A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker

Cristian Cardellino
University of Córdoba, Argentina
crscardellino@gmail.com

Milagro Teruel
University of Córdoba, Argentina
milagro.teruel@gmail.com

Laura Alonso Alemany
University of Córdoba, Argentina
alemany@famaf.unc.edu.ar

Serena Villata
Université Côte d'Azur, CNRS, Inria, I3S, France
villata@i3s.unice.fr

ABSTRACT

In this paper we try to improve Information Extraction in legal texts by creating a legal Named Entity Recognizer, Classifier and Linker. With this tool, we can identify relevant parts of texts and connect them to a structured knowledge representation, the LKIF ontology.

More interestingly, this tool has been developed with relatively little effort, by mapping the LKIF ontology to the YAGO ontology and through it, taking advantage of the mentions of entities in the Wikipedia. These mentions are used as manually annotated examples to train the Named Entity Recognizer, Classifier and Linker.

We have evaluated the approach on holdout texts from the Wikipedia and also on a small sample of judgments of the European Court of Human Rights, resulting in a very good performance, i.e., around 80% F-measure for different levels of granularity. We present an extensive error analysis to direct further developments, and we expect that this approach can be successfully ported to other legal subdomains, represented by different ontologies.

CCS CONCEPTS

•Information systems → Ontologies; Information extraction;

KEYWORDS

Legal information extraction, legal ontologies, NER

ACM Reference format:
Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker. In *Proceedings of International Conference on Artificial Intelligence and Law, London, UK, June 12–15, 2017 (ICAIL'17),* 10 pages.
DOI: http://dx.doi.org/10.1145/3086512.XYZ

1 INTRODUCTION

Named Entity Recognition and Classification (NERC) is a cornerstone for Information Extraction (IE). Accurate and specific NERC allows for improved Information Retrieval (IR) and a more informative representation of the contents of documents. It is the basis for the identification and formal representation of propositions, claims and arguments in legal texts, as shown by Surdeanu *et al.* [25].

Information Retrieval and Extraction are key issues in legal practice nowadays, because they allow for an extensive and quick exploitation of jurisprudence. If law practitioners are provided with relevant cases when they are building their arguments for a new case, they are more liable to produce a sounder argumentation. It is also to be expected that cases are resolved more definitely if compelling jurisprudence is provided, even at an early stage in the judicial process. More and more technological solutions are being developed in this line, which shows the feasibility and utility of this line of work. In this context, open-source tools and resources are important also to provide equity to the access of law.

In the legal domain, Named Entities are not only names of people, places or organizations, as in general-purpose NERC. Named Entities are also names of laws, of typified procedures and even of concepts. Named Entities may also be classified differently, for example, countries and organizations are classified as *Legal Person*, as can be seen in the following example extracted from a judgment of the European Court of Human Rights[1]:

Example 1.1. The [Court]$_{organization}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{organization}$, because it was not indicated in the [judgment]$_{abstraction}$ that [Eğitim-Sen]$_{person}$ had carried out [illegal activities]$_{abstraction}$ capable of undermining the unity of the [Republic of Turkey]$_{person}$.

Different levels of granularity can be distinguished in NERC. The most fine-grained level of NERC, Named Entity Linking (NEL) has acquired much attention from the community in recent years, mostly because of the availability of knowledge bases and computational resources that make NEL feasible. The task of NEL consists in determining the identity of entities mentioned in text with respect to a knowledge base. Example 1.1 can be tagged for NEL as follows:

Example 1.2. The [Court]$_{\text{European\_Court\_of\_Human\_Rights}}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{\text{Yargıtay Hukuk Genel Kurulu}}$, because it was not indicated in the [judgment]$_{\text{Court\_of\_Cassation's\_judgment\_of\_22\_May\_2005}}$ that [Eğitim-Sen]$_{\text{Education\_and\_Science\_Workers\_Union\_(Turkey)}}$ had carried out [illegal activities]$_{\emptyset}$ capable of undermining the unity of the [Republic of Turkey]$_{Turkey}$.

[1] Extracted from the case Eğitim ve Bilim Emekçileri Sendikası v. Turkey, ECHR, Second Section, 25 September 2012, http://hudoc.echr.coe.int/eng.

ICAIL'17, London, UK
© 2017 ACM. 978-1-4503-3522-XYZ…$15.00
DOI: http://dx.doi.org/10.1145/3086512.XYZ

In the legal domain, Named Entities are best represented using ontologies. While this is true of any domain, the need for an ontology representing the underlying semantics of Named Entities is crucial in the legal domain, with the severe requirement of precision, a rich hierarchical structure, and well-founded semantics for some of its sub-domains (see, for example, the Hohfeldian analysis of legal rights [18]). Some ontologies have been created to model the legal domain, with different purposes and applied to different sub-domains, e.g., [2, 3, 17]. However, their manual creation and maintenance is a very time-consuming and challenging task: domain-specific information needs to be created by legal experts to ensure the semantics of regulations is fully captured. Therefore, such ontologies have little coverage, because they have a small number of entities or dwell only in abstract concepts. Moreover, only very few annotated legal corpora exist with annotations for entities. All this constitutes an important barrier for Information Extraction from legal text.

In this paper, we tackle this issue by addressing the following research question: *how to populate legal ontologies, with a small number of annotated entities, to support named entity recognition, classification and linking?*

We take a "cheap" approach, by exploiting the information already available in Wikipedia, and connecting it with an ontology of the legal domain. More concretely, we aligned the WordNet- and Wikipedia-based YAGO ontology[2] [24] and the LKIF ontology[3] [17] specifically conceived for representing legal knowledge. By doing this, we are transferring the semantics of LKIF to Wikipedia entities and populating the LKIF ontology with Wikipedia entities and their mentions. At the same time, we obtain a high number of manually annotated examples, taking linked strings in the Wikipedia as examples of entity mentions.

