FrEX: Extracting Property Expropriation Frame Entities from Real Cases*

Roberto Salvaneschi[1], Daniela Muradore[2], Andrea Stanchi[3], and Viviana Mascardi[1]

[1] Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, University of Genova, Italy,
ro.salvaneschi@gmail.com, viviana.mascardi@unige.it
[2] Freelance Lawyer in Milan, Italy,
avv.muradore@gmail.com
[3] Managing Partner of Stanchi Studio Legale, Milan, Italy,
a.stanchi@stanchilaw.it

Abstract. We describe FrEX, a Frame Entity eXtractor for real estate property expropriation cases rooted in Frame Semantics and heavily exploiting Natural Language Processing approaches. FrEX has been tested on 24 real, non anonymized cases shared with us by the Tribunal of Milan, described by almost 1000 PDF documents. FrEX results were compared with the relevant entities associated with those cases, namely debtors, creditors, lawyers, judges, experts, cadastral data of the property to be expropriated, manually inserted by domain experts. Although FrEX's development is still under way, the results are very encouraging and suggest that it can effectively relieve lawyers and judges from the highly repetitive task of looking for entities relevant for expropriation cases, when retrieving, filtering, and classifying legal documents.

Keywords: NLP4Law, civil law, property expropriation case, frame semantics.

1 Introduction

Artificial Intelligence (AI) will transform the field of law. This is widely recognized by professionals involved in both AI and law, and by observers of societal changes[4]. For AI experts the potential, limitations and risks of predictive justice and algorithmic law are almost clear, and many technical journals[5] and conferences[6] address these themes.

[4] https://www.forbes.com/sites/robtoews/2019/12/19/ai-will-transform-the-field-of-law/, published on December 2019; https://www.technewsworld.com/story/86521.html, published on February 2020.

[5] *Artificial Intelligence and Law*, https://www.springer.com/journal/10506.

[6] *JURIX, the International Conference on Legal Knowledge and Information Systems*, http://jurix.nl/; *ICAIL, the International Conference on Artificial Intelligence and Law* https://dl.acm.org/conference/icail.

However, despite many efforts to make AI and legal experts reach the same knowledge and awareness[7], most professionals from the law field are overwhelmed by news about smart courts and AI-powered judges and cannot easily understand the technical tools behind these robotic surrogates. Sometimes they are worried, and they are not completely wrong. At the time being, all the "robotic judges" used in trials employ machine learning, most often deep learning, and some of them became famous for their biased decisions. The *State v Loomis 881 N.W.2d 749 (Wis. 2016) case* is one among the most well known examples: the Wisconsin Supreme Court upheld a lower court's sentencing decision informed by a COMPAS risk assessment report and rejected the defendant's appeal on the grounds of the right to due process. COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a case management and decision support tool developed and owned by Equivant[8], "legally opaque", because its source code cannot be inspected, and "technically opaque", being based on deep learning [24]. Using machine learning for boosting predictive justice is becoming a very lively research field, although in many cases the developed applications are academic prototypes, not yet used in real trials. Applications range from predicting decisions of the European Court of Human Rights [26] to predicting recidivism of many different crimes [8], to risk assessment in criminal justice [5]. Actually, many scientists warn about opaque predictive models also from a technical point of view, besides an ethical one [37], and advocate the adoption of interpretable models instead [34]. Looking at the struggle between scientists pushing some approaches and scientists warning against them, judges and lawyers, whose computer literacy is often a basic one (as shown for example by the results of a questionnaire compiled by 17 magistrates in 2020 [25]), become more and more confused. Is AI good or bad for them?

In this paper we present the results of a project involving two computer scientists and two lawyers, aimed at implementing a tool that suits the lawyers' needs and that – up to the authors – does not hide any unexpected ethical threat. The tool, named FrEX (Frame Entity eXtractor), is a Python application that roots into Frame Semantics [15] and exploits Natural Language Processing (NLP) to identify the *main actors*, their *role*, and the *cadastral data* of real estates in property expropriation cases. Its purpose it neither predicting trial results nor making risk assessments, but rather making life of professionals in the legal field easier by automatically retrieving data that would require a manual inspection of thousand documents otherwise.

The paper is organized as follows: Section 2 introduces the FrEX domain and overviews the related works; Section 3 describes the FrEX design and implementation; Section 4 presents the results of our experiments on real cases; Section 5 concludes and discusses the future developments of FrEX.

[7] The *International Association for AI and Law*, `http://www.iaail.org/`; the *Stanford Artificial Intelligence & Law Society*, `https://law.stanford.edu/stanford-artificial-intelligence-law-society-sails/`; the *Digital Forensics: Evidence Analysis via Intelligent Systems and Practices (DigForASP) COST Action, CA17124*, `https://digforasp.uca.es`, involving magistrates and lawyers as participants.

[8] `https://www.equivant.com/`, last accessed September 2020.

2 Background and Related Work

Background. The FrEX application domain is that of real estate distraint. Suppose that a subject is a creditor of a certain sum of money towards another, be it a person or a financial institution/company: even if the creditor manages to obtain a sentence that establishes her claim, her right cannot be satisfied if the debtor refused to give her that sum of money spontaneously. This credit can be compensated for by forced execution, by proceeding with the expropriation of a real estate of the debtor. During their life cycle, real estate expropriation cases enter in many phases and a variable amount of documents in digital form[9] is associated with each of them. Our data set consists of 24 expropriation cases for a total of 1157 PDF files that the Tribunal of Milan shared with us, under Non Disclosure Agreement. Each case also has a variable set of XML files associated with, including no XML file at all. These XML files are generated by hand, and contain structured information about the entities and the parties involved in the case including name, surname and fiscal code of debtors, creditors, lawyers, judges, experts, along with cadastral data of the debtor's real estate that should be distrained.

