Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Legal Entity Extraction with NER Systems

Master Thesis

Master In Artificial Intelligence

Author: Ines Badji, inesbadji@gmail.com
Advisor(s): Óscar Corcho and Víctor Rodríguez-Doncel

https://github.com/ibadji/Legal-NER

June, 2018

Acknowledgments

I am very grateful to have been given the opportunity to work with the Ontological Engineering Lab and would like to thank all its members for the warmth, kindness and support they showed me. A particular thanks to my advisors Óscar Corcho, Víctor Rodríguez-Doncel and Elena Montiel for helping me with this Master Thesis. It was a pleasure collaborating with you and I hope to have the chance to work with you in the future.

Abstract

Named Entity Recognition over texts belonging to the legal domain focuses on categories (*legal entities*) like references to specific laws, judgments, name of courts or stages in a legal process. Although there is a rich choice of libraries for implementing NER systems, these late ones are not domain specific and do not work well on text pertaining to the Legal domain. Similarly, little focus is given to Spanish since most research is done on the English language.

The objective of the work presented in this thesis is the identification of legal entities in Spanish and English texts, with a main focus on informal references to legislative documents found in news, Twitter, contracts or journal articles. The work is framed in the H2020 Lynx project, aimed at creating a Legal Knowledge Graph enabling the provision of compliance-related services.
A Rule Based approach can be used to recognize references to norms in Spanish and English documents belonging to the legal domain applied on top of a combination of Natural Language Processing Tools. To recognize the mentions in documents of a less formal nature, a number of vulgar variants for the names of the public acts or judgments is necessary. By querying on Wikidata, DBpedia and BOE a table of synonyms is produced. These resources have been published along with a small annotated data set taken as gold standard.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Lawyers and professionals of the Legal field find extremely valuable having references to legal entities identified and highlighted in legal information systems, possibly with hyperlinks to source documents. Machines on the other hand, make an even more important use of these references, using them to improve search algorithms, anonymize documents, make data analysis, summarize long texts or provide recommendation services.

Named-entity recognition (NER) is a Natural Language Processing task aimed at identifying references to specific entities in a text. Those entities can be of different types such as persons, organizations, places, dates, quantities, monetary values or percentages. In the legal domain, references to other entities are also of interest, such as norms, judgments, courts or jurisdictions. These entities are hereinafter referred as *legal entities*.

For example, in the financial aid information document for postdoctoral studies in UPM [19]the following text is stated:

> Las ayudas objeto de esta convocatoria estan sometidas a las Bases Reguladoras contenidas en la Resolución Rectoral de 23 de febrero de 2017 para la concesión de ayudas del programa propio de I+D+i de la UPM, a la Ley 39/2015, de 1 de octubre, del Procedimiento Administrativo Común de las Administraciones Publicas, la Ley 35/2006, de 28 de noviembre, del Impuesto sobre la Renta de las Personas Fisicas y de modificación parcial de las leyes de los Impuestos sobre Sociedades, sobre la Renta de no Residentes y sobre el Patrimonio, la Ley 3 8/2003, de 17 de noviembre, General de Subvenciones, Real Decreto 462/2002, de 24 de mayo, sobre indemnizaciones por razón del servicio así como la Resolución de 9 de febrero 2018 por la que se dictan instrucciones sobre comisiones de servicio con derecho a indemnización, y a los Estatutos de la UPM, aprobados por Decreto 74/2010, de 21 de octubre (BOCM del 15 de noviembre).

Fig. 1: Sample text with references to norms.

In the text above, references to different norms have been identified: three laws (*leyes*, in red colour), one royal decree (*Real Decreto*, in magenta), one *decree of the Autonomous Community of Madrid* (in orange), one statute (in pink) and two internal UPM dispositions (*Resoluciones Rectorales*, in brown). In addition, other legal entities might be found, such as institutions (UPM), official publications BOCM), etc.

Tab. 1: Example of Legal Entity extraction

Detected Entity	Entity Category
Ley 39/2015	Ley Organica
Ley 39/2006	Ley Organica
Ley 38/2003	Ley Organica
Real Decreto 462/2002	Real decreto
Resolution de 9 de febrero 2018	Resolution
Decreto 74/2010	Decreto
BOCM	Abbreviation

The excerpt above comes from a formal document, and references to norms are expected to appear in their full form (*Ley 3 8/2003, de 17 de noviembre, General de Subvenciones*) with little variations (capital letters, commas or slashes). References in other less formal contexts can vary greatly. As an example, Fig. 2 shows the tweet of a famous lawyer making reference to two laws, *Ley Sinde* (Ley 2/2011, de 4 de marzo, de Economía Sostenible) and *Ley Mordaza* (Ley Orgánica 4/2015, de 30 de marzo, de protección de la seguridad ciudadana). To show the complexity, one may consider that Ley Sinde is sometimes referred to as Ley 2/2011, Ley Sinde-Wert, Ley de Economía Sostenible or simply LES.

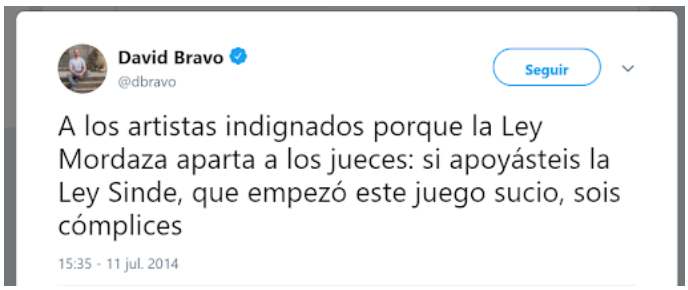

Fig. 2: Twitter Example

Much software has been developed for the purpose of named entity recognition covering different languages. The most used and known in the field are the Stanford NER[1] or CoreNLP, the OpenNLP[2], GATE[3] or IxaPipes[4]. An extensive list can be checked in *Section 2.2.5 (Tools)*.

These NLP libraries are good at finding general named entities common to all fields, such as Locations or Person. However, in order to detect legal entities, domain specific training is needed. Further more, some filtering has to be taken in consideration as the reference to norms (like Ley 2/2011) outputs in NLP libraries references to dates (problem addressed by Navas [21]). However, three difficulties hinder a fair evaluation after training these tools: first, the vague definition of legal

[1] `https://stanfordnlp.github.io/CoreNLP/`
[2] https://opennlp.apache.org/
[3] https://gate.ac.uk/
[4] http://ixa2.si.ehu.es/ixa-pipes/

entity; second, the performance of the tools strongly depends on the training material and third, the scarcity of corpora with documents where legal entities have been annotated (there are no adequate gold standards). For that reason, efforts have being made in order to adapt those capabilities to other fields such as the medical domain or in the task at hand, the legal domain, using mainly two methods rule based and/or machine learning.

Currently, the state of the art is focused on already defined entities particularly for the English language, and little attention has been paid to Spanish. Thus, having such a system for legal references would have a clear added value in the domain. Annotation of legal entities could be either used directly as an easier way to find information or as a base for further research in the following fields, among others:

- Finding cases similarity for defense purposes
- Search engine for NER
- Anonymizing documents
- Automatic generation of contracts
- Summarization of documents
- Raking lawyers based on the number and type of cases solved

This system is implemented while having in mind any type of text, legal document as well as non formal documents in order to cover a wider range of needs and possibilities.

A *non-formal document in our case is seen as any text file that is not per say a document specifically belonging to the legal domain, and it can be anything from a Tweet to a comment on a social media platform.* Formal documents are those having any power, such as contracts, acts or judgments. Other documents lie in between such as journal articles (with no power but still using a more formal style) or official recommendations.

This work is done in the context of **H2020 Lynx project**[5]. The Lynx project aims at creating a Legal Knowledge Graph enabling the provision of compliance-related services, as stated in the website `http://lynx-project.eu`

> Lynx envisions an ecosystem of smart cloud services to better manage compliance documents. A one-stop shop for SMEs and companies operating internationally seeking legal information and knowledge-based services. Lynx will rely on a Legal Knowledge Graph of heterogeneous compliance data sources (legislation, case law, standards, industry norms and best practices) duly interlinked and integrated. [20].

[5] Lynx has received funding from the Horizon 2020 European Union (EU) Research and Innovation program under Grant Agreement: 780602

1.2 Objective and approach

The objective of this Master Thesis is **to design algorithms and implement services to extract *legal entities* in text from both formal and informal contexts**. Languages to be covered are Spanish (limited to the jurisdiction in Spain) and English (limited to the European law in EurLex[6]).

The approach to reach this objective combines the use of Natural Language Processing frameworks namely: CoreNLP, OpenNLP, IxaPipes, as GATE (Annie), Apache Lucene and Text Similarity algorithms such as FuzzyWuzzy. Another necessary part of the work is the generation of dictionaries and Regular Expressions/-Parsing Expression Grammar (PEG).

In order to obtain non-official terms (Nicknames or common names) to refer to the norms (For Example: *Ley de Economia Sostenible is known as Ley Sinde*), SPARQL queries on Wikidata[7], DBpedia[8] and BOE[9] were made. The result of these queries has been parsed and stored in an spreadsheet file (see Section 3.1.2 for more details).

In this project, a few features were required, flexibility and extendable architecture, stability and performance, maintainability, and finally a broad variety of entities to be detected.

Similarly to [16], three methods for NE recognition are used in this project:

- Look-up: Dictionaries or lists with the terms needed to be detected in the text.
- Pattern rules: Patterns (such as regular expressions) and rules (such as Jape rules) are a more flexible way of expressing the terms to be search, and higher accuracy can be obtained. However, this method requires the considerable effort of manually creating the rules and patterns.
- Statistical models: requires manually annotated documents for learning purposes.

Advantages of a system following this approach are:

- The system is maintainable, as SPARQL queries can be regularly launched to update dictionaries and regular expressions;
- The system is extendable, as it is easily portable to other languages and jurisdiction;
- The data is toold-agnostic, and the knowledge gathered can be used in different software flavors (e.g. CoreNLP etc.).

The entities to be detected are summarized in Figure 3 (for Spanish) and 4 (for English), where they have been hierarchically arranged.

[6] EurLex is the official website of the European Union to publish law and other public documents. It is available in the 24 official languages and it also publishes the Official Journal (OJ).

[7] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

[8] `http://dbpedia.org/page/Spain`

[9] `https://www.boe.es`

```
Spanish
        1- Service 1:
                Sentencia
                Articulo
                Constitution
                LeyOrganica
                Directiva
                Recurso
                Reglamento
                Decreto
                Orden
                Dictamen
                Apelacion

        2- Service 2:
                Nicknames

        3- Service 3:
                Abreviation (ABREVIATION, INSITUTION_ABR)
                Governemental_Institution (Gov_instit_regi, Ministeries, LAW_Entities)
                Location_Spain (Geographical Divisions, Regions, Cities)
                Detection_words
```

Fig. 3: Entities List that can be extracted for Spanish

```
English:
        1- Service 1:
                Treatie
                Agreement
                RegDirDec
                Judgment
                CaseLaw
                OfficialJ
                Directive
                Article
                Regulation
                Decision
                Law

        2- Service 2:
                Governemental_Institution
                Abreviation
                Language
```

Fig. 4: Entities List that can be extracted for English

In the above figures Fig. 3 and Fig. 4, we can see the different entities that can be detected by the system developed in both Spanish and English. The list was selected by virtue of the relevance of the entities and their interest for the Lynx project. As an example of an alternative, systematic choice, would have been considering the exhaustive list of types of documents published by BOE.

> Acuerdo, Acuerdo Internacional, Auto, Circular, Código Internacional, Comunicación, Constitución, Corrección (errores o erratas), Decisión, Declaración, Decreto, Decreto Foral, Decreto Foral, Legislativo, Decreto Legislativo, Decreto-ley, Decreto-ley Foral, Directiva, Edicto, Enmienda, Instrucción, Ley, Ley Foral, Ley Orgánica, Nota Diplomática, Orden, Orden Foral, Otros, Providencia, Real Decreto, Real Decreto Legislativo,

Real Decreto-ley, Recomendación, Reforma, Reglamento, Resolución, Sentencia

The following points give an idea of what some of those entities represent. This information can be found on the EurLex website[10].

- "Implementation laws *(Leyes orgánicas)*: laws implementing fundamental rights and public freedoms, approving the statutes of the autonomous communities, and implementing the general electoral system"
- "Laws *(Leyes)*: laws adopted by parliament in plenary session by a simple majority, not concerning matters governed by an implementation law"
- "Royal decree-laws *(Reales decretos-leyes)*: acts adopted by the government in exceptional circumstances and emergencies. The scope of these acts is restricted; they cannot relate to the functioning of the key State institutions, the rights, duties and freedoms of individual citizens, the autonomous community system, or general electoral law"
- "Regulations *(reglamentos)*: Regulations can be adopted by the central government, governments of the autonomous communities and the administration. Their function is to implement, develop or supplement laws. The most important regulations in the Spanish legal system are royal decrees, ministerial decrees *(órdenes ministeriales)*, resolutions, instructions and circulars."

1.3 Hypotheses

In the project at hand, a few hypotheses have been assumed:

H1. Most legal references are identifiable with rules (regular expressions, PEG...).

H2. Dictionaries / regular expressions / PEG can be automatically created from open resources (e.g. SPARQL queries to open data sources). For example, by fetching nicknames of laws, an always up-to-date system is possible where colloquial references to legal documents are understood.

H3. Combining machine learning methods (via standard implementations) and rule-based systems can lead to good performances.

H4. Formal documents are considered to reference laws in their full name with no typos.

H5. Non-Formal documents are considered to be typo prone.

1.4 Methodology

This section tackles the main steps followed in order to complete the project. It will cover the Models used, the corpus used, as well as the basic development steps carried out.

[10] `http://eur-lex.europa.eu/n-lex/info/info-es/index_en`

- Model:

  In order to recognize the structure of the law references, patterns were identified and created using Regex. In order to identify those patterns an extensive online search was carried out and the result of the patterns and information gathered can be seen in the Indexes Fig 14, Fig 15, Fig 16, Fig 17, Fig 18, Fig 19 as well as the Section 4 for more details about the construction and sources of those rules.

- Collecting resources:

  The English corpus was found on the Eurolex official website, 10 different text documents were gathered, 4 from Directives and 6 from EU court decisions. A bigger corpus was not necessary in our case since the chosen documents covered the entities that were targeted.
  Similarly for the Spanish Corpus, Lynx partners provided 10 sample text documents. In addition, 3 articles related to law were gathered as well as other texts from different origins namely, 4 documents from university decisions and scholarships proposals as well as 325 tweets.
  The focus was on the Spanish language thus the importance of different types of documents. Overall the project was run on a total of 28 text documents[11]. Those same documents were also manually annotated in order to test for the accuracy of the algorithms and rules used.

- Integrate and develop the system:
  - Identifying patterns
  - Develop RE and PEG
  - Parsing the documents (annotating)
  - Run on the different programs and algorithms (coreNLP, openNLP, GATE, Ixapipes, Apache Lucene...)
  - Creating a combination algorithm to pick the best results and avoid overlaps
  - Evaluation Algorithm
  - Find relations between entities (relation extractor)
  - Find way to relate entities to links
  - Visualizing of results (highlight on text)
  - Find list of nicknames for laws (Spanish DBpedia and Wikidata)
  - Exporting to NIF as an output format

- Publish it online as a web service.

[11] `https://github.com/ibadji/Legal-NER/tree/master/resources/inputText`

- Perform experimentation and evaluation: runs on the annotated documents were done, the results and accuracy of the test can be seen in the *Section 5*.