With these examples, we can automatically learn a Named Entity Recognizer, Classifier and Linker. We have applied different approaches, including a customized learner and an off-the-shelf NERC, i.e., the Stanford CRF NERC. Both approaches achieve state-of-the-art performance for a 5-way classification granularity. For finer-grained distinctions, each approach has its own advantages, but both offer good results. For Named Entitiy Linking, the performance needs to be refined but this can be achieved implementing well-known techniques.

We see that, while results on Wikipedia documents are good, there is a drop in performance when we change the domain and apply NERC to judgments of the European Court of Human Rights (ECHR). To deal with this domain change, we have explored the usage of word embeddings, without much improvement. After an analysis of error, we have identified a number of factors that will most probably impact in significant improvements.

The rest of the paper is organized as follows. First, we highlight the main insights of the related literature, and we compare it with the proposed approach. Then, we describe the alignment between YAGO and LKIF, the resulting populated ontology and annotated corpus. In Section 4, we describe different approaches to learn a NERC, and in Section 5 we address the NEL task. We present the methods exploited for the evaluation in Section 6, and we discuss the obtained results in Section 7.

2 RELATED WORK

There exist few of ontologies to represent the legal domain. LRI-Core [8] is intended as a core ontology for law, but it contains very few legal concepts. However, it is thoroughly based on principles of cognitive science, and its top structure is the base of LKIF. The Core Legal Ontology [15] organizes legal concepts and relations on a commonsense basis inspired by DOLCE+ [14]. The LegalRuleML ontology [4] aim to represent machine-readable legal knowledge, with a particular attention to legal sources, time, defeasibility, and deontic operators. Moreover, general-purpose ontologies usually contain some representation of the legal domain, but legal concepts are either not explicitly delimited or very few, or both. We have chosen the LKIF ontology because it is based on previous ontologies and is a well-principled ontology of the legal domain. In the future, we plan to extend our work to LegalRuleML to tackle issues like temporal aspects of norms, violation-reparation, and defeasibility that are not dealt with in LKIF.

In the literature, only few approaches addressed the problem of legal ontology population. More precisely, Bruckschen and colleagues [9] describe an ontology population approach to legal data, whose experimental evaluation is run over a corpus of legal and normative documents for privacy. The goal of this research is to provide a resource that can help software industry project managers to calculate, understand and lower privacy risks in their projects. Ontology population is then obtained through the task of NER. Lenci *et al.* [20] report an experiment on an ontology learning system called T2K. They use NLP and Machine Learning methods to extract terms and relations from free text. The experimental evaluation is conducted on Italian legal texts, and it is able to identify the classes of the ontology, as well as many hyponymy relations. Related approaches to legal ontology population are presented by Boella and colleagues [7, 19]. The former discusses the results of the classification and extraction task of norm elements in European Directives using dependency parsing and semantic role labeling. The experimental system takes advantage of the way the Eunomos system [6] they developed present norms in a structured format. This approach focuses on how to extract prescriptions (i.e., norms) and other concepts (e.g., reason, power, obligation, nested norms) from legislation, and how to automate ontology construction. Similarly, they [7] propose an approach that provides POS tags and syntactic relations as input of a SVM to classify textual instances to be associated to legal concepts. While the approaches in [9, 20] tackle the issue of legal ontology population, they differentiate from our approach regarding many aspects. The main difference with all the above mentioned approaches is the generality of the approach we propose in this paper, that can be easily adapted to any legal ontology and that shows good performance. Moreover, the goal of our approach, i.e., Named Entity Recognition and Entity Linking, and the populated ontologies respectively, are different.

3 ALIGNING YAGO AND LKIF

On the one hand, LKIF [17] is an abstract ontology describing a core of basic legal concepts developed within the EU-funded Estrella Project. It consists of various modules with high-level concepts, and then three modules with law-specific concepts, with a total of 69

[2] www.yago-knowledge.org/

[3] http://www.estrellaproject.org/lkif-core/

law-specific classes. It covers many areas of the law, but it is not populated with concrete real-world entities.

On the other hand, YAGO is a knowledge base automatically extracted from Wikipedia, WordNet, and GeoNames, and linked to the DBpedia ontology[4] and to the SUMO ontology[5]. It represents knowledge of more than 10 million entities, and contains more than 120 million facts about these entities, tagged with their confidence. This information was manually evaluated to be above 95% accurate.

In our alignment process, we do not map relations but only classes. The *manual* alignment is done by mapping a node in one ontology to a node in the other ontology. All children nodes of a connected node are connected by their most immediate parent. Therefore, all children nodes of the aligned YAGO nodes are effectively connected to LKIF through this mapping. The alignment has been addressed by two different persons in parallel, with an agreement phase at the end of the process to decide about controversial mappings, i.e., a concept in one ontology was aligned with two different concepts in the other ontology.

The mapping was carried out using the following methodology: for each LKIF concept, we try to find an equivalent in YAGO. If there is no direct equivalent, then we try to find a subclass, if not, a superclass. When some equivalent concept has been found, we establish the alignment using the OWL primitives `equivalentClass` and `subClassOf`. Finally, we navigate YAGO to visit the related concepts and check whether they could be aligned with another LKIF concept or if they were correctly represented as children of the selected concept.