According to frame semantics as defined in the FrameNet portal[10], "*the meanings of most words can best be understood on the basis of a semantic frame, a description of a type of event, relation, or entity and the participants in it.*" Although FrameNet provides some frames for the legal domain, no frame suits our needs by involving debtors, creditors, and the property that is used to satisfy the credit. Indeed, "creditor" and "debtor" lexical units are not even included in the FrameNet database. We designed our own frame as follows with the aim of complementing FrameNet from a logical point of view, but no integration with the database contents has been performed so far. For space constraint, we do not provide the definition of the entities, since they can be found in any dictionary:

Real_Estate_Distraint_Frame

Core Entities
- Debtor(s)
- Creditor(s)
- Real estate(s) to be distrained

Non-Core Entities
- Lawyer(s)
- Judge(s)
- Expert(s)

FrEX aims at automatically extracting the frame entities above from the PDF files associated with property expropriation cases.

[9] The Civil Telematic Process ("PCT") is a project initiated by the Italian Ministry of Justice for improving the quality of judicial services in the civil law sector by making – besides other goals – documents associated with sentences available in digital form.

[10] FrameNet is a lexical database containing over 1,200 semantic frames, 13,000 lexical units, 202,000 example sentences, `https://framenet.icsi.berkeley.edu`, accessed on September, 2020.

Related Work. "*Many industries have embraced NLP approaches, which have altered healthcare, finance, education and other fields. The legal domain however remains largely underrepresented in the NLP literature despite its enormous potential for generating interesting research problems.*" This statement, quoted from the Introduction to the Proceedings of the First Natural Legal Language Processing Workshop [3], describes the current situation of NLP in the legal domain. Scientists and legal experts are in fact just starting to understand that, among the many AI techniques and tools, NLP may play a major role to help lawyers and magistrates in document filtering, tagging and retrieval, without generating those ethical and legal concerns that algorithms used for predictive justice may indeed raise.

Although the idea of exploiting NLP for law has been explored, in a fragmented way, since the seventies of the last century [18, 22, 27, 38], and even more in recent times [16, 19, 36], the full awareness of its potential is almost recent. As an example, the MIning and REasoning with Legal texts project[11] funded by the European Union's Horizon 2020 research and innovation programme closed at the end of 2019, the Natural Legal Language Processing Workshop was at its second edition in 2020 [2], and the special issue *NLP for legal texts* of the AI and Law journal was published in 2019 [33].

Many research activities and projects in the NLP and law domain are now being carried out; those more closely related to ours can be divided into two main categories: those where NLP is exploited for performing highly repetitive tasks in the legal domain and those dealing with frame semantics for the legal domain.

Most works in the first category address the problem of summarizing legal texts and courts decisions [14, 17, 20, 29, 32], that is very distant from the problem we tackle. The works by Cardellino et al. [10, 11], who implemented an information extraction tool based on active learning for natural language licenses that need to be translated to RDF, are the most similar to ours, but the different application domain and the different approach they followed make them not easily comparable.

As far as the adoption of frame semantics for the legal domain is concerned, a few attempts exist, along with works on legal ontologies and their learning from text. Ontology learning [12] requires to identify the main concepts that characterize a document, taking their lexical semantics into account, the role they play therein, and to associate them with the right existing concepts in the ontology, if any, or to create new ones. This "entity finding" activity, aimed at putting each concept or individual in the right place in the ontology, shares some similarities with the frame entity finding activity described in this paper. The connections between FrameNet-style knowledge description and ontologies has indeed been recognized by many authors, also for managing fundamental legal concepts [1]. In a paper dating back to 2009 [39], Venturi et al. focussed on methodological and design issues, ranging from the customization and extension of the general FrameNet for the legal domain to the linking of the developed resource with already existing Legal Ontologies. Bertoldi et al., instead, pointed out the limitations of using FrameNet frames to build legal ontologies [7] and moved some initial steps in the development of a legal frame-based lexicon for the Brazilian legal language [6]. Although not based on frames, we may also mention – being tailored to the Italian language – the work by Lenci et al. who presented a method and preliminary results of

[11] `https://www.mirelproject.eu/index.html`, accessed on September 2020.

a case study in automatically extracting ontological knowledge from Italian legislative texts [23].

3 FrEX Design and Implementation

For designing FrEX we followed the pipes and filters architectural pattern [28] where independent entities, called filters, perform transformations on data. Once filters have processed the received input, the other type of component, called pipes, serve as connectors for the stream of data being transformed in the way requested by the program. Figure 1 shows the FrEX pipes and filters:

- white rectangles stand either for inputs to FrEX (the PDF and XML files), or for off-the-shelf software resources that we used to implement FrEX (the RDRP Tagger and the Italian Dictionary);
- blue ellipses are filters that receive in input what written in their entering arrows and return as output what specified in the outgoing arrows;
- green rectangles represent the FrEX final output;
- the orange rectangle stands for a feature under development at the time of writing: the filter that links cadastral data in their standard, but barely readable, Italian cadaster format with their explicit civic address is not yet available.

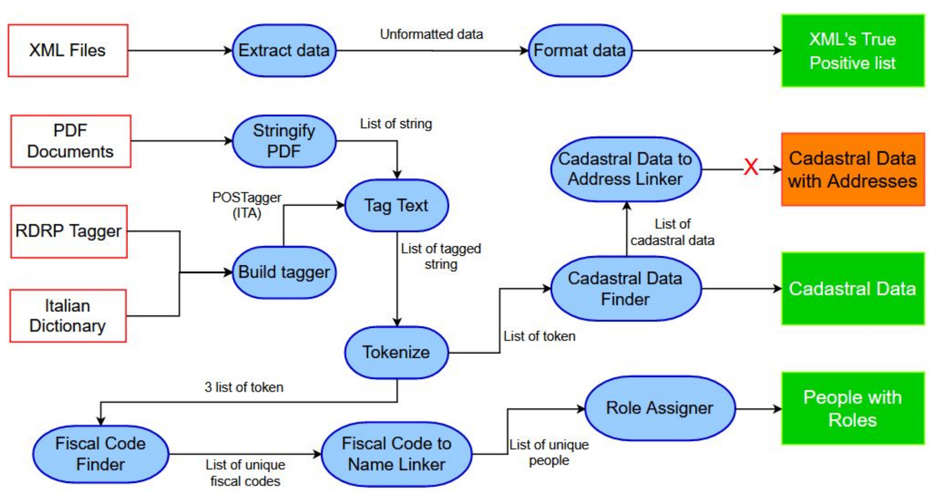

Fig. 1. FrEX architecture.