1.5 Terminology

1.5.1 Abbreviations

- BOE: Boletin Official del Estado
- CFG: Context Free Grammars:
- ML: Machine Learning
- NE: Named Entity
- NER: Named Entity recognition
- NLP: Natural Language Processing
- PEG: Parsing expression grammar
- Regex or RE: Regular Expression

1.5.2 Definitions

- Dictionaries: are lists of terms. The system tries to find NE in the dictionary for each word in order to mark it. This method is usually used in combination with other more complex systems. Dictionaries of NEs are often called gazetteers.
- Gazetteers: are lists of NEs. Systems with gazetteers are obviously losing the possibility of direct use for other languages.
- Law Entity: is defined as a detected word that references a legal information such as *High court of Justice or Ley 13/2018*.
- Non-formal document: in our case is seen as any text file that is not per say a legislative document such as Journal articles, contracts which however do have references to official documents or laws
- Regular Expressions: REs are a grammar classified as regular in Chomsky hierarchy. They can thus be processed by finite state automaton in a very short time.

2 State of the Art

2.1 Literature Overview

Named-entity recognition (NER) (also known as *entity identification and* entity extraction) is a sub-task of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, places, expressions of times, quantities, monetary values, percentages and more.
A lot of effort has been made in the field particularly on the English language, however most of the systems created focus mainly on non-specific domains rendering what were taught of as good results on common texts very inaccurate when more domain specific corpus is used.

Originally developed by computational linguists as a sub-task to information extraction, named-entity recognition quickly attracted the attention of researchers in various fields. The first research paper pertaining to NER was presented at the Seventh IEEE Conference on Artificial Intelligence Applications by Lisa F. Rau (1991), describing a system that extract and recognized [company] names, relying on heuristics and handcrafted rules.[17] The original concept of a 'named entity'(NE), was first introduced in MUC-6 (sixth in a series of Message Understanding Conferences. 1995), it covered names of people, organizations, and geographic locations as well as time, currency, and percentage expressions [3]. Since then, the interest never declined with steady research and numerous scientific events: HUB-4, MUC-7 and MET-2, IREX, CONLL, ACE and HAREM. The Language Resources and Evaluation Conference (LREC) has also been staging workshops and main conference tracks on the topic since 2000 [17].

A good portion of NER research is devoted to the study of English, due to its significance as a dominant language that is used internationally for communications, science, information technology, business, seafaring, aviation, entertainment, and diplomacy. NER can be defined as the task that attempts to locate, extract, and automatically classify named entities into predefined classes or types in open-domain and unstructured texts, such as newspaper articles.
The task of identifying named entities must be distinguished from entity tracking, which involves identifying mentions, relations, and the co-references that may exist between them.

Examples of applications for which NER is useful:

- Information Retrieval: Recognizing the NEs on both the query and the document to be searched, the system will be able to extract the relevant documents by finding how NEs in the document relate to the ones in the query.
- Question Answering. The NER task can be utilized in the phase of analyzing the question so as to recognize the NEs within the question that will help later in identifying the relevant documents and constructing the answer from relevant passages.

- Machine Translation. The type of the NE can help into deciding which part of the NE should be meaning-translated and which parts should be phoneme-transliterated.

- Text Clustering. Search results clustering will be able to rank the clusters based on the NEs. "This enhances the process of analyzing the nature of each cluster and also improves the clustering approach in terms of selected features". [12].

- Infoboxes: Facilitate the problem of searches which can result in a list of documents to be read by the user. Data sheets have been created in order to provide a summary of the most important data about the entity searched by the user. Examples of this type of data sheets, commonly known as "infoboxes", can be found in both general-purpose Web search engines such as Google or encyclopedic tools like Wikipedia. The construction of these infoboxes can be manual, semi-automatic, or fully automatic. [10]

- Text Summarizing: automatizing text summarizing using NER. Systems would be able to extract the most important information in order to construct a well formed summary. For example this case, in law related texts, information such as the Name of the Judge, the Defendant, the reason of the case, the decision taken... are of at most importance in the formation of a good summary, information that can be extracted by NER systems.

- Language Modelling: used in speech recognition, machine translation... it constraints searches by providing a likelihood of possible successor words.

- Sentiment Analysis: also known as Opinion analysis in order to determine the attitude or opinion of the subject about a certain matter. For example predicting the outcome of an election or knowing the general thought of the population about the voting of a new Law.

On its own, a NER can also provide users who are looking for person or organization names with quick information. NER systems were used in their early days primarily for extractions from journalistic articles. [17].

2.2 Technical Overview

This section discusses some of the major technical details behind NER system. Namely, the information and decisions that need to be taken in consideration before starting the implementation of such a tool.

Tab. 2: State of the Art: Summary of the Technical Overview

Name	Description
Parameters Affecting Performance	• Language • Corpora • Entities • Detail Level
Implementation Methods	• Rule Based Systems • Statistical Methods
Features	• Local Features • Global Features • List Look up Features • External Features
Processing Steps and Resources	• Corpora • Pre-processing • Feature processing • Post-processing • Output

2.2.1 Parameters Affecting Performance

There are many factors that can radically change and influence the performance of a NER system:

- **The language**: The first systems based on rules were build for a specific language and it was not possible to easily alter them to a different one. With the

advent of systems based on machine learning, it was possible to choose features independent on the language and use the system for different languages. [17]

- The domain of **corpora**, some domains seems to be easier for NER than others, e.g. news articles and texts from social networks.
- The types of **entities**, some categories of NEs are easier to find then others, e.g. countries are easier then organizations.
- The **levels of detail** The common NE categories are Person, Organization, Location (GPE), Date (and time), Numbers (of different kinds) and Miscellaneous. Another branch of NER is focused on biology and thus uses categories like Protein, DNA etc. [1].

2.2.2 Implementation methods

For the purpose of the development of NER systems, two main approaches can be followed:

- **Rule-based systems** are handcrafted (parameters set by human) and deterministic (assign for each word only one label). Those system use mainly Dictionaries, Regular Expressions and Context Free Grammars.
- **Statistical Methods** are stochastic (based on probability distributions, assign a set of labels and their probabilities for each word.) and use machine learning (estimated parameters by computer:Supervised learning—Semi-supervised learning—Unsupervised learning). Statistical methods for NER are modelling the probability distribution $p(y|x)$, where y is the sequence of NE classes and x is the sequence of words.

Applying the best approaches is not possible in all cases, since each approach presents different technical requirements. However, when the appropriate resources are available, Machine Learning based solutions present several advantages over other methods, and provide the best performance results. [9]

2.2.3 Features

In the process of recognition of the Entities to be detected, different ways have been implemented and can be used:

- **Local features:** use only a small neighborhood of the classified word such as Orthographic features (based on the appearance of the word, e.g. the first letter is a capital letter), Stemming and lemmatization, etc.
- **Global features:** uses the whole document or corpus. Sometimes some meta-information about the document is also considered as global feature.
- **List-look up features:** such as Gazetteers, Trigger words, etc.

- **External Features:** such as Wikipedia which is a rich source of information. In order for example to automatically create corpus for NER, automatic creation of gazetteer, for Disambiguation...[1]

2.2.4 Processing Steps and Resources:

Both training and annotation tasks depend on various processing steps and resources [9]

- Corpora: collection of texts related with the target domain;
  - Gold Standard Corpora (GSC): annotations are performed manually by expert annotators, following specific and detailed guidelines.
  - Silver Standard Corpora (SSC): annotations are automatically generated by computerized systems.
- Pre-processing: helps in the recognition process by processing the input from the full text to sentences to tokens.
  - Sentence splitting: process of breaking a text document into its respective sentences.
  - Tokenization: process of breaking a sentence into its constituent meaningful units, called tokens.
  - Annotation encoding: Internal representation of the annotated entity names.
- Feature processing: extract features from the pre-processed input data; Machine Learning model: use those features to automatically define a set of rules to learn the pattern and characteristics of entity names;
- Post-processing: Remove or correct recognition mistakes, Extend or make annotations more precise, Remove uninformative terms.
- Output

2.2.5 Tools

All the tools presented in the table bellow are free and use English as their main language.[6]

Tab. 3: Sample of available NER Tools.

Name	Description
LingPipe	Set of Java libraries developed by Alias-I for natural language processing.

Continued on next page

Table 3 – continued from previous page

Name	Description
ClearForest SWS	is a commercial tool made by ClearForest Ltd., currently acquired by Reuters. It allows the analysis of English texts and the identification of ENAMEX types, in addition to some other types such as products, currencies, etc.
Annie (GATE)	is open-source and under a GNU license, developed at the University of Sheffield. It is implemented in Java and incorporates in the form of plug-ins and libraries its own or external resources for a variety of aspects related to natural language processing.
Freeling	is a tool developed in C++ at the TALP Research Center of the Polytechnic University of Catalonia. It is an open source tool with GNU license that may be used as an API or independently.
Afner	is an open-source NERC tool, under GNU license, developed in C++ at the University of Macquaire. Currently it is used as part of a Question Answering tool called AnswerFinder, which is focusing to maximizing recall.
Supersense Tagger	is an open-source tagger developed in C++ with a version 2.0 Apache license. It is designed for the semantic tagging of nouns and verbs based on WordNet categories which include persons, organizations, locations, temporal expressions and quantities.
TextPro	tools suite is developed in C++ at the Center for Scientific Research and Technology (ITC-irst), in Trento, and offers various NLP functionalities interconnected in a pipeline order.
YooName	is a tool developed at the University of Ottawa by David Nadeau. It incorporates semi-supervised learning techniques applied to the web, that permit the identification of entities using a predefined classification of nine types of NEs (person, organization, location, miscellanea, facility, product, event, natural element and unit) and 100 subtypes.
IXA pipeline	modular set of Natural Language Processing tools (or pipes) which provide easy access to NLP technology. It offers robust and efficient linguistic annotation to both researchers and non-NLP experts with the aim of lowering the barriers of using NLP technology either for research purposes or for small industrial developers and SMEs.
NEREA	Automatic NER and NED system, whose main purpose is to obtain infoboxes, but with the particularity of being intended for local environments.
OpenNLP	is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution.

Continued on next page

Table 3 – continued from previous page

Name	Description
CoreNLP or StanfordNLP	is one of the most used NLP tools. It is based on a statistical model and it is known to be reliable and fast even on large input data. It supports several languages other than English, and it can be run as a simple web service.
NeuroNER	an easy-to-use program for named-entity recognition based on neural networks 2017
J-NERD	Based on a supervised, non-linear graphical model that combines multiple per-sentence models into an entity coherence-aware global model.
Spacy and TensorFlow	is a statistical tool for large-scale information extraction tasks. It is well-known for its speed in parsing very large textual input.

2.3 Recognition of Legal References

It is nowadays a common consensus, AI can perform tasks which save hours in billable time, AI is quickly becoming the new norm ranging all the way from Automatic Contract analysis, Smart document generation, Smart knowledge management, Data visualization (graphics) to facilitating interactions with the clients or Annotation and Extraction of Textual Legal Case Factors. [14]

Taking advantage of technology in the legal domain is however not a new idea. Legal informatics is a branch of applied computer science covering law related tasks. Legal information retrieval being one of the main research topics of legal informatics. The first attempt at summarization of legal documents dates back to 1970-1974. The project's aim was to automatically paraphrase German legal texts. Even though the project did not succeed, it is remarkable for being an early and large-scaled interdisciplinary enterprise [4].

In the legal domain, Named Entities are not only names of people, places or organizations, as in general-purpose NER. Named Entities are also names of laws, of typified procedures and even of concepts. [13]

A few interesting projects in the domain using NER systems can be taken as examples in order to have an idea of what the literature is focusing on nowadays:

- "NER study for ontology population. The NER module identifies Law, Act and Rule entities and classifies them. After that, passes the entities' list for OntoPopulate, which populates the received taxonomy with the entities as instances."[4]

- "Machine learning and NLP techniques are used for extracting legal rules on the basis of a semantic model for legislative texts, which is oriented to knowledge re-usability and sharing. More over the identified entities of the regulated domain can be a starting point to a bottom-up implementation of domain ontologies. This approach is aimed at giving a contribution to bridge the gap between consensus and authoritativeness in legal knowledge representation. The proposed approach is based on knowledge modelling oriented to interoperability and re-usability, and it is based on the separation between types of knowledge to be represented by Semantic Web standards."[22]
- "Work motivated by several immediate applications: case summarization, semi-structured search inside claim texts, structured search over claim entities... showing that the use of a combination of pseudo-likelihood and Gibbs sampling, outperform the top-down approach significantly."[15]
- "Detection and resolution of references to legislation, case law, parliamentary documents and official gazettes applied on EU documents."[24]
- NER system on Swedish text[25]

A few other projects are also worth mentioning and can be found in Tab. 4 below.

Tab. 4: Named entity recognition: software and projects related to the legal domain.

Name	Description
Eunomos	"Legal document management system. The TULE parser is used in order to recognize concepts and named entities as well as for classifying the legal documents stored in the system."[18]
Vico-Calegari, 2015	Software for document anonymization, Using OpenNLP with MultiNER.[18]
Gensim	"Scalable, robust realize unsupervised semantic modelling from plain text, such as probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). Gensim can be integrated with other Python libraries like NLTK (Natural Language Toolkit) for carrying out NLP pre-processing tasks."[18]
OpenSentenze	"Aims at publishing anonymized case law from Italian courts as open data. The TULE parser is used in a semi-automatic way together with the LIME annotator (University of Bologna)."[18]
SPeLT	"CIRSFID (UNIBO) has developed SPeLT (Semantic Parser of Legal Text), a framework of tools for parsing and analyzing legal texts. SPeLT has two main tools for parsing; SPELT-ref aim to identify legal references in judgments and other legal documents, while SPELT-struc aim to identify the logical structure of legal documents."[18]

Continued on next page

Table 4 – continued from previous page

Name	Description
Spelt tool	"Spelt tool detects from plain text the main legal knowledge information inside of the EU legal acts as: structure of the document, number of the document, type of document, authority that emits the act, normative references, dates, persons, organizations, locations, roles."[18]
Parse-IT	"Is a proprietary web-based system able to automatically analyze and translated legislation into sets of rules. The rules created by the tool can be used to perform various advanced legal reasoning tasks such as business process compliance and analysis of contracts."[18]

Similarly to the projects in Tab. 4 which are however in most cases European projects directly related to NER. The Tab. 5 below present the different companies worldwide offering services based on NLP systems or more generally using AI technology.

Tab. 5: Companies working on technology related to legal domain.