Because of this methodology, LKIF is effectively the backbone of the resulting ontology, which can be then thought of as an extension of LKIF, including the alignment of the concepts with YAGO ones. This implies that some legal concepts in YAGO are not in our ontology because they were not represented in LKIF. This is the case, for example, of the subdomain of *Procedural Law* or *Crime*, which were two annotate entities in the judgments of the ECHR. We can expect that whenever the ontology is applied to a specific subdomain of the law, it will need to be extended with the relevant concepts.

There are a total of 69 classes in this portion of the LKIF ontology, of which 30 could be mapped to a YAGO node, either as children or as equivalent classes. Two YAGO classes were mapped as parent of an LKIF class, although these we are not exploiting in this approach. 55% of the classes of LKIF could not be mapped to a YAGO node, because they were too abstract (i.e., *Normatively_Qualified*), there was no corresponding YAGO node circumscribed to the legal domain (i.e., *Mandate*), there was no specific YAGO node (i.e., *Mandatory_Precedent*), or the YAGO concept was overlapping but not roughly equivalent (as for "*agreement*" or "*liability*").

From YAGO, 47 classes were mapped to a LKIF class, with a total of 358 classes considering their children, and summing up 4'5 million mentions. However, the number of mentions per class is highly skewed, with only half of YAGO classes having any mention whatsoever in Wikipedia text. Of these 122 populated YAGO classes, only 50 were heavily populated, with more than 10,000 mentions, and 11 had less than 100 mentions. When it comes to particular entities, more than half of the entities had less than 10 mentions in text, only 15% had more than 100 and only 2% had more than 1000. This is a problem for a machine learning approach, since classes with less population cannot be properly learnt by classical methods. Even if for the Named Entity Classification it is not an acute problem, we are planning to apply this approach for Named Entity Linking as well, and then it becomes a serious problem. Moreover, the most populated classes are not core of legal domain, e.g., *company*, *association*.

Level 2 NERC (6 classes, all populated)	**Level 3** LKIF (69 classes, 21 populated)	**Level 4** YAGO (358 classes, 122 populated)
Person	Legal Role ...	judge lawyer ...
Organization	Company Corporation Public Body ...	company limited company corporation foundation court ...
Document	Regulation Contract ...	legal code law contract ...
Abstraction	Legal Doctrine Right ...	case law liberty indebtedness ...
Act	Statute ...	legislative act ..

Figure 1: Levels of abstraction of our legal ontology.

3.1 Level of granularity

The LKIF and YAGO ontologies are very different, and the task of NERC and NEL also differ from each other. In order to assess the performance of the classification at different levels, we established some orthogonal divisions in our ontology, organized hierarchically and effectively establishing different levels of granularity for the NERC and NEL algorithms to work with. Then, we assessed the performance in each level.

The hierarchy of concepts we developed is displayed in Figure 1. We did not use the hierarchy provided by the two ontologies themselves because LKIF, which is our backbone ontology, is not hierarchical, but more aimed to represent interrelations and mereology. YAGO, on the other hand, often presents the multi-parent structure so characteristic of WordNet. The top distinction in our hierarchy is between Named Entities and non-Named Entities, then within Named Entities we distinguish Person, Organization, Document, Abstraction and Act, within those we distinguish LKIF classes and within those we distinguish YAGO classes.

(1) NER (2 classes): The coarsest distinction, it distinguishes NEs from non-NEs.

[4] http://wiki.dbpedia.org/

[5] http://www.adampease.org/OP/

(2) NERC (6 classes): Instances are classified as: Abstraction, Act, Document, Organization, Person or Non-Entity.
(3) LKIF (69 classes, of which 21 have mentions in the Wikipedia): Instances are classified as belonging to an LKIF node.
(4) YAGO (358 classes, of which 122 have mentions in the Wikipedia): Instances are classified as belonging to the most concrete YAGO node possible (except an URI), which can be either child of a LKIF node or an equivalent (but it is never a parent of an LKIF node).
(5) URI (174,913 entities): Entity linking is the most fine-grained distinction, and it is taken care of by a different classifier, described in Section 5.

In Figure 2, we show the Example 1.1 with respect to these different levels of abstraction.

3.2 Wikipedia as a source of annotated examples

Wikipedia has been used as a corpus for NERC because it provides a fair amount of naturally occurring text where entities are manually tagged and linked to an ontology, i.e., the DBpedia [16] ontology. One of the shortcomings of such approach is that not all entity mentions are tagged, but it is a starting point to learn a first version of a NERC tagger, which can then be used to tag further corpora and alleviate the human annotation task.

To build our corpus, we downloaded a XML dump of the English Wikipedia[7] from March 2016, and we processed it via the WikiExtractor [22] to remove all the XML tags and Wikipedia markdown tags, but leaving the links. We extracted all those articles that contained a link to an entity of YAGO that belongs to our mapped ontology. We considered as tagged entities the spans of text that are an anchor for a hyperlink whose URI is one of the mapped entities. We obtained a total of 4,5 million mentions, corresponding to 102,000 unique entities. Then, we extracted sentences that contained at least one mention of a named entity.

We consider the problem of Named Entity Recognition and Classification as a word-based representation, i.e., each word represents a training instance. Then, words within the anchor span belong to the I class (**I**nside a Named Entity), others to the O class (**O**utside a Named Entity). The O class made more than 90% of the instances. This imbalance in the classes results largely biased the classifiers, so we randomly subsampled non-named entity words to make them at most 50% of the corpus. The resulting corpus consists of 21 million words, with words belonging to the O-class already subsampled.