The RDRP Tagger is a tool developed by Nguyen et al [30] to perform part-of-speech (POS) tagging. It follows a set of rules to label words in a given text; since those rules are inferred via machine learning techniques, the RDRP Tagger can label text in many different languages as long as a proper training set is available. On the RDRP Tagger's web site[12] it is possible to find and download dictionaries for more than 15 different languages and Italian is one of them. We used it to train the tagger for Italian.

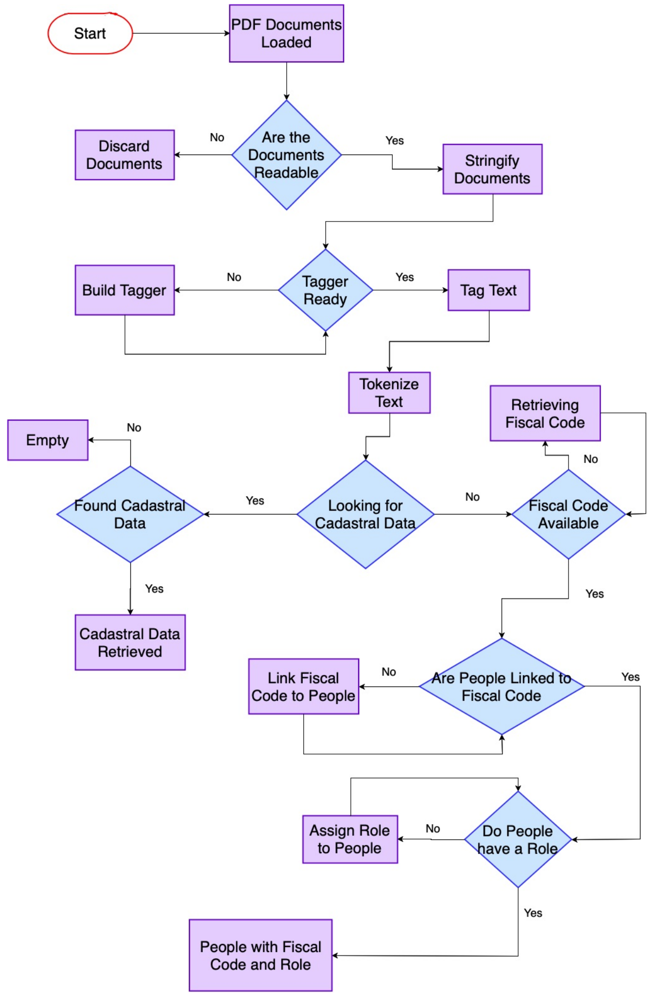

Fig. 2. FrEX workflow.

Below we briefly discuss the implemented filters. The workflow they are involved in, shown in Figure 2, was built for real estate expropriation cases, but as suggested by Software Reuse in Practice [21] it was kept as general as possible, to possibly adapt to other legal cases. The functions implementing them amount to almost 1400 lines of

[12] `http://rdrpostagger.sourceforge.net/`, accessed on September 2020.

Python code, whereas the FrEX main consists of almost 500 lines of code.

Stringify PDF receives in input all `PDF` documents for one case and returns as output an array of array of strings. The output list contains all documents as lists and each cell of the inner list is the string version of the document's phrases. To convert `PDF` text to string we relied on the library called Tika[13].

Tag Text receives in input the string version of the PDF text and a tagger, which is the one built by the Build Tagger filter explained below, and returns as output an array which follows the structure of the string passed as input, but where each word has an associated tag from the Coarse-grained and Fine-grained tags [9] of the ILC/PAROLE tagset [31], compliant with the EAGLES international standard[14].

Tokenize receives in input the tagged version of the PDF text transformed into a string and returns as output three new arrays, each with one word per cell:

- Raw Tokens array – each cell of this array contains one word with no tag;
- Tagged Tokens array – each cell contains one word with the corresponding assigned tag in the form '*[word]/[tag]*';
- SP Tokens array – contains the untagged version of words tagged as SP, which is the 'Proper Noun' tag in the EAGLES standard.

Below we show an example of the Tokenize output:

```
Source: TRIBUNALE ORDINARIO - MILANO NOTA DI ACCOMPAGNAMENTO PER L'ISCRIZIONE A
        RUOLO DI UNA PROCEDURA DI ESPROPRIAZIONE IMMOBILIARE Si chiede [...]
Raw:    ['TRIBUNALE', 'ORDINARIO', 'MILANO', 'NOTA', 'DI', 'ACCOMPAGNAMENTO',
        'PER', 'L'ISCRIZIONE', 'A', 'RUOLO', 'DI', 'UNA', 'PROCEDURA', 'DI',
        'ESPROPRIAZIONE', 'IMMOBILIARE', 'Si', 'chiede']
Tagged: ['TRIBUNALE/S', 'ORDINARIO/A', 'MILANO/SP', 'NOTA/SP', 'DI/E',
        'ACCOMPAGNAMENTO/S', 'PER/E', 'L'ISCRIZIONE/S', 'A/SP', 'RUOLO/S', 'DI/E',
         'UNA/RI', 'PROCEDURA/S', 'DI/E', 'ESPROPRIAZIONE/S', 'IMMOBILIARE/A',
         'Si/PC', 'chiede/V']
SP:     ['MILANO', 'NOTA', 'A']
```