Name	Country	Description
Vlex	Spain	Large collection of legal information powered by Artificial Intelligence
ROSS	US	ROSS is an AI system designed to improve the efficiency, accuracy, and profitability of legal research.
LawGeex	US/Israel	Online analysis of contracts, resulting in the generation of a report that states which clauses don't meet common legal standards. The report also details any vital clauses that could be missing, and where existing clauses might require revision.
LeBonBail	France	Contract drafting service in compliance with a law that most people find incomprehensible.
Doctrine	France	It provides precise legal search of French legislation and jurisprudence in a few seconds.
VakiliSearch	India	Offers a platform for outsourcing paralegals services by establishing a relationship with third parties.
CaptainContract	France	Enable companies to generate various customized forms based on smart and quick forms

Continued on next page

Table 5 – continued from previous page

Name	Country	Description
LawBox	Belgium	Enable companies to generate various customized forms based on smart and quick forms.
Raven	UK	Group of applications that help automatically organize, discover and summarize documents.
Zegal	China	Automatic creation of documents
Kira	unknown	Uncovering relevant information from contracts and related documents.
Narrative Science	US	Data interpretation service, transforms data into Intelligent Narratives.
Mitra	India	Understand the context and relevance of a search query and provide the case researcher with the most accurate recommendation, helping to prepare defensible arguments.
BlueJlegal	Canada	Legal outcome prediction by helping with data analysis, pointing out missed parts in a fast, smart and accurate way
Seal Software	unknown	Help companies efficiently uncover what is in their contracts.
NExtLp	US	The service combines behavior and emotional analysis, unsupervised content classification and natural language processing to help users navigate unstructured data (emails, text messages, legal documents, etc.) and identify case-relevant facts.
Linklaters	unknown	Computer program that can sift through 14 UK and European regulatory registers to check client names for banks.
Garrigues	Spain	Started Using AI to speed up the process, whether for documents classification or analysis , ect
Urian Mandez	Spain	Started Using AI to speed up the process, whether for documents classification or analysis, ect
Ashrut	Spain	Started Using AI to speed up the process, whether for documents classification or analysis, ect
WING (National University of Singapore)	Singapore	NER in Legal Domain, aim to increase precision of existing Named Entity(NE) types, and train new NE types for the feature of legal domain(e.g. Law).

3 Algorithms and Services

In this project, a few features were required, flexibility and extendable architecture, stability and performance, maintainability, and finally a broad variety of entities to be detected. Similarly to [16], three methods for NE recognition are used in this project:

- Look-up: Dictionaries or lists terms needed to be detected in the text.
- Pattern rules: high accuracy can be obtained. However, this method required manually created development data and rule creation.
- Statistical models: requires manually annotated documents for learning purposes. However, already trained models were used for part of the entities to be detected, namely: Person, Location and Organization.

3.1 Algorithms

In the scope of this work, different algorithms, patterns and tools were used in order to create the proposed services. Those late ones were chosen because of the maintenance and reputation they have in the field. The Justification of each usage will be done in their respective sections namely 3.1.1 and 3.1.2

The tools used were mainly: GATE, OpenNLP, CoreNLP, IxaPipe, Apache Lucene, FuzzyWuzzy library in java.

The rules used were based on Regex with initial trials on PEG.

The algorithms used were mainly: Similarity mesures based on Jaro-Winkler algorithm which is a variation of the Damerau-Levenshtein algorithm.

All the queries and codes used will are available on Github[12]

3.1.1 Patterns and resources of the different codes used

The decision of using PEG rules combined with Regex rules was due to the limitations of those late ones. Indeed, time complexity of Regex patterns go from linear to exponential very fast. They also present a limitation when loops or more complex patterns are needed since they are dependent on the context. PEGs on the other hand are deterministic context-free languages making them a better choice since they can express Regex rules and have a similar writing pattern on top of the context independent advantage.[23]

[12] `https://github.com/ibadji/Legal-NER`

"Generally, because of their ability to cope with ambiguity, CFGs are often considered to be better suited for natural language processing than PEGs, but for references from the legal domain the non-ambiguity is an advantage rather than a drawback, and its – implicit – longest match recognition capability makes a PEG the better choice." [24]

In order to create the Dictionaries (look-up) and rules (Pattern rules) used, many resources were gathered. The following were the most important ones.

For the English (European Laws) the following was used as a base:

- Types of legal documents in EurLex[13]
- Citing European Union legislation [14]

For the Spanish laws the same process was followed:

- Guide for citing Legislation in Spain [15]
- A Brief Guide to Select Databases for Spanish-Speaking Jurisdictions [16]
- Guide To Legal Research in Spain [17]

Once gathered, the information was compiled and divided in two main categories, Dictionaries were filtered and separated in excel files while rules were created based on both the information about citation method found as well as text annotations (Rule formation process in Fig 5). The information that resulted from the extensive search is represented in trees in the Annex as well as excel sheets (described below, and can be found on Github) that will be used as dictionaries in the core code.
The Excel sheets regroup information about:

- English:
  - Abbreviations such as EU for European Union, ect
  - Judicial entities such as The Committee of the Regions with information about their status since some of them might not exist anymore but are still referred to in some documents.
  - Languages in the E-Union
- Spanish:

[13] http://eur-lex.europa.eu/content/tools/TableOfSectors/types_of_documents_in_eurlex.html?locale=en

[14] https://ilrb.cf.ac.uk/citingreferences/oscola/tutorial/page14.html

[15] http://biblioguias.uam.es/derecho/encontrar-legislacion-y-jurisprudencia/citar

[16] http://www.nyulawglobal.org/globalex/Databases_Spanish-Speaking_Jurisdictions.html

[17] http://www.nyulawglobal.org/globalex/Spain.html#Political

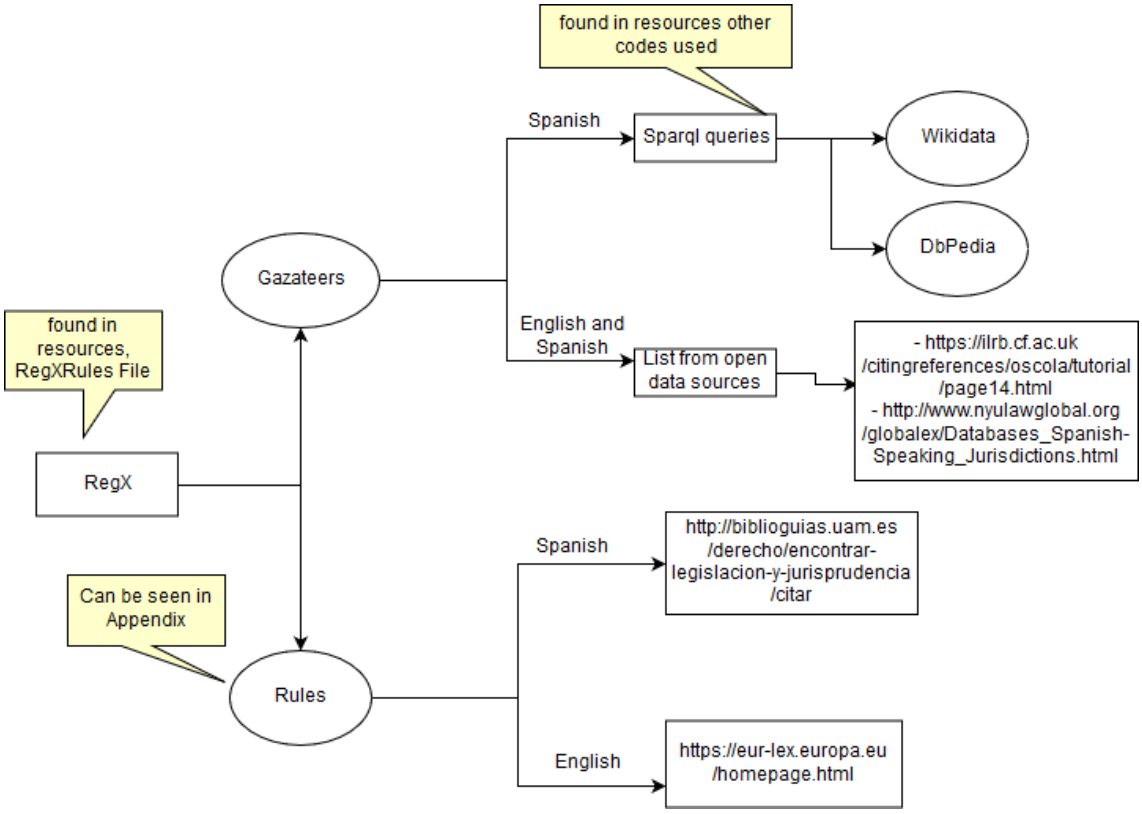

Fig. 5: Flow chart showing the processed followed in order to gather the information for the rule creation

- Abbreviation 1: art for artículo
- Abbreviation 2: such as AJEC for *Acuerdo de la Junta Electoral Central*
- Different Acts against a judgment
- Bulletins
- Cities
- Detection words such as *Demandante or Acusado*
- Geographical division
- Governmental institutions *Corts Valencianes*
- Regions
- Ministeries (need to be changed depending on the government)
- Law entities such as: *Audiencia Provincial* which are more general than the governmental institution
- Law sush as *Circulares, Consejo de Estado, Defensor del Pueblo*
- Others: are all those extra information found related to law that did not have a specific category to be put on such as *Sociedad de Responsabilidad Limitada, Ley de Aguas, Ley de Contratos de las Administraciones Publicas*

A representation of the government and law composition can be found in the Annex section. A sample of the patterns used on Spanish laws can be seen in Fig. 6 and Fig. 7. In total, 25 rules were created for Spanish and 38 rules for English. It should be noted that PEG rules in this case are used as a proof of concept rather than a full implementation, future work will include them fully.

```
Pattern[] articulo = {
            Pattern.compile("artículo\\s[0-9]{0,5}.[a-zA-Z]{1}"),
            Pattern.compile("artículo\\s[0-9]{0,5}.[0-9]{0,5}"),
            Pattern.compile("artículos\\s[0-9]{0,5}.[0-9]{0,5}.[0-9]{0,5}[0-9]{0,5},\\s[0-9]{0,5}\\s[a-zA-Z]\\s[0-9]{0,5}"),
            Pattern.compile("(art.|artículo|artículos|ART.) [0-9]{0,5} (CE|LRJSP)"),
};
Pattern[] Constitution = {
            Pattern.compile("CE\\w[0-9]{4}"),
            Pattern.compile("CE-[0-9]{1,7}-[0-9]{1,5}")
};
Pattern[] LeyOrganica = {
            Pattern.compile("Ley [0-9]{1,5}/[0-9]{1,5}"),
            Pattern.compile("(LRJSP|RDL) [0-9]{1,5}/[0-9]{4}")
};
```

Fig. 6: Regex rule sample

```
grammar URL

url       <-  scheme "://" host pathname search hash?
scheme    <-  "http" "s"?
host      <-  hostname port?
hostname  <-  segment ("." segment)*
segment   <-  [a-z0-9-]+
port      <-  ":" [0-9]+
pathname  <-  "/" [^ ?]*
search    <-  ("?" query:[^ #]*)?
hash      <-  "#" [^ ]*
```

Fig. 7: PEG rule sample

3.1.2 Algorithms for NER of legal references

The Algorithms and tools used were chosen due to their known high performance as well as the fact that they are well maintained The algorithms were mainly coded in Java and are all available on `https://github.com/ibadji/Legal-NER`, other codes in python and SPARQL queries can also be noted. Different algorithms and tools were used in order to answer the requirements of the project at hand. For that purpose, depending on the entities needed to be detected:

- Law References such as Regulations, Decisions...(that have an existing pattern such as *REC 14/2017*) in both English and Spanish used a combination of Regex and PEG rules. The PEG rules used independent codes using the canopy[18] on node.js while the Regex were applied on OpenNLP using the Java Regex Parser.
- Dictionaries were used for both Law References such as *High Court of Justice* as well as Nicknames of laws they were thus applied on OpenNLP, CoreNLP as well as the Nickname recognition algorithm.

[18] `http://canopy.jcoglan.com/langs/java.html`

- The Nickname algorithm used but was not limited to Apache Lucene [19] in order to have a text search that caters for changes. However, due to typos in the documents that tend to use nicknames such as tweets, two other algorithms were used: the FuzzyWuzzy Java version of the python code [20] which is a string matching algorithm using Levenshtein distance as well as a similarity algorithm based on the Jaro-Winkler[21] method. This last algorithm was used in order to detect the false positive outputted by the first two algorithms as well as the False negative.

- A python code was used outside the main code in order to fetch the corpus for the informal text namely Tweets.

- SPARQL queries were used in order to populate the nickname dictionary which was then filtered with OpenRefine [22] The Sparql queries for the retrival of law nicknames can also be found on Github. [23] [24]

In Fig. 8 and 9 below, a sample of the query used on the Spanish DBPedia is shown. The query fetched all information starting with *ley* or information belonging to categories *Leyes de España* that have as an external link a BOE reference.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?uri ?label ?sameAs_link ?redirect ?redirect_lable  ?external_redirect
WHERE {
?uri rdfs:label ?label.
FILTER regex( ?uri, "http://es.dbpedia.org/resource/Ley" )
?sameAs_link  (owl:sameAs|^owl:sameAs) ?uri.
#?sameAs_link  rdfs:label ?sameAs_label.
?uri dbo:wikiPageRedirects ?redirect.
?redirect rdfs:label ?redirect_lable.
?redirect dbo:wikiPageExternalLink ?external_redirect
FILTER regex( ?external_redirect, "http://www.boe.es" )
}
```

Fig. 8: Dbpedia Query

On Wikidata query in Fig. 10 below, different partial queries were run because of the nature of Wikidata. Indeed, the partial queries were run based on categories such as *Published in Boletín Oficial del Estado*. It should be noted that in order

[19] https://lucene.apache.org/core/
[20] https://github.com/xdrop/fuzzywuzzy
[21] https://github.com/tdebatty/java-string-similarity
[22] http://openrefine.org/
[23] https://github.com/ibadji/Legal-NER/tree/master/resources/other%20Codes%20used
[24] https://github.com/ibadji/Legal-NER/blob/master/src/main/java/Dictionary/WikiScrapper.java

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT DISTINCT (?uri), ?label, ?wiki, ?x
WHERE {
  ?uri rdfs:label ?label .
  ?uri <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?wiki .
optional {
  ?uri ?x <http://es.dbpedia.org/resource/Categoría:Leyes_de_España>
}
  FILTER regex( ?uri, 'http://es.dbpedia.org/resource/Ley_' )
} LIMIT 3000";
```

Fig. 9: Dbpedia Query

to execute the partial queries seen in Fig. 10 each part with a preceding comment should be run alone.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?item ?label ?cosa ?link ?URL ?p WHERE {

#BOE
  ?item wdt:P1433 wd:Q2029273.
 ?item rdfs:label ?label.
  ?item skos:altLabel ?cosa.
  FILTER((LANG(?cosa)) = "es")
  FILTER((LANG(?label)) = "es")

#Instance of law
?item wdt:P31 wd:Q7748.
?item rdfs:label ?label.
  ?item skos:altLabel ?cosa.
  FILTER regex( ?label, "(Ley|ley)+" )
  FILTER((LANG(?cosa)) = "es")
  FILTER((LANG(?label)) = "es")