The corpus was divided into three parts: 80% of the corpus for training, 10% for tuning and 10% for testing. The elements on each part were randomly selected to preserve the proportion of each class in the original corpus, with a minimum of one instance of each class appearing in each part of the corpus (training, tuning and testing). We consider only entities with a Wikipedia page and with more than 3 mentions in Wikipedia.

4 LEARNING A NERC

Using the corpus described in the previous section, we trained a classifier for Named Entity Recognition and Classification. The objective of this classifier is to identify in naturally occurring text mentions the Named Entities belonging to the classes of the ontology, and classify them in the corresponding class, at different levels of granularity. Note that we do not consider here the URI level, which is treated qualitatively differently by the Named Entity Linking approach, and we will detail it in Section 5.

4.1 Learners

We have applied different approaches to exploit our annotated examples. First of all, we have trained a linear classifier, namely a Support Vector Machine (SVM) with a linear kernel, and the Stanford CRF Classifier model for NERC [23], with our corpus with Wikipedia annotations for the LKIF classes. Decision trees and Naive Bayes (NB) classifiers were discarded because the cardinality of the classes was too large for those methods. The Stanford NERC could not handle the level of granularity with most classes, the YAGO level.

Moreover, we have learnt a neural network, carrying out experiments with one, two and three hidden layers, but it resulted that a single hidden layer, smaller than the input layer, performed better, so we set this architecture. We have explored more complex configurations of the neural network, including Curriculum Learning [5], a learning strategy that is specially adequate for hierarchically structured problems like ours, with subsequent levels of granularity. However, none of these more complex configurations improved performance. For more details about the use of Curriculum Learning in our NERC, we refer the reader to [11].

4.2 Representation of examples

We represented examples with a subset of the features proposed by Finkel *et al.* [13] for the Stanford Parser CRF-model. For each instance (i.e., each word), we used: current word, current word PoS-tag, all the n-grams ($1 \leq n \leq 6$) of characters forming the prefixes and suffixes of the word, the previous and next word, the bag of words (up to 4) at left and right, the tags of the surrounding sequence with a symmetric window of 2 words, and the occurrence of a word in a full or part of a gazetteer. The final vector characterizing each instance has more than 1.5e6 features, too large to be handled due to memory limitations. In addition, the matrix was largely sparse. As a solution, we applied a simple feature selection technique using Variance Threshold. We filtered out all features with variance less than 2e-4, reducing the amount of features to 11997. For the Stanford NERC, we used the same features as the MLP classifiers, except the presence in gazetteers and the PoS tags of surrounding words.

The experiments were also carried out using word embeddings. We originally did some exploration using the Google News corpus pre-trained embeddings,[8] which are 3 million dense word vectors of dimension 300, trained on a 100 billion words corpus. However, we decided to go with some embeddings trained by ourselves using Word2Vec's skip-gram algorithm, based solely in the Wikipedia corpus we later use for the NERC task. All words with less than 5 occurrences were filtered out, leaving roughly 2.5 million unique tokens (meaning that a capitalized word is treated differently than an all lower case word), from a corpus of 1 billion raw words. The trained embeddings were of size 200, and taking them we generate a matrix where each instance is represented by the vector of the instance word surrounded by a symmetric window of 3 words at each size. Thus, the input vector of the network is of dimension

[7] https://dumps.wikimedia.org/

[8] https://code.google.com/archive/p/word2vec/

NER

The [Court] is not convinced by the reasoning of the [combined divisions of the Court of Cassation], because it was not indicated in the [judgment] that [Eğitim-Sen] had carried out [illegal activities] capable of undermining the unity of the [Republic of Turkey].

NERC

The [Court]$_{organization}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{organization}$, because it was not indicated in the [judgment]$_{abstraction}$ that [Eğitim-Sen]$_{person}$ had carried out [illegal activities]$_{abstraction}$ capable of undermining the unity of the [Republic of Turkey]$_{person}$.

LKIF

The [Court]$_{PublicBody}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{PublicBody}$, because it was not indicated in the [judgment]$_{Decision}$ that [Eğitim-Sen]$_{LegalPerson}$ had carried out [illegal activities]$_{Crime}$[6] capable of undermining the unity of the [Republic of Turkey]$_{LegalPerson}$.

YAGO

The [Court]$_{\text{wordnet\_trial\_court\_108336490}}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{\text{wordnet\_trial\_court\_108336490}}$, because it was not indicated in the [judgment]$_{\text{wordnet\_judgment\_101187810}}$ that [Eğitim-Sen]$_{\text{wordnet\_union\_108233056}}$ had carried out [illegal activities]$_{\text{wordnet\_illegality\_104810327}}$ capable of undermining the unity of the [Republic of Turkey]$_{person}$.

Figure 2: An example of legal entities annotated at different levels of granularity.

1400 as it holds the vectors of a 7 word window total. If the word was near the beginning or the end of a sentence, the vector is padded with zeros. We also pad with zeros in case no representation of the word (capitalized or not) is found in the Word2Vec model.

Word embeddings are known to be particularly apt for domain transfer, because they provide some smoothing over the obtained model, preventing overfitting to the training set. Therefore, we expect them to be useful to transfer the models obtained from Wikipedia to other corpora, like the judgments of the ECHR.

However, it is also known that embeddings are more adequate the bigger the corpus they are learnt from, and if the corpus belongs to the same domain to which it will be applied. In our case, we have a very big corpus, namely Wikipedia, that does not belong to the domain to which we want to apply the embeddings, namely the judgments. Therefore, we have experimented with three kinds of embeddings: embeddings obtained from Wikipedia alone (as described above), those obtained with the same methodology but from the judgments alone, and those obtained with a mixed corpus made of judgments of the ECHR, and a similar quantity of text from Wikipedia. To train word embeddings for judgments of the ECHR, we obtained all cases in English from the ECHR's official site available on November 2016, leading to a total of 10,735 documents.