Fiscal Code Finder receives in input an array of tokens and returns a dictionary of unique fiscal codes extracted from those tokens. Italian Fiscal Codes have a peculiar structure that stands out from other words and is built on personal information: some of them are clearly readable, such as the person's year of birth, others require some decoding, like the place the person was born in, while others are guessable at most, such as name and surname. Figure 3 (left) gives an example of how a fiscal code looks like. Understanding if tokens in a list are or are not fiscal codes can be performed in linear time, if the tokens have been correctly parsed, but unfortunately the tokens we worked on were produced from processing PDF documents and the final result was not as clean as if starting from fiscal codes written in ASCII. Most fiscal codes turned out to have been split so a token-by-token checking approach was not possible: we had to merge adjacent tokens in the array until all the fiscal code parts were combined into a

[13] `https://tika.apache.org/`, accessed on September 2020.
[14] `http://www.ilc.cnr.it/EAGLES/browse.html`, accessed on September 2020.

single string again. This merging process starts every time the program finds a word, or a couple of words, that suggest that a fiscal code may appear immediately after, such as ’cf’ or ’fiscal code’. Recognized fiscal codes are stored in a dictionary as keys and their corresponding values are their indexes in the used tokens list. Storing fiscal codes in this way allowed us to ensure their uniqueness.

Fiscal Code to Name Linker receives in input the dictionary of fiscal codes produced by the Fiscal Code Finder and returns as output a list of couples where the first element is the fiscal code and the second is the person’s name and surname, stored as a string. Figure 3 (right) shows this structure.

Role Assigner receives in input the list of couples produced by the Fiscal Code to Name Linker filter and returns a list of couples where the first element is the person’s name and the second is the role w.r.t. the *Real_Estate_Distraint_Frame* semantic frame (debtor, creditor, etc).

Cadastral Data Finder receives in input the list of tokens produced by the Tokenize filter and returns as output a dictionary where keys are the cadastral data in a very technical – but standard for the Italian cadastre – format, namely “Foglio [some integer] Particella [some integer] Subalterno [some integer]”. We choose to use a dictionary to store cadastral data to ensure their uniqueness.

Cadastral Data to Address receives in input the dictionary produced by the Cadastral Data Finder and returns as output another dictionary where keys are cadastral data and values are their addresses. Unfortunately, extracting full and human readable addresses (street, civic number, postal code, municipality) from cadastral data is even harder than associating personal data of people with their fiscal codes: there is little possibility of guessing the civic address from Foglio, Particella, and Subalterno. The only proper way to face the problem would be to query a database that we could not access, and hence the actual dictionary values are lists of tokens that may or may not contains the address. This explains the orange box in Figure 1.

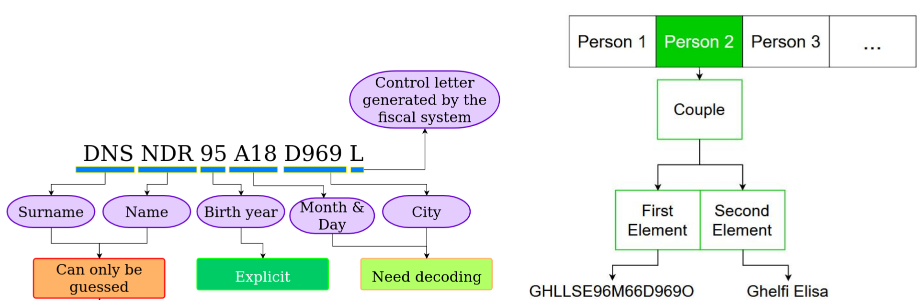

Fig. 3. Fiscal Code example (left) and Fiscal Code to Name Linker filter’s output (right).

Build Tagger receives the RDRP Tagger and a dictionary – in our case the Italian one – and returns a ready-to-use POS Tagger for the dictionary's language.

Extract Data receives in input an `XML` file and returns the lists of frame entities belonging to the *Real_Estate_Distraint_Frame* semantic frame:

- Debtors List - with name and surname of all people whose associated role is debtor;
- Creditors List - with name and surname of all people recognized as creditors;
- Lawyers List - list with name and surname of all people recognized as lawyers;
- Experts List - list with name and surname of all people recognized as experts;
- Judges List - list with name and surname of all people recognized as judges;
- Cadastral List - list of all triples of integer numbers that stands for the cadastral data;

Debtors and creditors may be juridical entities, in which case Extract Data recognizes them as '*people*' and saves them in the proper list along with their data.

Format Data receives in input the six lists generated by Extract Data and returns two lists as output:

- People List – a list with all people listed in some of the input lists, saved as string with the following format: `"[FiscalCode] [Name Surname];[Role;]+||[CompanyCode] [CompanyName TypeOfCompany];[Role;]+"`;
- Cadastral List – a list of cadastral data linked to the real estate property and stored with the following format: `"[Address] - Foglio [int] Particella [int] Subalterno [int]"`;

It is in theory possible that one person or company plays two or more distinct roles in the same case. If this is the case, the second, third, n-th role are attached to the previous one with a ";" in the back.

4 Experiments

In this section we discuss the results obtained by using FrEX for people retrieval, roles recognition and cadastral data retrieval.

FrEX allows users to manually tune some parameters; all the experiments presented here refer to the standard settings, which are those that allow FrEX to achieve the best overall results according to our empirical tests: for people retrieval, "windows" involve 20 tokens; at the first round of search for persons, only proper nouns are considered, but if no result is obtained in this way then the window is enlarged by 5 tokens and common nouns are also considered. For role retrieval, we consider a "window" of 40 tokens, 20 left and 20 right the current word. To verify the correctness of FrEX results we compared its output to those in the `XML` files associated with the 24 cases. Unfortunately, we soon realized that `XML` files were far from being complete, as most people retrieved by FrEX did not appear therein, but – by performing a random manual check – we could verify that they were indeed persons mentioned in the `PDF` documents, often with correctly associated Fiscal Code and Role. To make the evaluation as scientific and reproducible as possible, we still assumed the `XML` files to contain the ‘gold’ labels. We tagged those entities that were not found in the `XML` files as “unknown”, but this does not necessarily mean a wrong retrieval result by FrEX. Given that the only gold labels we could count on are not ‘as gold as expected’, the significance of FrEX’s precision and recall is limited.