#Jurisdiction
  ?item wdt:P1001 wd:Q29.
  ?item rdfs:label ?label.
  ?item skos:altLabel ?cosa.
  FILTER regex( ?label, "(Ley|ley)+" )
  FILTER((LANG(?cosa)) = "es")
  FILTER((LANG(?label)) = "es")
```

Fig. 10: Wikidata Query

A final fetch was made on BOE website in order to get a list of all the laws applied in Spain. The queries' output were filtered using a combination of GoogleRe-

fine and handmade modifications resulting in more than 2000 unique instances with on some instances Nicknames commonly used for them, around 100 nickname was included. Fig. 11 is a sample of the final output after the filtering, the first column represents the official name of the laws, the following columns are respectively the Wikidata/Wikipedia link, the DBPedia link, the BOE link followed by all the nicknames found with the query made for that specific law.

id	Name	Wikidata	DbPedia	BOE	nickname1	nickname2	nickname3	nickname4	nickname5
1	ley general de educacion de 1970	wd:Q9022041	http://es.dbpedia.org/resource/Ley_Organica_de_Educacion_(España)	http://www.boe.es/boe/dias/2006/05/04/pdfs/A17158-17207.pdf	ley organica de calidad de la educacion	ley organica de educacion	ley 14/1970	ley general de educacion	none
2	ley estatal de modernizacion del gobierno local		http://es.dbpedia.org/resource/Ley_Estatal_de_Modernizacion_del_Gobierno_Local_(España)	http://www.boe.es/boe/dias/2003/12/17/pdfs/A44771-44791.pdf	ley de grandes ciudades	none	none	none	none
3	ley de evaluacion de impacto ambiental		http://es.dbpedia.org/resource/Ley_de_Evaluacion_de_Impacto_Ambiental_de_España	http://www.boe.es/boe/dias/2010/03/25/pdfs/BOE-A-2010-4908.pdf	none	none	none	none	none
4	ley de reforma politica	wd:Q9022091	http://es.dbpedia.org/resource/Ley_de_Reforma_Politica	http://www.boe.es/g/es/bases_datos/doc.php?coleccion=iberlex&id=1977/00165	ley de la reforma politica	ley para la reforma politica	none	none	none
5	ley del mercado de valores		http://es.dbpedia.org/resource/Ley_del_Mercado_de_Valores	http://www.boe.es/buscar/act.php?id=BOE-A-1988-18764	none	none	none	none	none
6	ley antiterrorista espanola		http://es.dbpedia.org/resource/Ley_antiterrorista_española	http://www.boe.es/g/es/iberlex/bases_datos_tc/doc.php?coleccion=tc&id=SENTENCIA-1987-0196	legislacion antiterrorista espanola	none	none	none	none
7	ley organica 15/1999, de 13 de diciembre de 1999, de proteccion de datos de caracter personal		http://es.dbpedia.org/resource/Ley_Orgánica_15/1999,_de_13_de_diciembre_de_1999,_de_Protección_de_Datos_de_Carácter_Personal	http://www.boe.es/buscar/doc.php?id=BOE-A-2008-979	ley organica de proteccion de datos de caracter personal	ley organica de proteccion de datos	none	none	none
8	ley de promocion de autonomia personal y atencion a las personas dependientes	wd:Q9022074	http://es.dbpedia.org/resource/Ley_de_Promoción_de_Autonomía_Personal_y_Atención_a_las_Personas_Dependientes	http://www.boe.es/g/es/bases_datos/doc.php?coleccion=iberlex&id=2006/21990	ley de dependencia	servicio de ayuda a domicilio	none	none	none
9	ley del suelo		http://es.dbpedia.org/resource/Ley_del_Suelo_(España)	http://www.boe.es/boe/dias/1997/04/15/pdfs/A11773-11775.pdf	none	none	none	none	none
10	ley 38/1999, de 5 de noviembre, de ordenacion de la edificacion		http://es.dbpedia.org/resource/Ley_38/1999,_de_5_de_noviembre,_de_Ordenacion_de_la_Edificación	http://www.boe.es/boe/dias/1999/11/06/pdfs/A38925-38934.pdf	ley de ordenacion de la edificacion	none	none	none	none
11	ley organica 1/2004, de 28 de diciembre, de medidas de proteccion integral contra la violencia de genero		http://es.dbpedia.org/resource/Ley_Organica_1/2004,_de_28_de_diciembre,_de_Medidas_de_Proteccion_Integral_contra_la_Violencia_de_Género	http://www.boe.es/g/es/bases_datos/doc.php?coleccion=iberlex&id=2004/21760	violencia contra la mujer	ley integral contra la violencia de genero	none	none	none
12	ley organica 5/1985, de 19 de junio, del regimen electoral general		http://es.dbpedia.org/resource/Ley_Orgánica_5/1985,_de_19_de_junio,_del_Régimen_Electoral_General	http://www.boe.es/diario_boe/txt.php?id=BOE-A-1985-21380	legislacion electoral espanola	ley organica del regimen electoral general	ley electoral espanola	ley electoral	none
13	ley 46/1977, de 15 de octubre, de amnistia	wd:Q3753327	http://es.dbpedia.org/resource/Ley_de_Amnistía_de_1977	http://www.boe.es/aeboe/consultas/bases_datos/doc.php?id=BOE-A-1977-24937	ley de amnistia en espana de 1977	ley de amnistia de 1977	ley de amnistia de 1977 en espana	none	none

Fig. 11: SPARQL Query Result

3.2 Services

As represented in Fig 12, the user has three steps to follow:

- Choosing the language
- Choosing one of the services proposed
- Pasting the text that needs to be annotated

3.2.1 Main

The main services are in the number of two:

- Law Reference detection: in both Spanish and English using Regex.
- Nickname search: Only available in Spanish.
  Upon the presentation of a text, the algorithm will detect all the references to Law nicknames such as *Ley Mordaza or Ley Sinde* while taking in consideration possible typos. In the fetched tweets, the reference to *ley mirdaza* or *ley de sostenible economia* were found.
  Weather it is a typo, an order inversion or a structure difference such as upper case, lower case... the algorithm will be able to detect it and match it to the correct law as long as it is found in the dictionary.

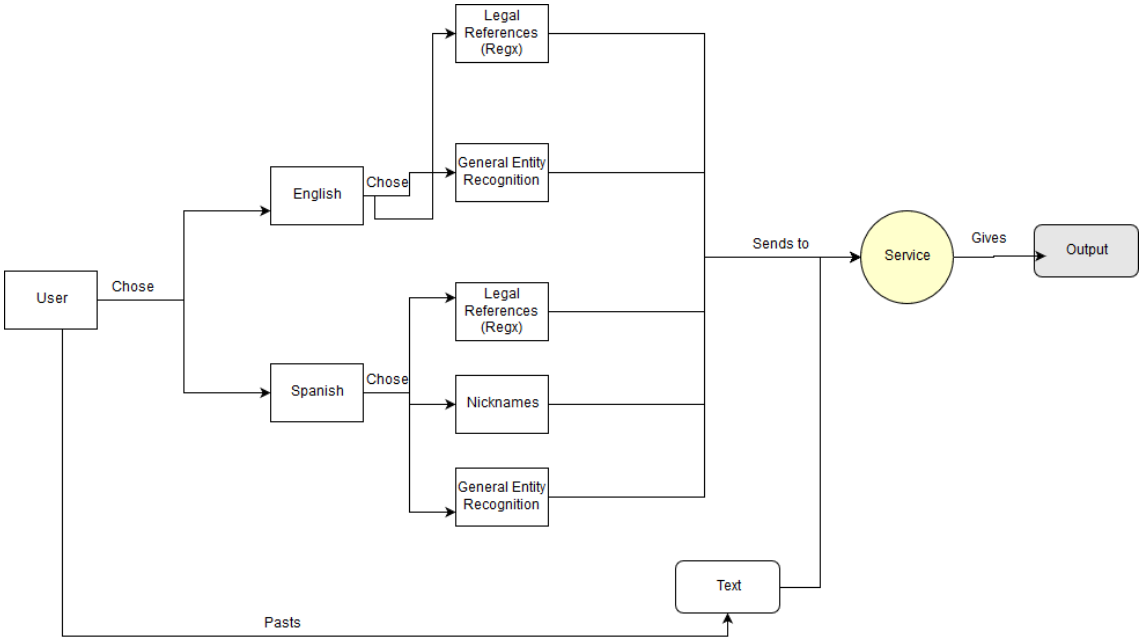

Fig. 12: Service Use from a user point of view

3.2.2 Auxiliary

The secondary service takes in consideration the other entities left out in the main service such as Abbreviations, Names of court laws, Name of people, location... the search is based on both dictionaries as well as a combination of already trained tools (Gate, OpenNLP, CoreNLP and IxaPipe).

4 Experimentation and evaluation

4.1 Implementation details

The algorithms and methods described in this work have been implemented and published on a Github software repo: `https://github.com/ibadji/Legal-NER`. The source code is open and available under an Apache license. The overall system works as described in Fig 7.

A text input is given to the main components of the code:

- The Nickname portion works on the combination of three text similarity algorithms: Apache Lucene and FuzzyWuzzy (Java version). Those two algorithms are both run on the input text matched with the Nickname dictionary gathered and filtered from DBPedia and Wikidata using Open Refine. The result of the run is then put through a basic Levenshtein algorithm in order to remove some of the noise and false positives that came from the first runs. The final output is a text file representing the the distance score followed by the real match and the corresponding string in the text. For example:

- The Gate, CoreNLP and IxaPipe operate the same way. The input text is given to them and the output is based on the pre-trained data on both Spanish and English text in order to detect entities such as Names, Locations...

- The OpenNLP, on top of the standard entity detection, Regex rules are added in order to find Law references in the input text.

- PEG is used to detect the Law references that need a more sofisticated approach such as loops or long phrases sentences not always covered by Regex.

- The combination Portion of the Code aims to filter out the redundant and false positive tags by using the Levenshtein similarity algorithm used in Nicknames as well as a priority and grading system. The priority system gives advantage to the entities created for the purpose of the project, for example: if when detecting *Ley 23/2017* OpenNLP detected it as *Ley Orgánica* while CoreNLP detected it as an *Organization*, the filtering should give priority to the *Ley Organica* and delete the reference to Organization. In other instances, when two different tools detect the same entity with the same entity tag two options are possible, either both of them are kept or a grading system is put in place were the tool with the most hit is given the best grade thus the priority. The grading system is updates with every run it however, at first gives the advantage to first OpenNLP and CoreNLP since after many runs they seem to be the most accurate tools considering the texts at hand.

- Linking: the linking portion takes place after the detection of all the entities in the text. This section aims to find more information using automatic Google searches and is devided in two main parts:

  - The nickname portion: in the case of the detection of nicknames, the algorithms first looks for the official name of the law in the dictionary created in order to search for it. Indeed when searching for the nickname of the law such as *Ley Sinde* no BOE link can be found, however when looking for *Ley*

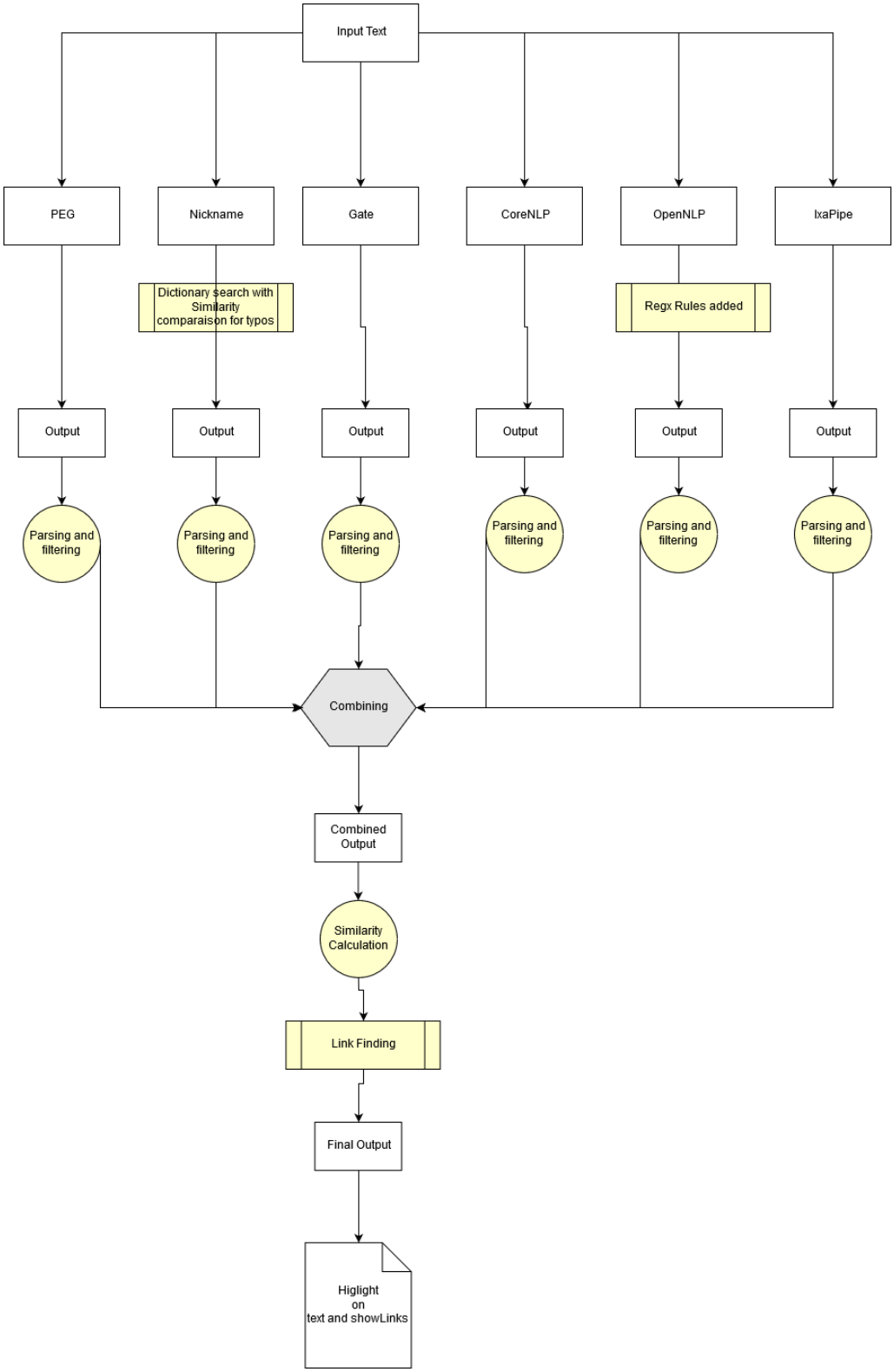

Fig. 13: Flow chart of the main components of the Code

```
0.79817045:::Tweet85:::ley mordaza:::Ley mordaza: "Qué
0.79817045:::Tweet216:::ley mordaza::: ley mordaza...
1:::Tweetx:::ley mordaza:::Ley mordaza
1:::Tweetx:::ley mordaza:::ley-mordaza
1:::Tweetx:::ley mordaza:::Ley mordaza
1:::Tweetx:::ley mordaza:::ley mirdaza
```

Fig. 14: Nicknames Output

```
Sevilla   :  [379..380) location
Tribunal Constitucional   :  [370..372) organization
artículo 43.1   :  [295..297) Articulo
artículos 112.1, 121 y 122   :  [7..13) Articulo
rec. 5244/2011   :  [344..346) Recurso
Ley 39/2015  :  [15..17) LeyOrganica
Ley 2/2011  :  [187..189) LeyOrganica
CE-001604-2015   :  [148..149) Constitution
```

Fig. 15: Rule Output

de economía sostenible BOE links can be noticed. Thus the need to first do a look up search on the dictionary (Excel file *Nicknames.xlsx* found in input document on github) to then lookitup on Google.

- The Law references go directly trough a Google search since they are considered to be presented in their official form such as *Ley 14/2018*.
- Organizations and Names: go trough a direct Google search as well, in this case the goal is to fetch the first (most accurate) link encountered.

- The final output is represented in two ways:
  - A text file with the detected entities, their category and links found (if any).