5 DEVELOPING A NAMED ENTITY LINKER

The Named Entity Linking task consists in assigning YAGO URIs to the Wikipedia mentions, as shown in Example 1.2. The total number of entities found in the selected documents is too big (174,913) to train a classifier directly. To overcome this problem, we use a two-step classification pipeline. Using the NERC provided by the previous step, we first classify each mention as its most specific class in our ontology. For each of these classes, we train a classifier to identify the correct YAGO URI for the instance using only the URIs belonging to the given class. Therefore, we build several classifiers, each of them trained with a reduced number of labels. Note that each classifier is trained using only entity mentions for a total of 48,353 classes, excluding the 'O' class.

The state of the art tool for NEL is Babelfy[9], but we could not compare to it comparison because it has a daily limit of 1000 queries.

The algorithm to train the two-step pipeline is provided in Figure 3, while the algorithm for classification is described in Figure 4.

```
1. Assign to each mention its ground truth
   ontology label.
2. Split the dataset into train/test/validation.
3. For each assigned ontology class:
   3.1. Build new train/test/validation
        datasets by filtering out mentions
        not tagged with this class.
   3.2. Train and evaluate a classifier with
        the new train/test/validation datasets.
```

Figure 3: Algorithm to train a NEL in two steps.

```
1. For each instance, assign a NE class to
   it using a previously trained NERC.
2. Select the classifier assigned to the
   class, and use it to obtain a YAGO uri
   prediction of the instance.
```

Figure 4: Algorithm to classify an unseen entity using the NEL.

The classifiers learnt for each of the classes were Neural Network classifiers with a single hidden layer, of size 2*number of classes with a minimum of 10 and a maximum of 500. Other classifiers cannot handle the high number of classes in this setting, in particular, the Stanford NERC is incapable of handling them.

As a comparison ground, we also evaluated two baselines, a random classifier and a k-nearest neighbors. For the random baseline, given the LKIF class for the entity (either ground truth or assigned by an automated NERC), the final label is chosen randomly among the

[9] http://babelfy.org/

YAGO URIs seen for that LKIF class in the training set, weighted by their frequency. The k-nearest neighbors classifier is trained using the current, previous and following word tokens, which is equivalent to checking the overlap of the terms in the entity.

We distinguish two types of evaluations: the performance of each classifier, using ground truth ontology classes, and the performance of the complete pipeline, accumulating error from automated NERC. The individual classifier performance is not related to the other classifiers, and is affected only by the YAGO URIs in the same LKIF class. It is calculated using the test set associated with each class, that does not include the 'O' class.

6 EVALUATION

To evaluate the performance, we computed accuracy, precision and recall in a word-to-word basis in the test portion of our Wikipedia corpus, totalling 2 million words of which the half belong to NEs and the other half to non-NEs. Thus, the evaluation consisted on calculating the proportion of words that had been correctly or incorrectly tagged as part of a NE and as belonging to a class of NEs at different levels of granularity.

For this particular problem, accuracy does not throw much light upon the performance of the classifier because the performance for the majority class, *non-NE*, eclipses the performance for the rest. To have a better insight on the performance, the metrics of precision and recall are more adequate. We calculated those metrics per class, and we provide a simple average *without the non-NE class*. Besides not being obscured by the huge *non-NE* class, this average is not weighted by the population of the class (thus an equivalent of macro-average). Therefore, the differences in these metrics are then showing differences in all classes, with less populated classes in equal footage with more populated ones.

Additionally, we discriminate the performance of some classifiers in the 20% most populated classes and in the 20% least populated classes, to have a global view of the errors. We also show the confusion matrix of classification (Figure 8), casting classes into bins according to their frequency to enable results to be displayed. This evaluation shows how errors are distributed, in order to address further developments in the right direction.

6.1 Evaluation on a corpus of judgments

Evaluating on Wikipedia has the advantage that NERC and NEL models have been learnt with Wikipedia itself, so they are working on comparable corpora. However, even if it is useful to detect NEs in the Wikipedia itself, it is far more useful for the community to detect NEs in legal corpora like norms or case-law. That is why we have manually annotated a corpus of judgments of the European Court of Human Rights, identifying NEs that belong to classes in our ontology or to comparable classes that might be added to the ontology. This annotated corpus is useful to evaluate the performance of the developed NERC and NEL tools, but it will also be used to train specific NERC and NEL models that might be combined with Wikipedia ones.

More precisely, we annotated excerpts from 5 judgments of the ECHR, obtained from the Court website[10] and totalling 19,000 words. We identified 1,500 entities, totalling 3,650 words. Annotators followed specific guidelines, inspired in the LDC guidelines for annotation of NEs [21]. Annotators were instructed to classify NEs at YAGO and URI levels, but no consistent annotation guidelines could be developed for the URI level, which is equivalent to Named Entity Linking, thus it has not been used for evaluation yet.

There were 4 different annotators, and three judgments were annotated by at least 2 annotators independently, to assess inter-annotator agreement using Cohen's kappa coefficient [12]. The agreement between judges ranged from $\kappa = .4$ to $\kappa = .61$, without significant differences across levels of granularity. Most of the disagreement between annotators was found for the recognition of NEs, not for their classification. The classes and subclasses of *Document*, *Organization* and *Person* were the most consistent across annotators, while *Act*, *Abstraction* and *non-NE* accumulated most discrepancies.