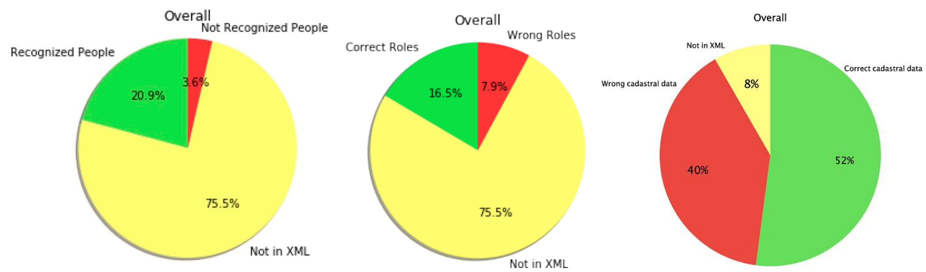

Fig. 4. People – left –, Roles – center –, and Cadastral Data – right – recognition: success (green), failure (red) and not in XML (yellow) rates.

People Retrieval. A person is considered as correctly retrieved by FrEX when his/her fiscal code is found and linked to the proper name and surname. If a fiscal code retrieved by FrEX does not belong to the `XML` file, then that person is considered as “unknown”. A retrieval is correct if the name and surname for a certain fiscal code found in the `XML` files and returned by FrEX are exactly the same, regardless the order. Figure 4, left, shows the FrEX performance on the people retrieval task. As we can see most people are unknown, but the retrieval success rate is higher than the failure rate, that is the most relevant result for our purposes.

In order to identify a person, FrEX searches first his/her fiscal code, then looks at adjacent words labeled as *proper noun* (SP tag) by the POSTagger in a “window” of tokens from the found fiscal code and checks if one of those proper nouns could be the one that generated the three-letter sets contained in a fiscal code and corresponding to name and surname (see Figure 3, left). Limiting the research to words labeled as SP allows FrEX to avoid retrieving wrong words, but the POSTagger is not working perfectly and some names may be tagged incorrectly; if the search fails, then names corresponding to fiscal codes can also be looked for among entities not tagged as proper nouns. This, of course, makes the algorithm much more inefficient, and it is in fact up to the user to switch this feature on or off.

In the FrEX standard settings, besides proper nouns also common noun are looked for. Dino Campana (an Italian poet) would be retrieved even if “Campana” is a common noun meaning “Bell”, but Francesca Neri (an Italian actress) would not, as “Neri” corresponds to the adjective “Black” in plural form, and would be labeled as an adjective and hence ignored during the proper noun search.

Summary of FrEX's performance on people retrieval.

- **Precision =** $\frac{|\{\text{People in XML}\}| \cap |\{\text{People recognized by FrEX}\}|}{|\{\text{People recognized by FrEX}\}|} = 22\%$
- **Recall =** $\frac{|\{\text{People in XML}\}| \cap |\{\text{People recognized by FrEX}\}|}{|\{\text{People in XML}\}|} = 85\%$
- **F-measure =** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 35\%$

Roles Recognition. A role can be considered as properly recognized by FrEX when the profession assigned by the program to a certain person is the same as the profession associated with that very same person in the `XML` files. People not mentioned in `XML` files are labeled as unknown, as before, since we cannot automatically decide whether the role assigned by FrEX was correct or not. This means that results from people retrieval and roles recognition are connected, as explained in the following example where Roberto Salvaneschi is a lawyer:

- if Roberto Salvaneschi is correctly retrieved and FrEX assigns him the role of Lawyer, then we have a hit for both people retrieval and roles recognition;
- if Roberto (without surname) is retrieved and FrEX assigns him the role Lawyer, then we have a miss for both people retrieval and roles recognition since we cannot be sure that the recognized Roberto, is actually Roberto Salvaneschi;
- if Roberto Salvaneschi is retrieved and FrEX assigns him the role of Debtor, then we have a hit for people retrieval and a miss for roles recognition.

To recognize a role we must first retrieve the person so the success rate of roles recognition will never exceed that of people retrieval. Figure 4, center, shows the results obtained by the FrEX on roles recognition. To assign a role to a person, the FrEX picks all words inside a certain "window" from the name of the interested person, then a formula is applied to each word to give them a weight and compute a value for each word. Weight of words depend on their presence inside a list of keywords and on the distance between the person's name/surname and the current word. Keywords are sets of words for each role that are linked to the roles, some of them may be shared among more roles. At the end of the computation all roles are ranked by their values and reordered according to a heuristics. The heuristics takes into account that some roles can easily get higher values than others due to the wide and common use of some of their keywords. For example, if Viviana Mascardi turns out to play both the debtor and creditor roles, and the ranking of those roles are the same, FrEX associates to her the debtor role since she is a person. If instead of Viviana Mascardi we had University of Genova, then FrEX would assign to it the role of creditor because it is an institute/company. More accurate heuristics together with a better and richer definition of keywords for each role would definitely improve the FrEX's role recognition performances.

Summary of FrEX's performance on roles recognition.