```
van Buitenlandse Zaken      [GateNLP, Person]        https://www.rijksoverheid.nl/ministeries/ministerie-van-buitenlandse-zaken
Sobitha Sumanan      [IxaPipe, Person]        https://eur-lex.europa.eu/legal-content/ES/ALL/%3Furi%3DCELEX%253A62017CN0680
Buitenlandse Zaken      [IxaPipe, Person]        https://www.rijksoverheid.nl/ministeries/ministerie-van-buitenlandse-zaken
Rechtbank Den Haag      [IxaPipe, Organization]        https://www.rechtspraak.nl/Organisatie-en-contact/Organisatie/Rechtbanken/Rechtbank-Den-Haag
Kamalaranee Vethanayagam      [IxaPipe, Person]        https://eur-lex.europa.eu/legal-content/ES/ALL/%3Furi%3DCELEX%253A62017CN0680
2018/C 063/13         [OpenNLP, OfficialJ]        https://eur-lex.europa.eu/legal-content/ES/ALL/%3Furi%3DOJ:C:2018:063:FULL
Case C-680/17         [OpenNLP, CaseLaw]        http://curia.europa.eu/juris/liste.jsf%3Flanguage%3Den%26jur%3DC,T,F%26num%3Dc-680/17
Case C-694/17         [OpenNLP, CaseLaw]        https://eur-lex.europa.eu/legal-content/EN/TXT/%3Furi%3DCELEX%253A62017CN0694
European Parliament         [OpenNLP, Organization]        http://www.europarl.europa.eu/portal/es
Netherlands      [GateNLP, Location]        https://en.wikipedia.org/wiki/Netherlands
Regulation (EC) No 810/2009         [OpenNLP, Regulation]        https://eur-lex.europa.eu/legal-content/ES/TXT/%3Furi%3Dcelex:32009R0810
2018/C 063/12         [OpenNLP, OfficialJ]        https://eur-lex.europa.eu/legal-content/ES/ALL/%3Furi%3DOJ:C:2018:063:FULL
```

Fig. 16: Final Output, Text Version

  - The input text with the detected entities highlighted on it

4.2 Running the Code

The main code can be found on `https://github.com/ibadji/Legal-NER` is organized as seen in Fig. 21. In order to run the code:

- Prerequisite: latest Java Version, NetBeans (8.2 version used)

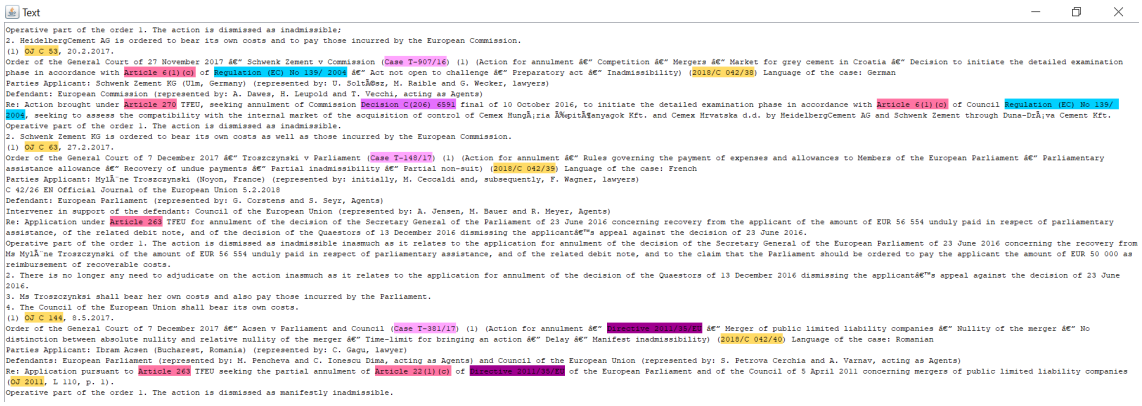

Fig. 17: Highlighted Corpus Focused on Law Entities

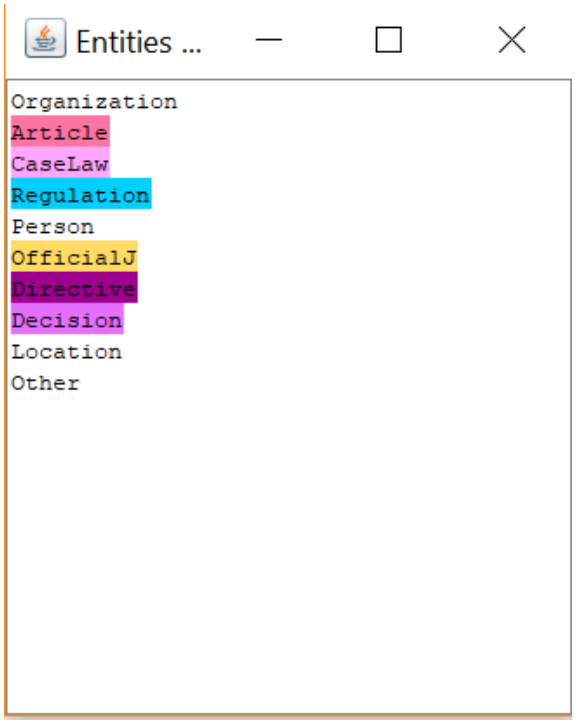

Fig. 18: Map for the Corpus in Fig. 17

- The code needs to be run using: *-Dfile.encoding=UTF-8*. On NetBeans it can be set by changing the VMoptions found by doing the following: run - project configuration - customize - run - compile and copy paste the instruction in the VMoptions box that shows in the compile portion.
- Download the code from Github or Clone the repository on your computer.
- The main.Java class can be run directly with the present information. The test is run on an EU court case found in resources/input, in English with the Type “rule”. Three types exist for Spanish: Nicknames, rule and other.
- In order to run the individual components such as CoreNLP in the NER folder, the

Fig. 19: Highlighted Corpus Focused on Person Organization and Location

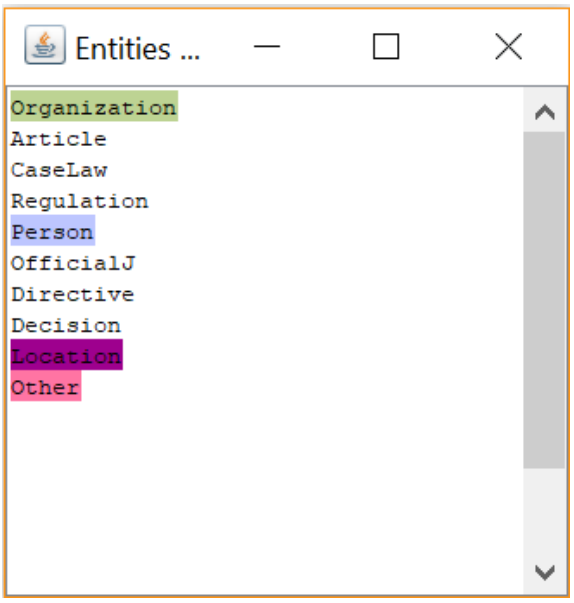

Fig. 20: Map for the Corpus in Fig. 19

main method needs to be changed similarly to the previous point.

- The output should resemble the ones seen on Fig.19 and Fig. 20.

Note: The Libraries used are heavy it might thus take time to load them as well as to run the code. Up to 4 minutes depending on the text length on a Toshiba Portege Z30 core i7, 8.00GB.

4.3 Corpus

The English corpus was created with documents from the EurLex official website. 10 different text documents were gathered, 3 from Directives (CELEX_32017L1371, CELEX_32017L1132, CELEX_32016L2102), 1 Order of General court (CELEX_62017TB0148) and 6 from EU court decisions (CELEX_62018CN0006, CELEX_62018TN0001, CELEX_62018TN0002, CELEX_62017CN0674, CELEX_62017CN0680, CELEX_62017CN0694). A bigger corpus

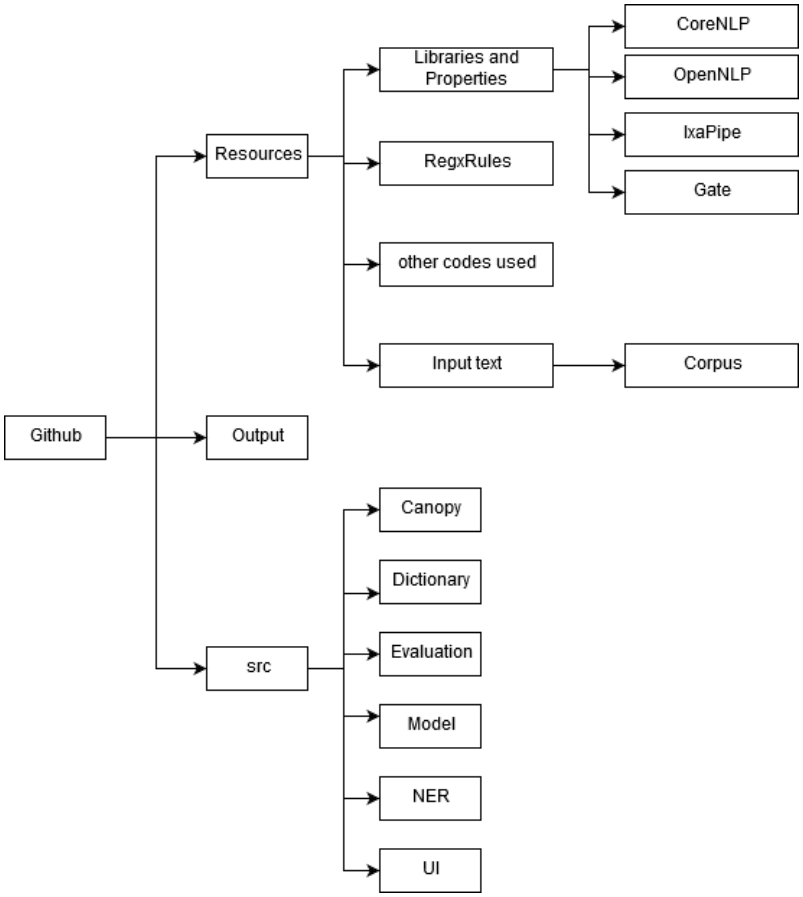

Fig. 21: Organization of github folders.

for the English language was not necessary in our case since the chosen documents covered the entities that were targeted. The Eurolex documents also present a very tidy and redundant way of presenting the information
Similarly for the Spanish Corpus, Lynx partners provided 10 sample text documents. In addition, 3 articles related to law were gathered as well as other texts from different origins namely, 4 documents from university decisions and scholarships proposals as well as 325 tweets.
The focus was on the Spanish language thus the importance of different types of documents. Overall the project was run on a total of 28 text documents[25]. Those same documents were also manually annotated in order to test for the accuracy of the algorithms and rules used.

The Gold standard was done by me. In the scope of this thesis it was not easy or possible to have a bigger corpus nor to receive the input of other colleagues or professionals in order to make sure that the way the gold standard was created was correct and the entities right. Thus the need in future work for a better corpus.

[25] `https://github.com/ibadji/Legal-NER/tree/master/resources/inputText`

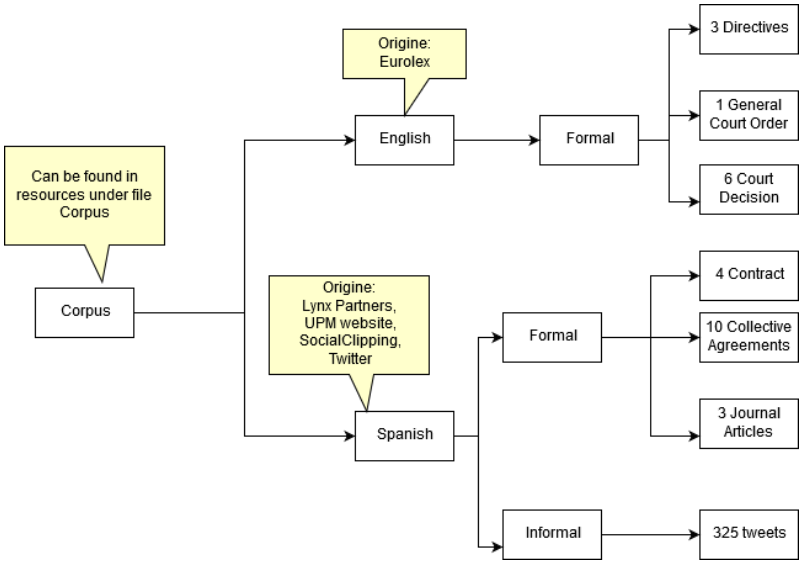

Fig. 22: Structure of the corpus

4.4 Methodology of evaluation

In order to evaluate the system created, a known general method for NER evaluation was used. The results are defined into four classes:

- True Positive (TP) - positive object marked as positive.
- True Negative (TN) - negative object marked as negative.
- False positive (FP)- negative object marked as positive.
- False negative (FN) - positive object marked as negative.

The method is based on metrics:

- Precision: measure of trust, that the objects marked as positive are really positive.
- Recall: measure of trust, that all the positive objects are marked.
- F-measure (also F-score or F1 score): harmonic mean between precision and recall and is something like overall perspective. (Named Entity Recognition, Michal Konkol (2012))

It should be noted that NER systems do not always output a complete and perfect detection of the entities. Example instead of detecting *Ms. Ines Badji*, the system might detect *Ines Badji* or simply *Badji*, which if evaluated automatically would be marked as wrong compared to the gold standard which would have annotated *Ms. Ines Badji* as the correct output. Thus the need for a more approximated detection rather than an exact one which would however introduce a lot of noise in the evaluation thus the decision of doing it manually in order to make sure that those approximations will not be a problem.

Another point should be taken into consideration, because of the novelty of the work, other companies or groups working on the same topic did not finish their work yet or do not provide their codes and methods freely. Even tough the services offered are different to the one at hand, I will not be able to compare to the state of the art.

4.5 Results

The following section describes the results of the runs on the system made on the 28 corpus against the same documents manually annotated. The experiment was run on a Toshiba Portege Z30 core i7, 8.00GB.

4.5.1 Results of NER of legal references

Precision, recall and f-measure were used in order to assess the results of the different runs.

- Accuracy - Ratio of correctly predicted observation to the total observations.
  Accuracy = TP+TN/TP+FP+FN+TN
- Precision - Ratio of correctly predicted positive observations to the total predicted positive observations.
  Precision = TP/TP+FP
- Recall - Ratio of correctly predicted positive observations to the all observations in actual class.
  Recall = TP/TP+FN
- F1 score - F1 Score is the weighted average of Precision and Recall.
  F1 Score = 2(Recall * Precision) / (Recall + Precision)*

It should be noted that in our case the True negative is a variable that was not taken in consideration since that would mean counting more or less every word/group of words present in the text. In order to run the tests, each service and language were run alone.

Overall, the F1 score for the English language on the rule and other service is respectively approximately 95% and 75%. The F1 score for the Spanish language is approximately 58% for the Other service, 53% for the Nickname service as well as 85% for the Rule service.

Starting with the English language on the European Union texts, the following was found: It can be noticed from figure Fig. 23, Fig. 24 and Fig. 25 the high accuracy of the rule service. Indeed the use of Regex rules insures high hits and avoid false positives. However, it can be noticed on some instances that the rules used did not detect all the law entities found in text mainly due to differences such as commas and spaces issue that will be resolved with the use of PEG rules. In other cases, the missed hits were due to new references to law entities that were not found on the official websites used to create the Regex rules for law entity detection. Also, in cases such as *Article 7(3) of Council Regulation (EC) No 139/2004* the current rules detect *Article 7(3)* and *Council Regulation (EC) No 139/2004* separately which can be considered incorrect in some cases were the same Article number is referenced for different regulations thus again the need to use PEG rules.
Similarly on Fig. 26, Fig. 27 and Fig. 28, the run resulted in a good Recall. However, because of the high False Positive, the overall score was lowered. The high number of hits is also due to partial hits such as *Tania Sanchez Lorenzo* and *Sra. Tania Sanchez Lorenzo* are considered two different entities, more broken down versions might exist as well such as *Tania, Lorenzo* and all those instances would be considered different even tough in reality

they refer to the same Person in this case. More filtering needs to be applied in order to remove a maximum of false hits. In this case interesting facts should be noted, Gate and IxaPipe are the most accurate at detecting Person names, while OpenNLP is very inaccurate, causing most of he False positive matches. Moreover, unexpectedly entities were detected by OpenNLP and not sited in their general detection entity list, entities such as *CAUSE OF DEATH, CRIMINAL CHARGE....*

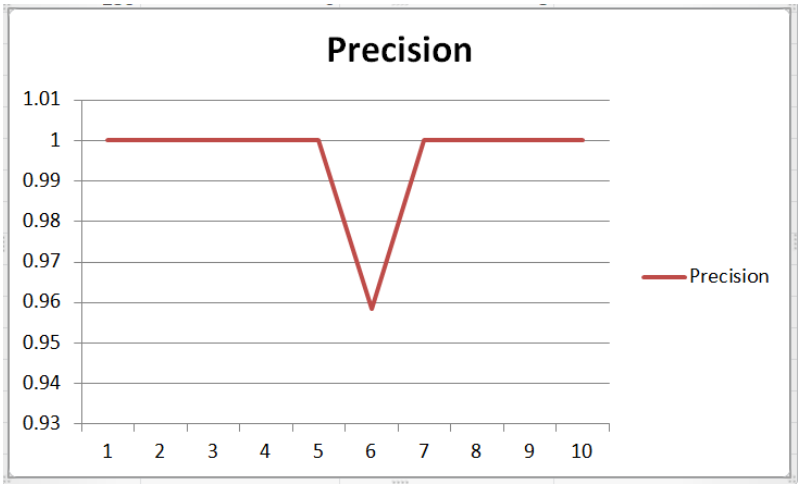

Fig. 23: English: Precision of the rule service

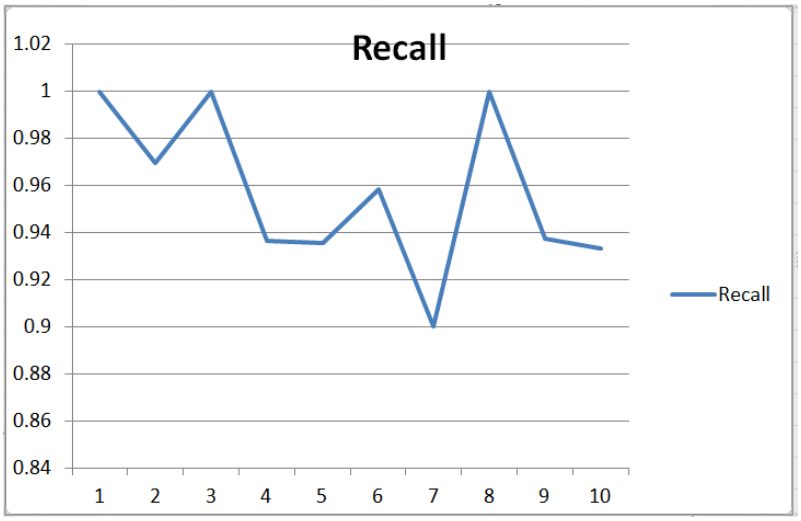

Fig. 24: English: Recall of the rule service

As for the Spanish language runs, similar results were obtained, with a high accuracy for the rule service. The other services that detects Person, Organization and Location has a high positive hit. However, the False positive hits that existed in the English language are even more accentuated because of the language showing the need for the use of a domain specific corpus in order to train the system. The results can be seen in Fig. 29 to Fig. 34.

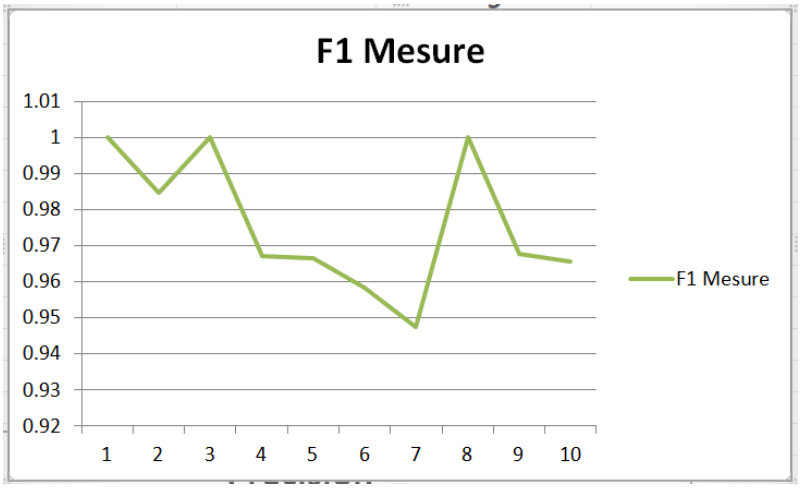

Fig. 25: English: F1 score of the rule service

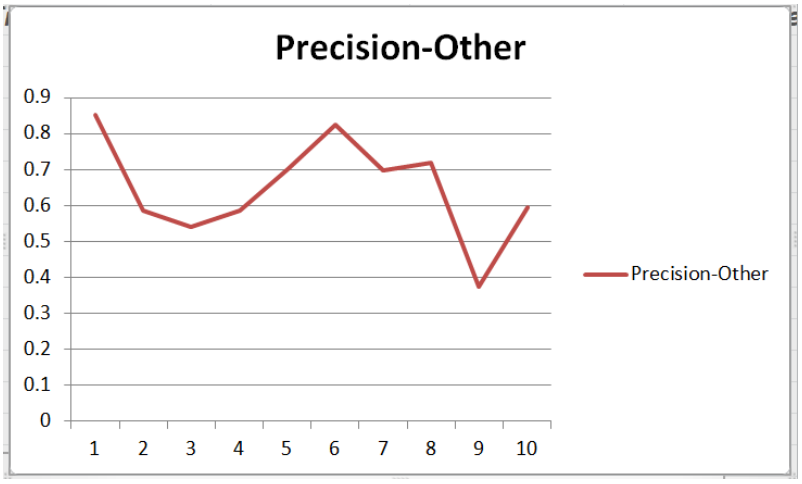