The inter-annotator agreement obtained for this annotation is not high, and does not guarantee reproducible results. We are planning to improve annotation guidelines, including discussion sessions to unify criteria. Then, a more reliable version of these annotations will be produced, useful for evaluation, and more importantly, to train domain-specific NERC and NEL. For the time being, these annotations can be used for evaluation to obtain results that are indicative of the performance of the tools on legal text.

approach	accuracy	precision	recall	F1
NER (2 classes)				
SVM	1.00	.54	.06	.11
Stanford NER	.88	.87	.87	.87
NN	1.00	**1.00**	**1.00**	1.00
NN+WordEmb	.95	.95	.95	.95
NERC (6 classes)				
SVM	.97	.37	.18	.24
Stanford NER	.88	.78	.82	.79
NN	.99	**.89**	**.83**	**.86**
NN+WordEmb	.94	.84	.78	.81
LKIF (21 classes)				
SVM	.93	.53	.26	.35
Stanford NER	.97	**.84**	**.71**	**.77**
NN	.97	.73	.65	.69
NN+WordEmb	.93	.67	.60	.63
YAGO (122 classes)				
SVM	.89	.51	.25	.34
Stanford NER	–	–	–	–
NN	.95	**.76**	**.64**	**.69**
NN+WordEmb	.90	.68	.61	.64

Table 1: Results for Named Entity Recognition and Classification on the test portion of the Wikipedia corpus, for different approaches, at different levels of granularity. Accuracy figures take into consideration the majority class of non-NEs, but precision and recall are an average of all classes (macro-average) except the majority class of non-NEs.

[10] hudoc.echr.coe.int

7 ANALYSIS OF THE RESULTS

In this section, we describe and analyze the results of different approaches to NERC and NEL in the Wikipedia and in the corpus of annotated judgments of the ECHR.

7.1 NERC results on Wikipedia

The results for NERC on the test portion of our Wikipedia corpus at different levels of abstraction are reported in Table 1. We show the overall accuracy (taking into consideration the 'O' class), and the average recall, precision and F-measure across classes other than the non-NE class. The Stanford NERC could not deal with the number of classes in the YAGO level, so it was not evaluated in that level. A summary of that information is provided in Figure 5, displaying accuracy and F-measure of the different approaches at different levels of granularity. We also show results with handcrafted features and with word embeddings obtained from the Wikipedia.

At bird's eye view, it can be seen that the SVM classifier performs far worse than the rest, and also that word embeddings consistently worsen the performance of the Neural Network classifier. The Stanford NERC performs worse than the Neural Network classifier at the NER level, but they perform indistinguishably at NERC level and Stanford performs better at LKIF level. However, it can be observed that the Neural Network performs better at the YAGO level than at the LKIF level, even though there are 122 classes at the YAGO level vs. 21 classes at LKIF level.

If we take a closer look at performance, we can see in Figure 7 that the Neural Network classifier performs far better in smaller classes (with less instances) than in bigger classes, for all levels of abstraction but most dramatically for the LKIF level, where F-score for the 20% biggest classes drops to .11 (in contrast with .62 for NERC and .42 for YAGO), while for the smallest classes it keeps within the smooth decrease of performance that can be expected from the increase in the number of classes, and thus an increase in the difficulty of classification.

These results corroborate an observation that has already been anticipated in general results, namely, that the LKIF level of generalization is not adequate for automated NERC, and that the NERC cannot distinguish the classes defined at that level, that is, in the original LKIF ontology. In contrast, the NERC does a better job at distinguishing YAGO classes, even if the classification problem is more difficult because of the bigger number of classes.

On the other hand, the fact that smaller classes are recognized better than bigger classes indicates that bigger classes are ill-delimited. It may be that these classes are built as catch-all classes, grouping heterogeneous subclasses. Therefore, it seems that the chosen level of granularity for legal NERC using our ontology should be the most fine-grained, because it provides most information without a significant loss in performance, or even with a gain in performance for the most populated classes. Another possibility to improve the performance at LKIF level would be to revisit the alignment, which is in our plans for the near future, but we do not expect to have important changes in that aspect.

We also explored the confusion matrix of classification (Figure 8), to obtain a more qualitative insight on the errors of the classifiers. In the first place, we can see that there is barely no confusion between non-NEs (the 'O' class) and the rest of classes. In the least fine-grained level, NERC, most of the confusion is between classes *Document* and *Act*. This can be explained by observing that, in the legal domain, acts are often materialized as documents, e.g., in declarations, prohibitions. However, the high rate of confusion indicates that this distinction is inadequate for the task.

In the other levels, most of the confusion is found between the most frequent classes, as it may be noticed in the confusion matrix for the YAGO level, where many words that belong to smaller classes are classified as belonging to bigger classes. This can also be found at LKIF level: we see that many instances of smaller classes are classified as belonging to the most populated class, *Company*. To address this bias, we will train our classifier with attention to class imbalance, forcing balance by reducing the number of instances of bigger classes. This bias will also be taken into account for the annotation of judgments of the ECHR.

7.2 NERC results on the judgments of the ECHR

The results for NERC in the corpus of judgments of the ECHR described in Section 6.1 are shown in Table 2 and in Figure 6. We can see the results with the models trained on Wikipedia and applied to the ECHR documents, and with models trained with and applied to the ECHR corpus (divided in training and test splits). We can also see models working on different representations of examples, as described in Section 4.2. The variations are handcrafted features and different combinations of embeddings: obtained from Wikipedia alone, obtained from the judgments of the ECHR alone, and obtained from Wikipedia and the ECHR in equal parts.