- **Precision =** $\frac{|\{\text{Roles in XML}\}| \cap |\{\text{Roles recognized by FrEX}\}|}{|\{\text{Roles recognized by FrEX}\}|} = 18\%$

- **Recall =** $\frac{|\{\text{Roles in XML}\}| \cap |\{\text{Roles recognized by FrEX}\}|}{|\{\text{Roles in XML}\}|} = 68\%$
- **F-measure =** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 28\%$

Cadastral Data Recognition. Cadastral data are considered as recognized by FrEX when the three integer numbers corresponding to Foglio, Particella, and Subalterno are the same as those in the XML files. In our dataset most cases involved only one property so for the majority of the times we dealt with just one cadastral datum. FrEX failure in retrieving cadastral data is usually due to one of the following problems:

- the documents where the cadastral data were mentioned were unreadable;
- cadastral data were never mentioned in any documents;
- the XML file did not include cadastral data.

To improve the results shown in Figure 4, right, we might improve the PDF-to-string transformation rather than the cadastral retrieval process, since – provided that the PDF document has been correctly translated into a textual representation – the structure of cadastral data is so peculiar that they are easily discriminated from other tokens.

Summary of FrEX's performance on cadastral data recognition.

- **Precision =** $\frac{|\{\text{Cad. data in XML}\}| \cap |\{\text{Cad. data recognized by FrEX}\}|}{|\{\text{Cad. data recognized by FrEX}\}|} = 87\%$
- **Recall =** $\frac{|\{\text{Cad. data in XML}\}| \cap |\{\text{Cad. data recognized by FrEX}\}|}{|\{\text{Cad. data in XML}\}|} = 57\%$
- **F-measure =** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 69\%$

FrEX case by case recall. Figure 5 summarizes the **recall** of FrEX. The results are shown case by case (24 cases on the horizontal axis): if all the persons (roles, property data, respectively) that belong to the XML have been correctly identified and the associated data have been correctly retrieved, then we associate 100% recall with the "Persone" light blue column ("Ruoli" green column, "Catast." pink column, respectively). If not all the persons (roles, property data) in the XML have been recognized, the associated recall is lower.

We can easily see that, most often, the retrieval of cadastral data either fully succeeds (cases 2, 4, 6, 8, 10, 11, 12, 13, 14, 17, 18, 20, 23, 24) or fully fails (cases 1, 3, 5, 7, 9, 16, 19, 21, 22). Only in case 15 we have a partial recognition. This is due to the fact that while cases may involve many debtors, creditors, lawyers, judges and experts, and discriminating between some roles (a lawyer and a judge, for example) is not easy for FrEX, leading to variable recall results, there is usually only one property mentioned in each case. Hence, either the property cadastral data are recognized (100% recall) or they are not (0% recall).

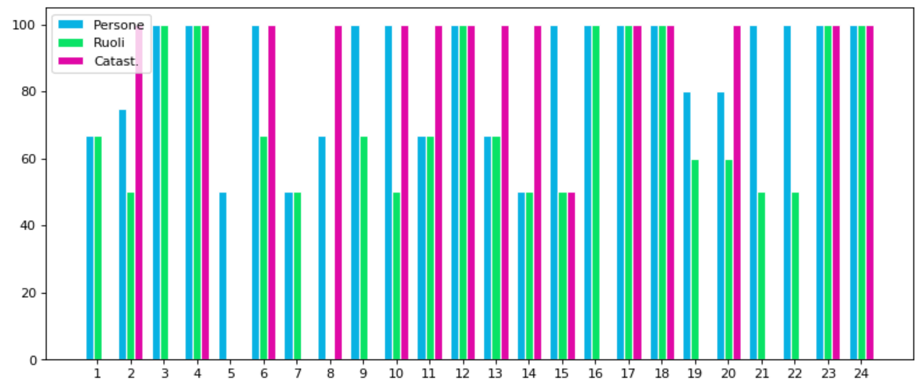

Fig. 5. FrEX recall over the 24 cases for all three tasks.

5 Conclusions and Future Work

We have presented FrEX, a working prototype for extracting frame entities compliant with the *Real_Estate_Distraint_Frame* we designed, from PDF documents in Italian. FrEX has been tested on 24 property expropriation cases, amounting to almost 1000 PDF documents, and its performances in terms of precision, recall and F-measure have been computed by comparing its results with (very imprecise...) existing manual tags for those cases. FrEX overall recall is almost good, with 6 cases out of 24 where all the entities in the `XML` file have been retrieved, and its precision is low but the result may be negatively affected by the gold labels we use, which do not include all the persons/roles actually mentioned in the analyzed documents. An improvement of the experimental results might be obtained by manually creating correct and complete gold labels, that is however a time-consuming activity not foreseen in the very near future, and adding an error analysis. Although more tuning and more tests are required, we believe that FrEX can soon evolve into a tool usable in practice.

The main and most urgent improvement concerns the *Real estate(s) to be distrained* frame entity, that is characterized by its cadastral data only, and is not yet correctly associated with the civic address. Also, the user interface should be better designed, to allow lawyers and magistrates to use FrEX in an intuitive way. We are also evaluating the possibility to boost FrEX performances by taking advantage of existing legal ontologies [4, 13, 35] and – once the property expropriation test case will produce fully satisfactory results – to address other kinds of cases, described by different semantic frames. Finally, in order to address the requirements of eXplainable AI (XAI), we plan to add an explanation function that keeps track of the reason of FrEX choices, and presents them to the user in a human-readable form. This would make professionals from the legal domain more comfortable with the tool, and – hopefully – more keen on trusting AI and accepting its help in their daily activities.

Acknowledgements. We thank Roberto Bichi, President of the Tribunal of Milan, and Marianna Galioto, President of Civil Section III of the Tribunal of Milan, for sharing, under Non-Disclosure Agreement, the set of real documents used for the experiments. In designing the tool presented in this paper, we took advantage of exciting and constructive discussions carried out during the meetings of the DigForASP COST Action, supported by COST (European Cooperation in Science and Technology). We thank the DigForASP participants for their inspiring ideas.