Fig. 26: English: Precision of the other service

Finally, when it comes to the Nickname detection run on the Tweets, it resulted in a Precision of 0.44, a Recall of 0.69 and an overall accuracy of 0.54. Those results are due to both the similarity algorithms which results in noise as well as a few missing nickname of laws in the original document use for the run. Following this test a more complete and accurate document was compiled and can be found on Github.

On a case by case, improvements can be made in order to have higher accuracy and better results:

- The Rule service on both languages: improvement can be made by using PEG rules on all of them as well as getting the opinion of professionals of the field in order to get more patterns of detection.

- The Other service: can be improved if machine learning is used. However, that method would need a high number of documents to be annotated which can be very time consuming which is why it was not done in this Thesis.

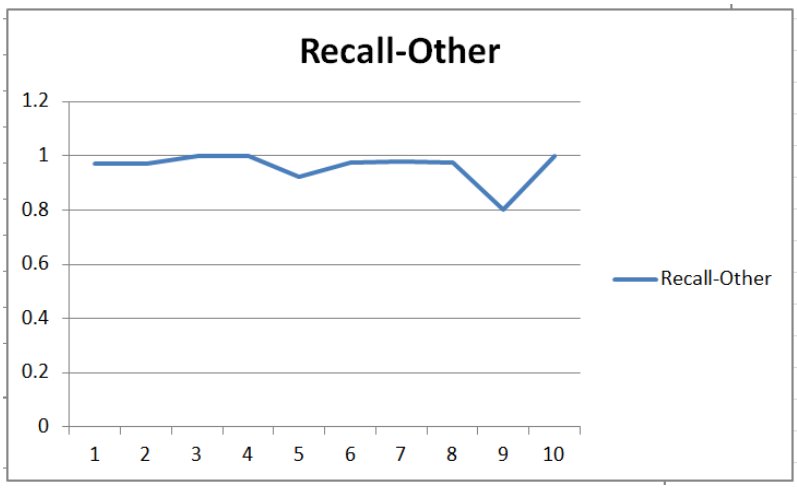

Fig. 27: English: Recall of the other service

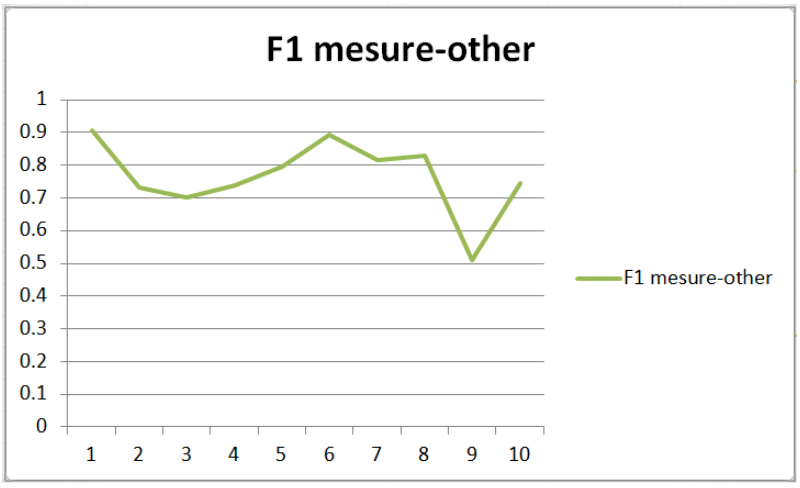

Fig. 28: English: F1 score of the other service

- The Nickname portion of the service can be improved by both a full list of law nicknames (which is now available) in order to improve the hits as well as reducing the noise of the similarity search.

4.5.2 Additional results

Even tough the main output of the work is the implemented algorithm to detect legal entities in text, it is however worthy to note some parts of the project that could be used independently in other contexts.

- Excel sheet presenting more than 2000 Spanish laws (*Ley, Ley Organica, Decreto Real*) with the most used Nicknames cited for some of tem.
- Dictionaries of Abbreviations, judicial authorities.. for both European Union and Spanish Government.
- Reusable SPARQL queries (DBPedia and Wikidata).

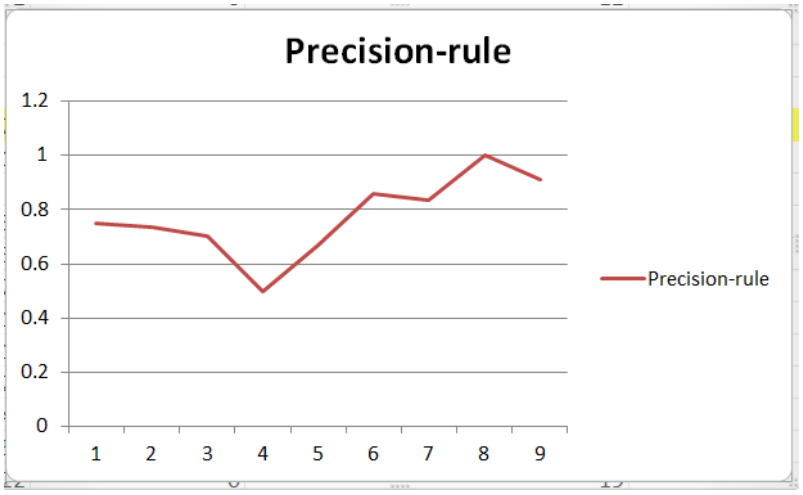

Fig. 29: Spanish: Precision of the rule service

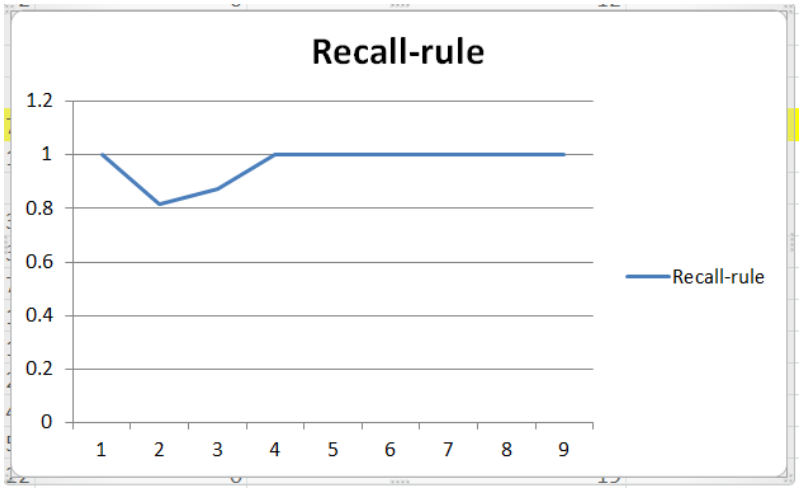

Fig. 30: Spanish: Recall of the rule service

- Reusable code for any language covered by the different tools.
- Libraries for text similarity and text search.

4.6 Publication as services Results

This work is public (licensed with a Creative Commons or Apache license) and Freely available on github `https://github.com/ibadji/Legal-NER`. It will be offered as a online service using the following link `http://api.lynx-project.eu/swagger-ui.html#\protect\kern-.1667em\relax/annotation/temporalUsingPOST`.

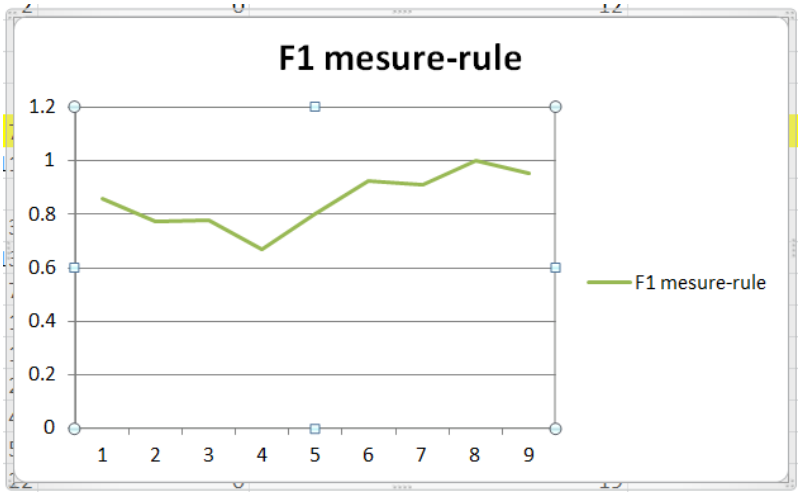

Fig. 31: Spanish: F1 score of the rule service

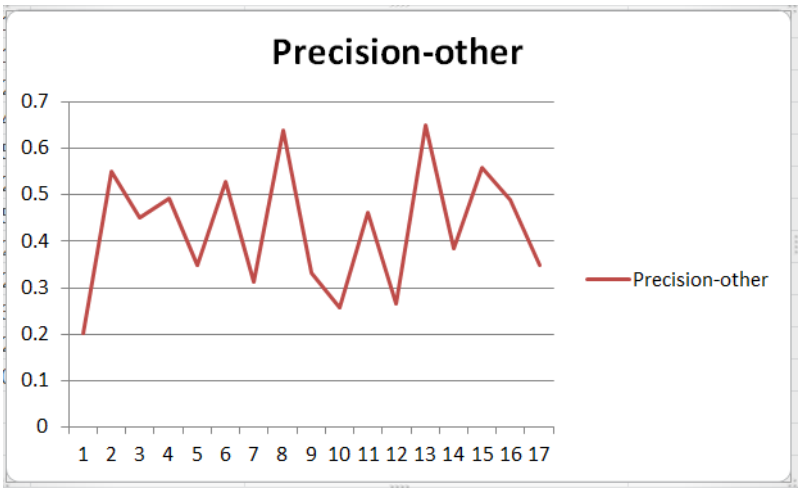

Fig. 32: Spanish: Precision of the other service

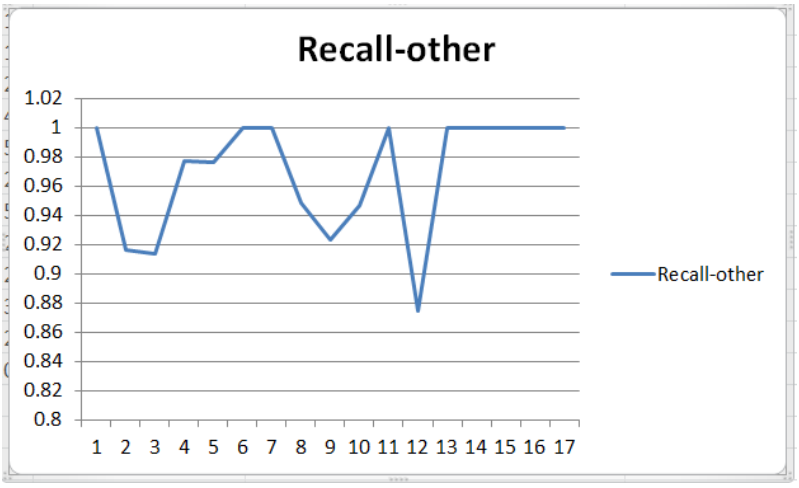

Fig. 33: Spanish: Recall of the other service

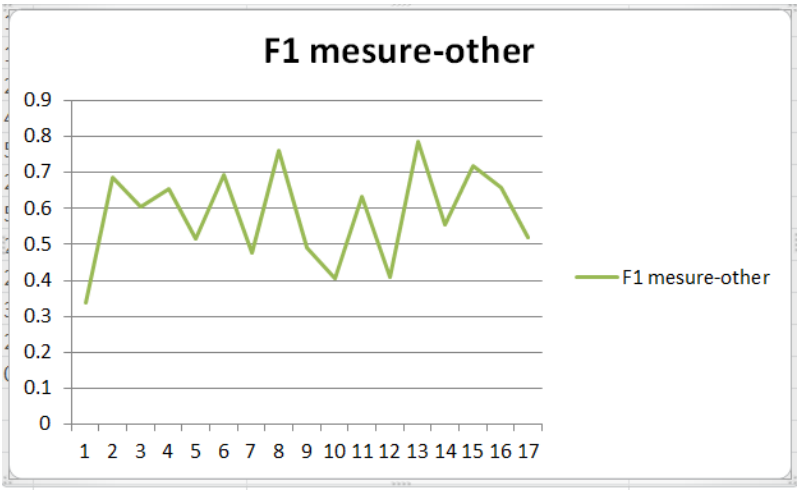

Fig. 34: Spanish: F1 score of the other service

5 Conclusions

In conclusion, this work contributes to the creation of a Named Entity Recognizer System for The Legal domain focusing on Spanish (Spain Legal System) and English (European Union) on both Formal and Informal documents. Those document can respectively be contracts, official laws or in the informal case, comments, tweets... Indeed the work presented in this thesis is novel for the domain and language it tackles and can be very useful when used either as a detection engine or as a base for further work such as text summarization, sentiment analysis... The developed tool is able to detect Law references in text using Regex rules, on top of it, using the different dictionaries gathered, nicknames of laws can be detected taking into consideration the possibility of typos in the text to be annotated. Finally using a combination of the different tools, the system also detects Names, Organizations and Locations using the pre-trained models of the different tools. Making it easily adaptable and portable to other languages and Legal systems. The advantages of the chosen approach are mainly Flexibility, portability to other languages/jurisdiction and availability as a web service. The Project also provides a free and open Access to the codes (Java) on github. However, limitations for the project can also be noted:

- Use of many heavy libraries making the runs slow and time consuming, however the project is aimed as a proof of concept, efficiency not being the main concern.
- Using the pre-trained tools makes the results not as accurate as they could be since the tools were not trained on the needed information
- Small Corpus.

However, even tough those limitations can be noted, the project still presents good results when it comes to the detected entities and can be adapted and improved in many ways, including the following potential propositions.
Future lines of work for improvement or evolution of the project would include but not limited to:

- Bigger corpus
- Better Gold Standard
- Collaboration with Lawyers in order to improve the Rules
- Using PEG rules on most if not all the law entities
- Extending the List of detected law entities
- Improving the recognition by training the systems on our own documents. Thus adding a Machine learning component to the project.
- Adding more information on the Nickname dictionary potentially by using machine learning algorithm on tweets, journals...