We can see that, on the ECHR corpus, results obtained for models trained with the annotated corpus of ECHR judgments perform significantly better than those trained with Wikipedia, even if the latter are obtained with a much bigger corpus. The differences in performance can be seen more clearly in the F-measure plot in Figure 6 (right). This drop in performance is mainly due to the fact that the variability of entities and the way they are mentioned is far smaller in the ECHR than in Wikipedia. There are fewer unique entities and some of them are repeated very often (e.g., "Court", "applicant") or in very predictable ways (e.g., cites of cases as jurisprudence).

For models trained with the annotated corpus of ECHR judgments, word embeddings decrease performance. This results are mainly explainable because of overfitting: word embeddings prevent overfitting, and are beneficial specially in the cases of very variable data or domain change, which is not the case when the NERC is trained with the ECHR corpus, with very little variability.

We also highlight that there is little difference between word embeddings trained with different inputs, although Wikipedia-trained word embeddings present better performance in general. There is no consistent difference between mixed and ECHR trained embeddings. In contrast, in Wikipedia-trained models, ECHR and mixed (ECHR+Wikipedia) word embeddings improve both precision and recall. This shows that, when we have a domain-specific model, embeddings obtained from a significantly bigger corpus are more beneficial. However, when no in-domain information is available, a representation obtained from many unlabeled examples yields a bigger improvement. For a lengthier discussion of these results, see Teruel and Cardellino (2017) [10].

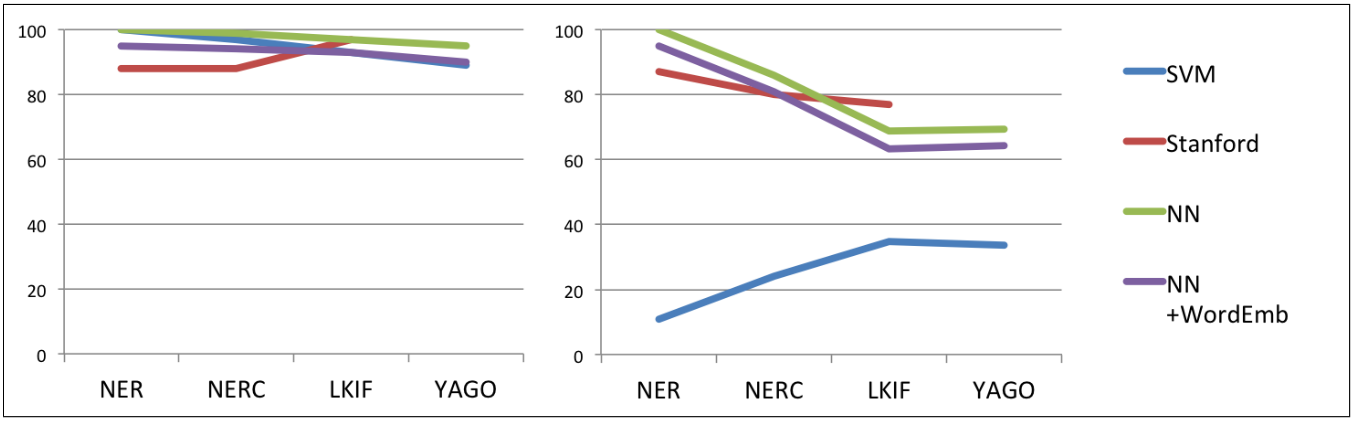

Figure 5: Results of different approaches to NERC on the Wikipedia test corpus, at different levels of granularity, with accuracy (left) and F-measure (right), as displayed in Table 1.

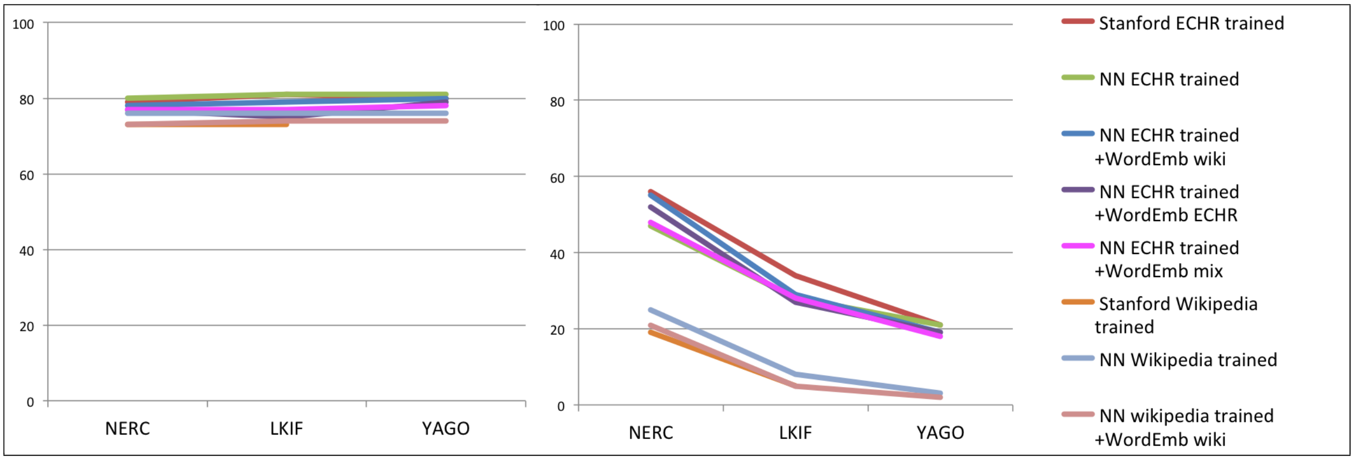

Figure 6: Results of different approaches to NERC on the judgments of the ECHR, at different levels of granularity, with accuracy (left) and F-measure (right), as displayed in Table 2. Approaches with different embeddings are distinguished.