References

1. Agnoloni, T., Barrera, M.F., Sagri, M., Tiscornia, D., Venturi, G.: When a framenet-style knowledge description meets an ontological characterization of fundamental legal concepts. In: Casanovas, P., Pagallo, U., Sartor, G., Ajani, G. (eds.) AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue - International Workshops AICOL-I/IVR-XXIV Beijing, China, September 19, 2009 and AICOL-II/JURIX 2009, Rotterdam,The Netherlands, December 16, 2009 Revised Selected Papers. Lecture Notes in Computer Science, vol. 6237, pp. 93–112. Springer (2009). https://doi.org/10.1007/978-3-642-16524-5_7, https://doi.org/10.1007/978-3-642-16524-5_7
2. Aletras, N., Androutsopoulos, I., Barrett, L., Meyers, A., Preotiuc-Pietro, D. (eds.): Proceedings of the Natural Legal Language Processing Workshop 2020 co-located with the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020), Virtual Workshop, August 24, 2020, CEUR Workshop Proceedings, vol. 2645. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2645
3. Aletras, N., Ash, E., Barrett, L., Chen, D.L., Meyers, A., Preotiuc-Pietro, D., Rosenberg, D., Stent, A. (eds.): Proceedings of the Natural Legal Language Processing Workshop 2019 co-located with the 2019 Annual Conference of the North American Chapter of the Association for ComputationalLinguistics. The Association for Computational Linguistics (2019), https://www.aclweb.org/anthology/W19-22.pdf
4. Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A. (eds.): Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications, vol. 3369 (2005). https://doi.org/10.1007/b106624, https://doi.org/10.1007/b106624
5. Berk, R.: Machine learning risk assessments in criminal justice settings. Springer (2019). https://doi.org/https://doi.org/10.1007/978-3-030-02272-3
6. Bertoldi, A., de Oliveira Chishman, R.L.: Developing a frame-based lexicon for the brazilian legal language: The case of the criminal_process frame. In: Palmirani, M., Pagallo, U., Casanovas, P., Sartor, G. (eds.) AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal Language and Legal Ontologies, Argumentation and Software Agents - International Workshop AICOL-III, Held as Part of the 25th IVR Congress, Frankfurt am Main, Germany, August 15-16, 2011. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7639, pp. 256–270. Springer (2011). https://doi.org/10.1007/978-3-642-35731-2_18, https://doi.org/10.1007/978-3-642-35731-2_18
7. Bertoldi, A., de Oliveira Chishman, R.L.: The limits of using framenet frames to build a legal ontology. In: Vieira, R., Guizzardi, G., Fiorini, S.R. (eds.) Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies, Gramado, Brazil, September 12-14, 2011. CEUR Workshop Proceedings, vol. 776, pp. 207–212. CEUR-WS.org (2011), http://ceur-ws.org/Vol-776/ontobras-most2011_paper26.pdf

8. Butsara, N., Athonthitichot, P., Jodpimai, P.: Predicting recidivism to drug distribution using machine learning techniques. In: 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE). pp. 1–5. IEEE (2019). https://doi.org/https://doi.org/10.1109/ICTKE47035.2019.8966834
9. Calzolari, N., McNaught, J., Zampolli, A.: Tanl POS tagset (2009), http://medialab.di.unipi.it/wiki/Tanl_POS_Tagset
10. Cardellino, C., Alemany, L.A., Villata, S., Cabrio, E.: Improvements in information extraction in legal text by active learning. In: Legal Knowledge and Information Systems - JURIX 2015: The Twenty-Eighth Annual Conference, Braga, Portugal, December 10-11, 2015. Frontiers in Artificial Intelligence and Applications, vol. 279, pp. 21–30. IOS Press (2015). https://doi.org/10.3233/978-1-61499-609-5-21, https://doi.org/10.3233/978-1-61499-609-5-21
11. Cardellino, C., Villata, S., Alemany, L.A., Cabrio, E.: Information extraction with active learning: A case study in legal text. In: Gelbukh, A.F. (ed.) Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9042, pp. 483–494. Springer (2015). https://doi.org/10.1007/978-3-319-18117-2_36, https://doi.org/10.1007/978-3-319-18117-2_36
12. Cimiano, P.: Ontology learning and population from text - algorithms, evaluation and applications. Springer (2006). https://doi.org/10.1007/978-0-387-39252-3, https://doi.org/10.1007/978-0-387-39252-3
13. van Engers, T.M., Boer, A., Breuker, J., Valente, A., Winkels, R.: Ontologies in the legal domain. In: Chen, H., Brandt, L., Gregg, V., Traunmüller, R., Dawes, S.S., Hovy, E.H., Macintosh, A., Larson, C.A. (eds.) Digital Government: E-Government Research, Case Studies, and Implementation, Integrated Series In Information Systems, vol. 17, pp. 233–261. Springer (2008). https://doi.org/10.1007/978-0-387-71611-4_13, https://doi.org/10.1007/978-0-387-71611-4_13
14. Farzindar, A., Lapalme, G.: Legal text summarization by exploration of the thematic structure and argumentative roles. In: Text Summarization Branches Out. pp. 27–34. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://www.aclweb.org/anthology/W04-1006
15. Fillmore, C.J.: Frame semantics and the nature of language. In: Annals of the New York Academy of Sciences. vol. 280, pp. 20–32 (1976). https://doi.org/10.1111/j.1749-6632.1976.tb25467.x
16. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.): Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language, Lecture Notes in Computer Science, vol. 6036. Springer (2010). https://doi.org/10.1007/978-3-642-12837-0, https://doi.org/10.1007/978-3-642-12837-0
17. Galgani, F., Compton, P., Hoffmann, A.G.: Citation based summarisation of legal texts. In: Anthony, P., Ishizuka, M., Lukose, D. (eds.) PRICAI 2012: Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7458, pp. 40–52. Springer (2012). https://doi.org/10.1007/978-3-642-32695-0_6, https://doi.org/10.1007/978-3-642-32695-0_6
18. Haft, F., Jones, R., Wetter, T.: A natural language based legal expert system for consultation and tutoring – the LEX project. In: Proceedings of the 1st International Conference on Artificial Intelligence and Law. pp. 75–83 (1987)
19. II, M.J.B., Katz, D.M., Detterman, E.M.: LexNLP: Natural language processing and information extraction for legal and regulatory texts. CoRR **abs/1806.03688** (2018), http://arxiv.org/abs/1806.03688