- Spreading it to different languages (Italian and German are foreseen). TULE could be used for Italian since REG rules are already implemented for it.
- Adding an entity tracking component in order to identify relations, co-references that may exist between the entities.

- Providing a more user friendly platform
- Using the system for Sentiment analysis about voted laws or law related opinions.

ANNEX

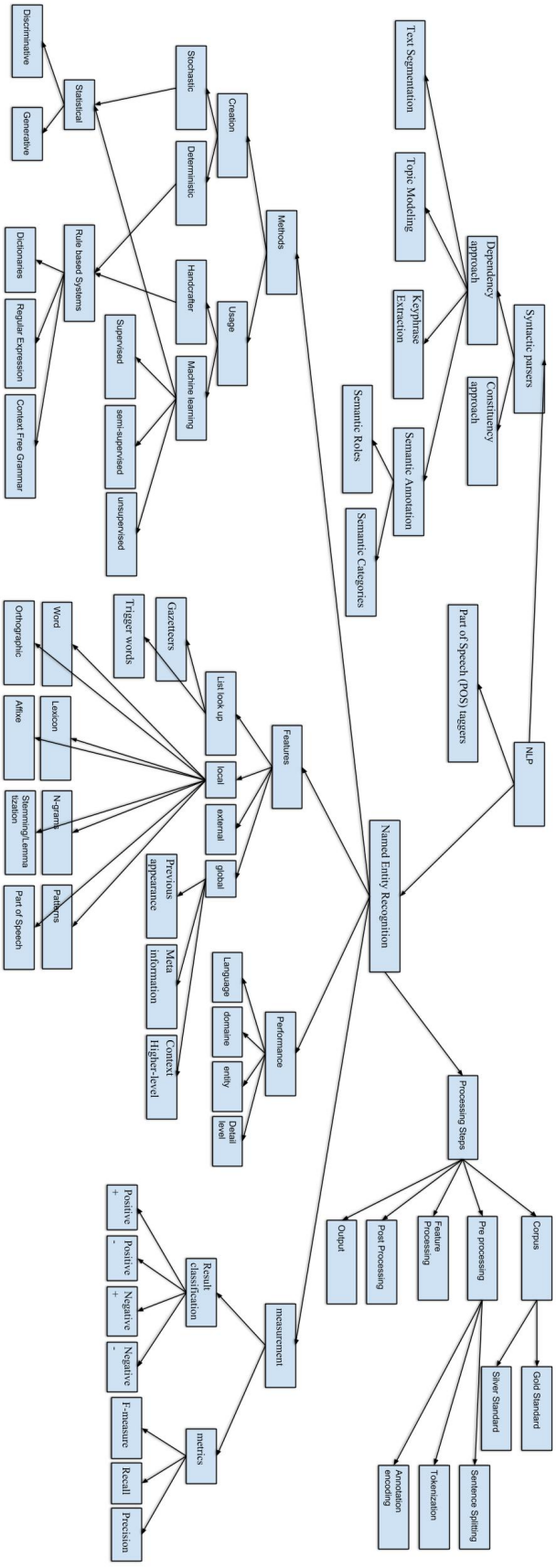

Fig. 35: NER State of the Art.

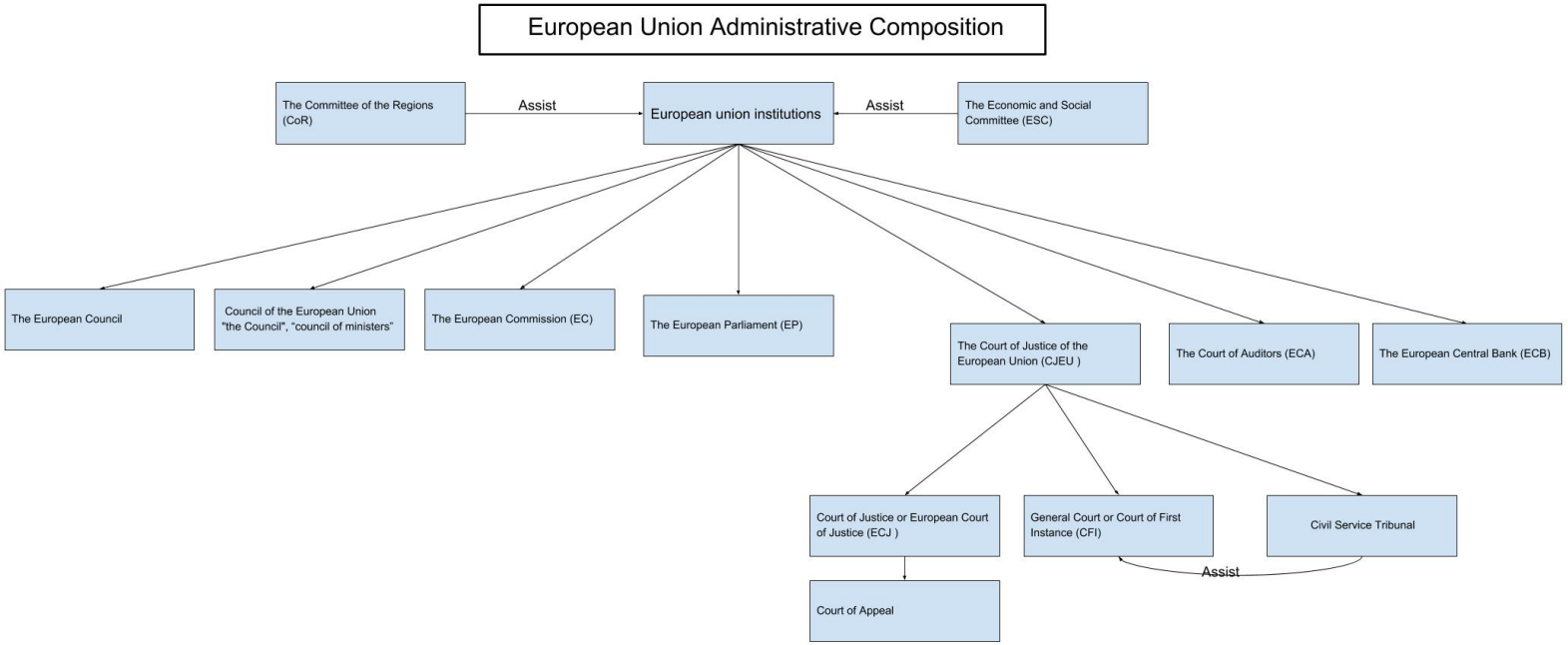

Fig. 36: European Union Administrative composition

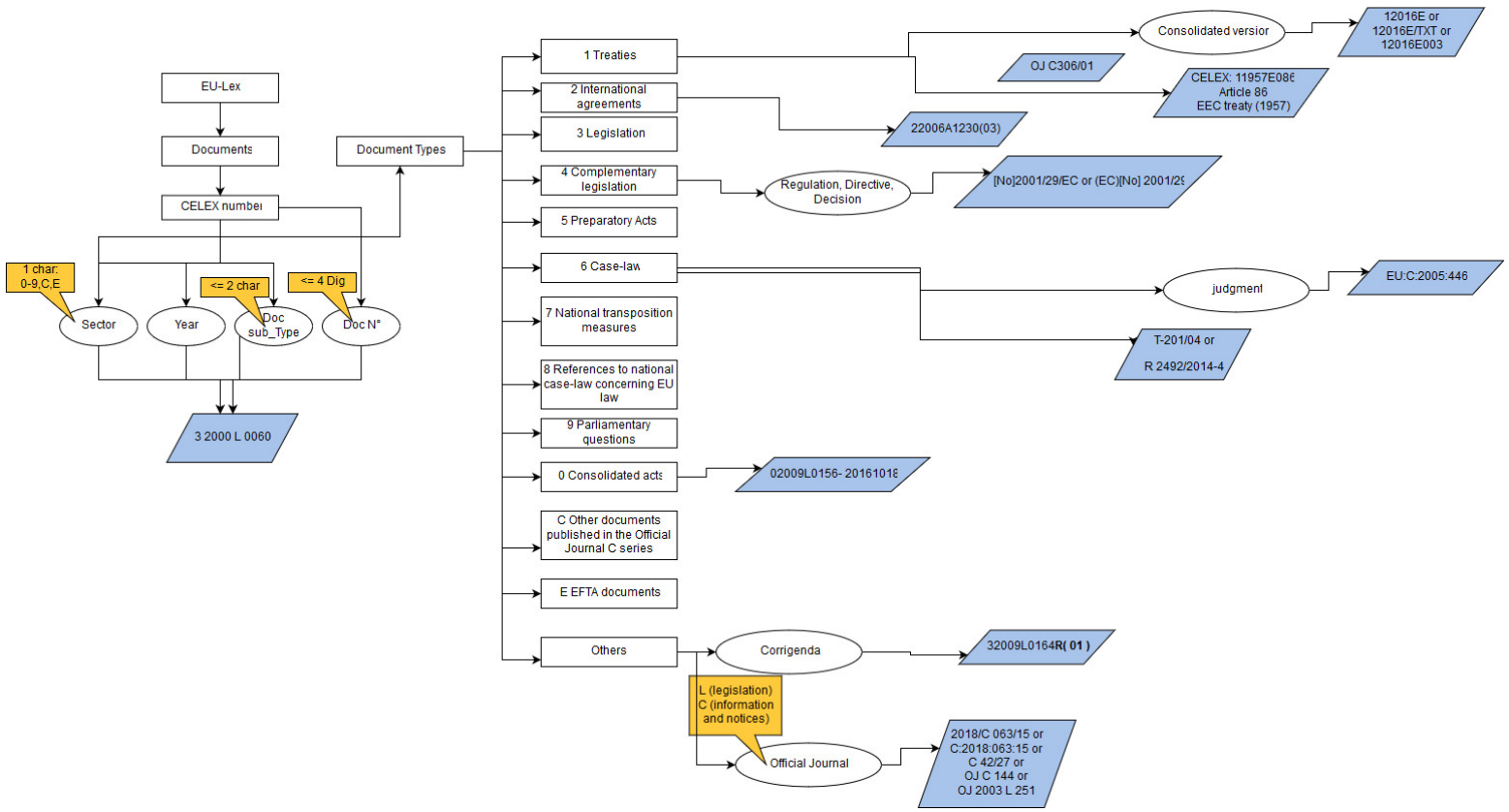

Fig. 37: European Union Law referencing

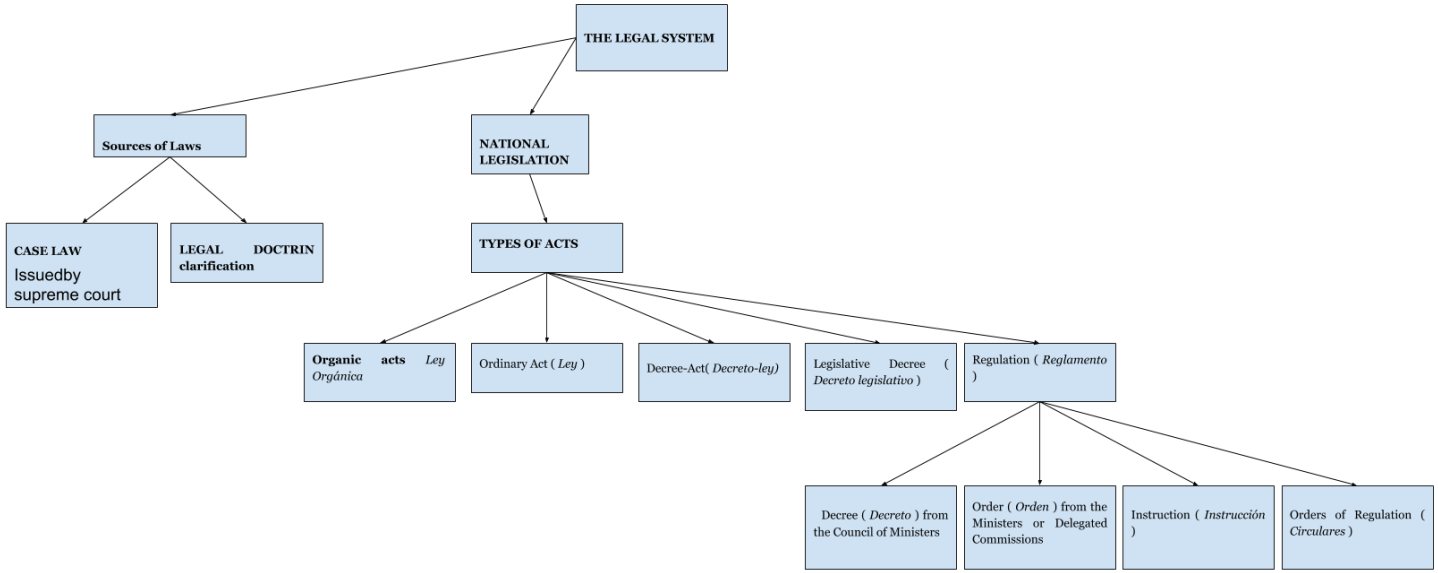

Fig. 38: Spanish Law

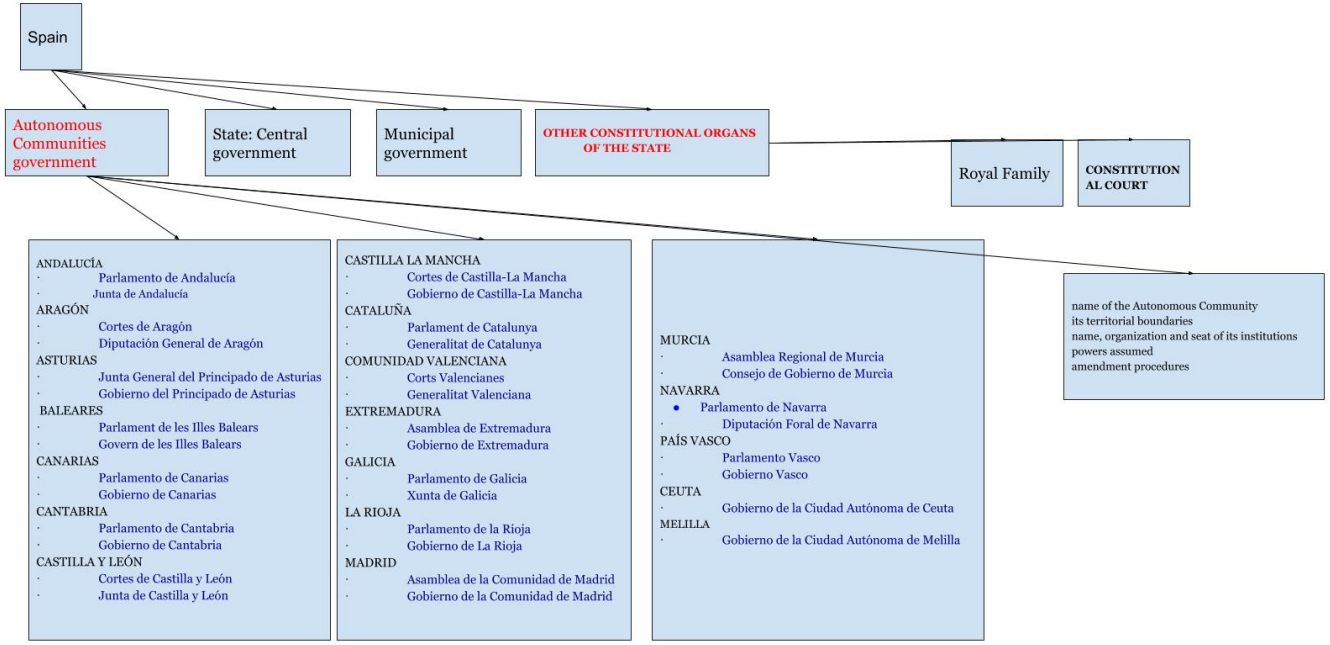

Fig. 39: Spanish Government: Part 1

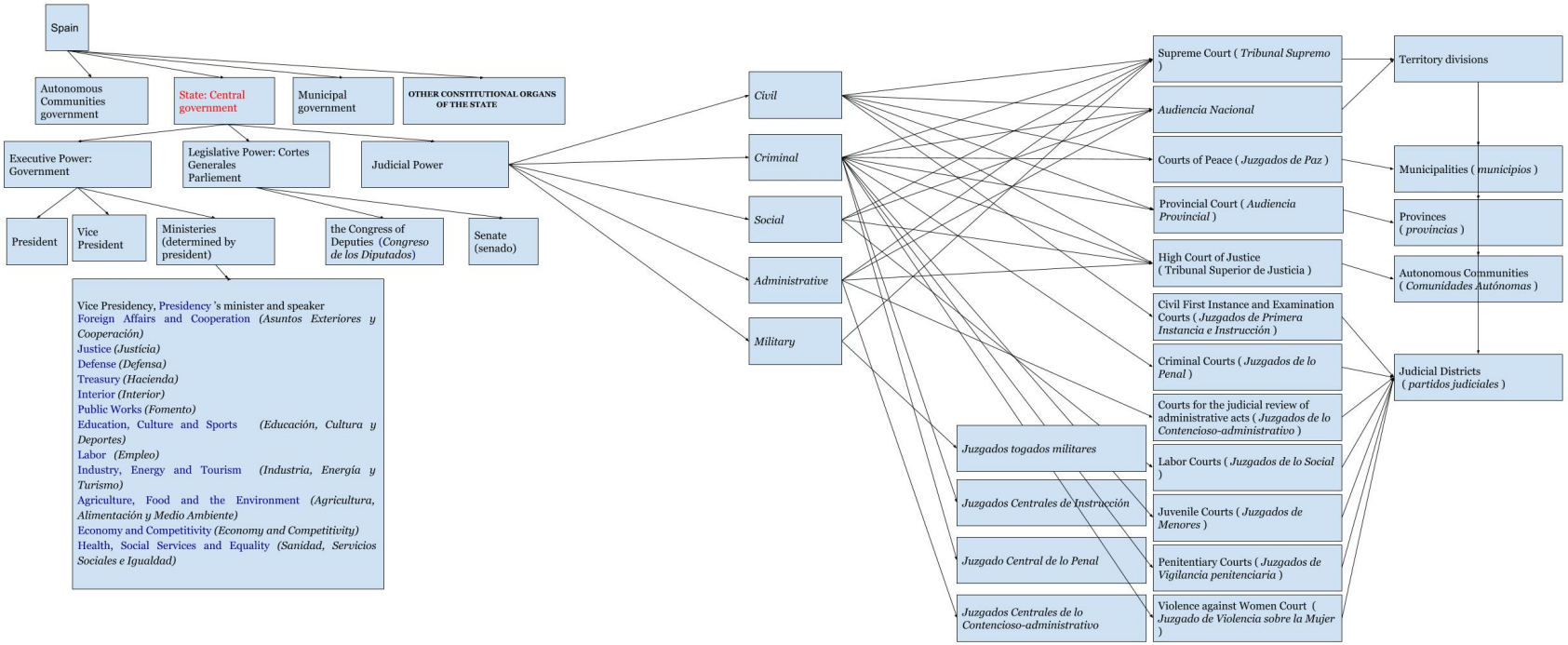

Fig. 40: Spanish Government: Part 2

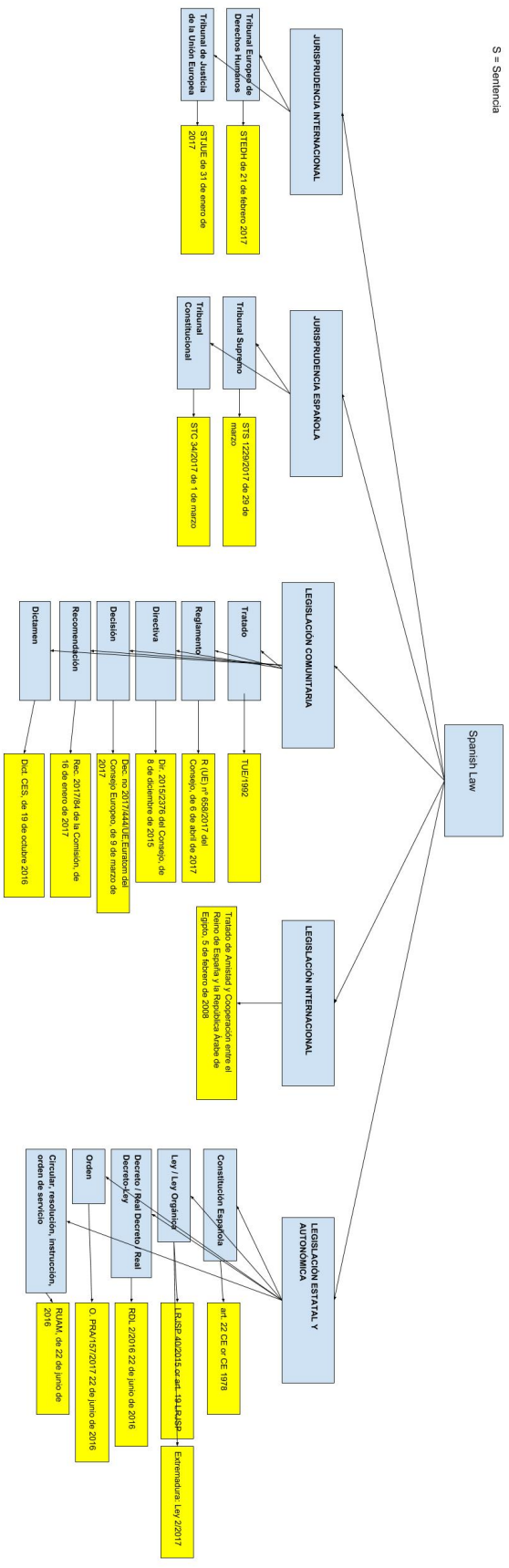

Fig. 41: Spain - Law Referencing

References

[1] Konkol, M. (2012). Named entity recognition: technical report no. DCSE/TR-2012-04.

[2] Nguyen, D. B., Theobald, M., & Weikum, G. (2016). J-NERD: joint named entity recognition and disambiguation with rich linguistic features. Transactions of the Association for Computational Linguistics, 4, 215-229.

[3] Van Hooland, S., De Wilde, M., Verborgh, R., Steiner, T., & Van de Walle, R. (2013). Exploring entity recognition and disambiguation for cultural heritage collections. Digital Scholarship in the Humanities, 30(2), 262-279.

[4] Bruckschen, M., Northfleet, C., Silva, D. M., Bridi, P., Granada, R., Vieira, R., ... & Sander, T. (2010). Named entity recognition in the legal domain for ontology population. In Workshop Programme (p. 16).

[5] Maarek, M. On the extraction of decisions and contributions from summaries of French legal IT contract cases. In Workshop Programme (p. 30).

[6] Marrero, M., Sanchez-Cuadrado, S., Lara, J. M., & Andreadakis, G. (2009). Evaluation of named entity extraction systems. Advances in Computational Linguistics, Research in Computing Science, 41, 47-58.

[7] Dernoncourt, F., Lee, J. Y., & Szolovits, P. (2017). NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. arXiv preprint arXiv:1705.05487.

[8] Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., & Anandkumar, A. (2017). Deep Active Learning for Named Entity Recognition. arXiv preprint arXiv:1707.05928.

[9] Campos, D., Matos, S., & Oliveira, J. L. (2012). Biomedical named entity recognition: a survey of machine-learning tools. In Theory and Applications for Advanced Text Mining. InTech.

[10] Garrido, A. L., Ilarri, S., Sangiao, S., Gañán, A., Bean, A., & Cardiel, O. (2016, November). NEREA: Named entity recognition and disambiguation exploiting local document repositories. In Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on (pp. 1035-1042). IEEE.

[11] Agerri, R., Bermudez, J., & Rigau, G. (2014, May). IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In LREC (Vol. 2014, pp. 3823-3828).

[12] Shaalan, K. (2014). A survey of arabic named entity recognition and classification. Computational Linguistics, 40(2), 469-510.

[13] Cardellino, C., Teruel, M., Alemany, L. A., & Villata, S. (2017, June). A low-cost, high-coverage legal named entity recognizer, classifier and linker. In Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law (pp. 9-18). ACM.

[14] LexisNexis, Lawyers and Robots? Conversations around the Future of the Legal Industry 3 (2017) (comment of David Halliwell of U.K. law firm Pinsent Masons).

[15] Surdeanu, M., Nallapati, R., & Manning, C. (2010, May). Legal claim identification: Information extraction with hierarchically labeled data. In Workshop Programme (p. 22).

[16] Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., & Wudali, R. (2010). Named entity recognition and resolution in legal text. In Semantic Processing of Legal Texts (pp. 27-43). Springer, Berlin, Heidelberg.

[17] Sun, B. (2010). Named entity recognition: Evaluation of existing systems (Master's thesis, Institutt for datateknikk og informasjonsvitenskap).

[18] MIREL Project. (2017).Collection of state-of-the-art NLP tools for processing of legal text.

[19] `http://www.upm.es/sfs/Rectorado/Vicerrectorado%20de%20Investigacion/Servicio%20de%20Investigacion/Ayudas_y_Convocatorias/ProgramaPropio/ProgramaPropio2018/Documentos/Ayudas_Contratos_Predoctorales.pdf`

[20] `http://lynx-project.eu`

[21] Navas-Loro, M. (2018) .LawORDate: a Service for Distinguishing Legal References from Temporal Expressions. In Proc. of the 1st Workshop on Technologies for Regulatory Compliance. Rodríguez-Doncel, V. et al. (eds.)

[22] Francesconi, E. 20100). Legal rules learning based on a semantic model for legislation. In Workshop Programme (p. 46).

[23] Medeiros, S., Mascarenhas, F., & Ierusalimschy, R. (2011). From regular expressions to parsing expression grammars. In Brazilian Symposium on Programming Languages.

[24] van Opijnen, M., Verwer, N., & Meijer, J. (2015). Beyond the experiment: the eXtendable legal link eXtractor. In Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts, held in conjunction with the 2015 International Conference on Artificial Intelligence and Law (ICAIL).

[25] Kokkinakis, D., Niemi, J., Hardwick, S., Lindén, K., & Borin, L. (2014, May). HFST-SweNER–A New NER Resource for Swedish. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).