.com/wiki/What_is_Freebase%3F.
- [FS] R. Feldman and J. Sanger. The text mining handbook: advanced approaches in analysing unstructured data. *Computational Linguistics*, 34(1).
- [Jac00] Robert Jacobson. *Information design*. MIT Press, Cambridge Mass., 1st MIT press paperback ed. edition, 2000.
- [kev] Kevin's word list page. <http://wordlist.sourceforge.net/>.
- [lev] Levenshtein distance - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Levenshtein_distance.
- [Mar07] J. Maroco. *Análise Estatística – com utilização do SPSS*. Edições Sílabo, Lda, 2007.
- [May06] M.T. Maybury. Expert finding systems. *Bedford, Massachusetts, The MITRE Corporation*, 2006.
- [MS08] C. Mota and D. Santos. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca, 2008.
- [nlx] NLX-Group. <http://nlx.di.fc.ul.pt/>.
- [P⁺09] A. Picón et al. Web people search. 2009.
- [per] The perl programming language - www.perl.org. <http://www.perl.org/>.
- [pyt] Python programming language – official website. <http://www.python.org/>.
- [RY02] Dan Roth and Wen-tau Yih. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [Sar06] L. Sarmento. Hunting answers with RAPOSA (FOX). In *Working Notes for the CLEF 2006 Workshop*, 2006.
- [SB04] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *Proceedings of HLT-NAACL*, volume 5764, 2004.
- [SCR03] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov models for information extraction. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 427–433. Citeseer, 2003.
- [SWPS07] M. Schedl, G. Widmer, T. Pohle, and K. Seyerlehner. Web-Based detection of music band members and Line-Up. In *Proceedings of the International Conference on Music Information Retrieval*, page 117–8, 2007.
- [ver] Verbatim - saber mais. <http://verbatim.labs.sapo.pt/sobre>.

REFERENCES

- [Wol] About Wolfram|Alpha: making the world's knowledge computable. <http://www.wolframalpha.com/about.html>.
- [YGTH00] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the sixth conference on Applied natural language processing*, ANLC '00, pages 282–289, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
- [ZG05] S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 419–426. Association for Computational Linguistics, 2005.
- [Zha07] J.-Z. Zhao. *A Research for Semantic Relation Automatic Extraction Among Named Entities in Chinese Professional Domain*. Huazhong Normal University, Shanghai, 2007.

REFERENCES

Appendix A

Survey

OntoExtract - Questionário

O OntoExtract é uma aplicação que pretende extrair informação sobre entidades públicas e as suas relações de forma semi-automática.

Este inquérito pretende averiguar o grau de satisfação da utilização do OntoExtract bem como a usabilidade tendo em conta o contexto e a eficiência de cumprimento de tarefas na Interface.

Adicionalmente pretende-se verificar a veracidade da informação contida na aplicação que foi extraída automaticamente de notícias.

Finalmente tenta-se averiguar quais as possíveis melhorias passíveis de serem implementadas no sistema.

* Required

Nome *

Idade *

Sexo *

☐ Masculino

☐ Feminino

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Figure A.1: Survey's first page

OntoExtract - Questionário

*** Required**

Usabilidade

Nesta secção tenta-se avaliar a usabilidade do sistema com o método SUS (System Usability Scale)

1 . Penso que gostaria de utilizar a interface do OntoExtract com frequência. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

2. Penso que a interface OntoExtract é desnecessariamente complexa. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

3. Penso que a interface OntoExtract é fácil de utilizar. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

4. Penso que iria necessitar de apoio de um técnico para ser capaz de utilizar a interface OntoExtract em pleno. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

Figure A.2: Survey's SUS questions page one

Survey

5. Penso que as diversas funcionalidades da interface OntoExtract foram bem integradas. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

6. Penso que a interface OntoExtract tem muitas inconsistências. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

7. Imagino que a maioria dos utilizadores aprenderia facilmente a utilizar a interface OntoExtract. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

8. Penso que a interface OntoExtract é muito cansativa de utilizar. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

9. Senti-me confiante na utilização da interface OntoExtract. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

10. Precisei de aprender muitas funcionalidades antes de conseguir utilizar a interface OntoExtract. *

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

Powered by [Google Docs](#)

Figure A.3: Survey's SUS questions page two

OntoExtract - Questionário

* Required

Resultados de Utilização
Nesta secção devem ser introduzidos todos os dados adquiridos de forma agregada.