20. Kanapala, A., Pal, S., Pamula, R.: Text summarization from legal documents: a survey. Artif. Intell. Rev. **51**(3), 371–402 (2019). https://doi.org/10.1007/s10462-017-9566-2, https://doi.org/10.1007/s10462-017-9566-2
21. Keswani, R., Joshi, S., Jatain, A.: Software reuse in practice. pp. 159–162 (02 2014). https://doi.org/10.1109/ACCT.2014.57
22. Lambiris, M., Oberem, G.: Natural language techniques in computer-assisted legal instruction: a comparison of alternative approaches. J. Legal Educ. **43**, 60 (1993)
23. Lenci, A., Montemagni, S., Pirrelli, V., Venturi, G.: Ontology learning from italian legal texts. In: Breuker, J., Casanovas, P., Klein, M.C.A., Francesconi, E. (eds.) Law, Ontologies and the Semantic Web - Channelling the Legal Information Flood. Frontiers in Artificial Intelligence and Applications, vol. 188, pp. 75–94. IOS Press (2009). https://doi.org/10.3233/978-1-58603-942-4-75, https://doi.org/10.3233/978-1-58603-942-4-75
24. Liu, H.W., Lin, C.F., Chen, Y.J.: Beyond State v Loomis: artificial intelligence, government algorithmization and accountability. International Journal of Law and Information Technology **27**(2), 122–141 (02 2019). https://doi.org/https://doi.org/10.1093/ijlit/eaz001
25. Mascardi, V., Pellegrini, D.: Logical judges challenge human judges on the strange case of B.C.–Valjean. In: Ricca, F., Russo, A., Greco, S., Leone, N., Artikis, A., Friedrich, G., Fodor, P., Kimmig, A., Lisi, F., Maratea, M., Mileo, A., Riguzzi, F. (eds.) Proceedings 36th International Conference on Logic Programming (Technical Communications), UNICAL, Rende (CS), Italy, 18-24th September 2020. Electronic Proceedings in Theoretical Computer Science, vol. 325, pp. 268–275. Open Publishing Association (2020). https://doi.org/10.4204/EPTCS.325.32
26. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the european court of human rights. Artif. Intell. Law **28**(2), 237–266 (2020). https://doi.org/https://doi.org/10.1007/s10506-019-09255-y
27. Meldman, J.A.: A preliminary study in computer-aided legal analysis. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1975), http://hdl.handle.net/1721.1/27423
28. Meunier, R.: The Pipes and Filters Architecture, pp. 427–440. ACM Press/Addison-Wesley Publishing Co., USA (1995)
29. Moens, M.F.: Summarizing court decisions. Information processing & management **43**(6), 1748–1764 (2007)
30. Nguyen, D.Q., Nguyen, D.Q., Pham, D.D., Pham, S.B.: RDRPOSTagger: A ripple down rules-based part-of-speech tagger. In: Bouma, G., Parmentier, Y. (eds.) Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden. pp. 17–20. The Association for Computer Linguistics (2014). https://doi.org/10.3115/v1/e14-2005, https://doi.org/10.3115/v1/e14-2005
31. Paola Baroni, L.C., Enea, A., Montemagni, S., Soria, C., Quochi, V., Carlino, M.: Istituto di linguistica - risorse, http://www.ilc.cnr.it/it/content/risorse
32. Polsley, S., Jhunjhunwala, P., Huang, R.: CaseSummarizer: A system for automated summarization of legal texts. In: Watanabe, H. (ed.) COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, December 11-16, 2016, Osaka, Japan. pp. 258–262. ACL (2016), https://www.aclweb.org/anthology/C16-2054/
33. Robaldo, L., Villata, S., Wyner, A., Grabmair, M.: Introduction for artificial intelligence and law: special issue "natural language processing for legal texts". Artif. Intell. Law **27**(2), 113–115 (2019). https://doi.org/10.1007/s10506-019-09251-2, https://doi.org/10.1007/s10506-019-09251-2

34. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019). https://doi.org/https://doi.org/10.1038/s42256-019-0048-x
35. Sartor, G., Casanovas, P., Biasiotti, M.A., Fernández-Barrera, M. (eds.): Approaches to Legal Ontologies, Law, Governance and Technology Series, vol. 1. Springer (2011). https://doi.org/10.1007/978-94-007-0120-5, `https://doi.org/10.1007/978-94-007-0120-5`
36. Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L.C., Dann, J.: Automated extraction of semantic legal metadata using natural language processing. In: Ruhe, G., Maalej, W., Amyot, D. (eds.) 26th IEEE International Requirements Engineering Conference, RE 2018, Banff, AB, Canada, August 20-24, 2018. pp. 124–135. IEEE Computer Society (2018). https://doi.org/10.1109/RE.2018.00022, `https://doi.org/10.1109/RE.2018.00022`
37. Tolan, S., Miron, M., Gómez, E., Castillo, C.: Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in Catalonia. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 83–92 (2019). https://doi.org/https://doi.org/10.1145/3322640.3326705
38. Turtle, H.: Text retrieval in the legal world. Artificial Intelligence and Law **3**(1-2), 5–54 (1995)
39. Venturi, G., Lenci, A., Montemagni, S., Vecchi, E.M., Sagri, M.T., Tiscornia, D., Agnoloni, T.: Towards a FrameNet resource for the legal domain. pp. 67–76. No. 465 in CEUR (2009)