Personalidades Portuguesas
Para os seguintes campos, diga o número de resultados bem capturados

Total de Personalidades Portuguesas Avaliadas: *

Total de Nomes correctos: *

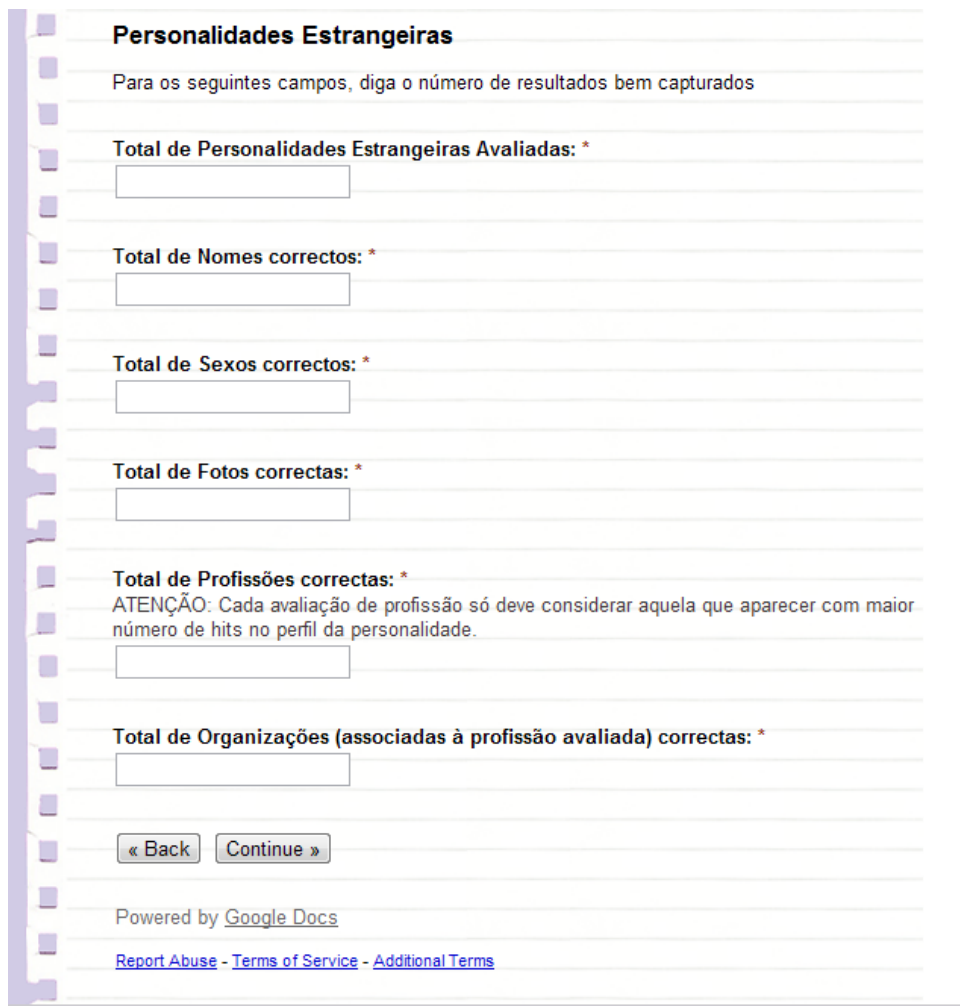
Total de Sexos correctos: *

Total de Fotos correctas: *

Total de Profissões correctas: *
ATENÇÃO: Cada avaliação de profissão só deve considerar aquela que aparecer com maior número de hits no perfil da personalidade.

Total de Organizações (associadas à profissão avaliada) correctas: *

Figure A.4: Survey's validation information for Portuguese *Persons*



Personalidades Estrangeiras

Para os seguintes campos, diga o número de resultados bem capturados

Total de Personalidades Estrangeiras Avaliadas: *

Total de Nomes correctos: *

Total de Sexos correctos: *

Total de Fotos correctas: *

Total de Profissões correctas: *
ATENÇÃO: Cada avaliação de profissão só deve considerar aquela que aparecer com maior número de hits no perfil da personalidade.

Total de Organizações (associadas à profissão avaliada) correctas: *

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Figure A.5: Survey's validation information for foreign *Persons*

OntoExtract - Questionário

Sugestões
Finalmente, nesta secção deve sugerir o que poderia ser melhorado no projecto OntoExtract.

Sugestões de melhorias:
Opcional

Comentários adicionais:
Opcional

[« Back](#) [Submit](#)

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Figure A.6: Survey's additional suggestions and comments