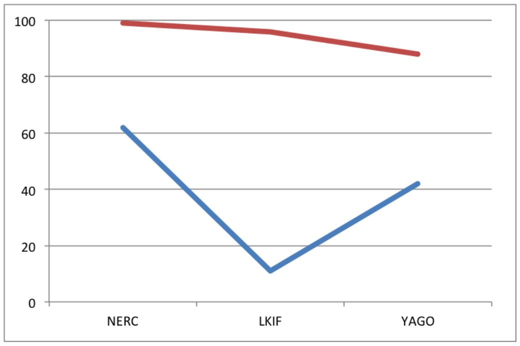

Figure 7: F-measure of the Neural Network classifier for NERC at different levels of granularity, discriminating the 20% most populated classes (blue) and 20% least populated classes (red).

7.3 NEL results on Wikipedia

As explained in Section 6.1, NEL could not be evaluated on the corpus of judgments, but only on Wikipedia, because annotation at the level of entities has not been consolidated in the corpus of judgments of the ECHR. Therefore, approaches to NEL have only been evaluated on the test portion of the corpus of Wikipedia.

Results are shown in Table 3. As could be expected from the results for NERC, word embeddings worsened the performance of prediction. We can see that the performance of NEL is quite acceptable if it is applied on ground-truth labels, but it only reaches a 16% F-measure if applied over automatic NERC at the YAGO level of classification. Thus, the fully automated pipeline for NEL is far from satisfactory. Nevertheless, we expect that improvements in YAGO-level classification will have a big impact on NEL.

We also plan to substitute the word-based representation of NEs by a string-based representation that allows for better string overlap heuristics and a customized edit distance for abbreviation heuristics.

		NERC (6 classes)				LKIF (21 classes)				YAGO (122 classes)			
		Acc.	Prec.	Recall	F-score	Acc.	Prec.	Recall	F-score	Acc.	Prec.	Recall	F-score
Wiki trained	NN	**.76**	**.56**	**.24**	**.25**	**.76**	**.13**	**.07**	**.08**	**.76**	**.06**	**.03**	**.03**
	NN+WordEmb wiki	.73	.34	.21	.21	.74	.08	.05	.05	.74	.03	.02	.02
	NN+WordEmb mix	.75	.42	.23	.23	.75	.10	.06	.06	.75	.04	.04	.03
	NN+WordEmb echr	.75	.38	.24	.24	.75	.11	.07	.07	.74	.04	.03	.03
	Stanford	.73	.36	.17	.16	.73	.07	.06	.05	-	-	-	-
ECHR trained	NN	**.80**	**.69**	.41	.47	**.81**	.46	.24	.28	**.81**	**.33**	.18	**.21**
	NN+WordEmb echr	.77	.52	.54	.52	.75	.27	.32	.27	.79	.22	.22	.19
	NN+WordEmb wiki	.78	.54	**.58**	.55	.79	.30	**.34**	.29	.80	.24	.22	.19
	NN+WordEmb mix	.77	.48	.50	.48	.77	.28	.32	.28	.78	.23	**.22**	.18
	Stanford	.79	.67	.51	**.56**	**.81**	**.49**	.30	**.34**	.80	.28	.21	**.21**

Table 2: Results for Named Entity Recognition and Classification on the corpus of judgments of the ECHR, for different approaches, at different levels of granularity, with models trained only with the documents of the ECHR themselves (divided in training and test) and with models trained with the Wikipedia, combined with embeddings obtained from the Wikipedia, from the ECHR or from both. Accuracy figures take into consideration the majority class of non-NEs, but precision and recall are an average of all classes (macro-average) except the majority class of non-NEs.

approach	accuracy	precision	recall	F1
NEL on ground truth				
NN	.94	.48	.45	.45
NN+word embeddings	.72	.25	.25	.25
NEL on automatic YAGO-level NERC				
NN	.69	.18	.15	.16
baselines				
Random	.51	.00	.00	.00
K-nn	.71	.14	.10	.10

Table 3: Results for Named Entity Linking on the test portion of the Wikipedia corpus, for different approaches, including random and K-nn baselines.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach to develop a Named Entity Recognizer, Classifier and Linker exploiting Wikipedia. The resulting tools and resources are open-source and freely available to anyone in the community, but, more importantly, this approach can be reproduced for any legal subdomain of interest.

We have created an alignment between the Wikipedia-based ontology YAGO and a well-established ontology for the legal domain, LKIF. Through this alignment, we have delimited the domain of legal entities that we are targeting, and we have obtained all mentions of those entities in Wikipedia. Mentions are then used as manually annotated examples to train a Named Entity Recognizer, Classifier and Linker. We have established four levels of granularity for the classification of the entities.

We have trained different kinds of classifiers and evaluated them on Wikipedia and on the manually annotated judgments of the European Court of Human Rights. A Neural Network classifier and the Stanford CRF NERC achieve state-of-the-art performance at the standard 5-way classification problem (with classes Person, Organization, Document, Abstraction, Act). For finer-grained distinctions, the Stanford NERC obtains slightly better performance but the Neural Network classifier can deal with a bigger number of classes.

We have also seen that the classes defined by the LKIF ontology are hard to capture using Wikipedia examples, probably because the conceptualization is very different. Although overall performance is not affected, we have seen that bigger classes (populated with more mentions in Wikipedia text) accumulate most of the error. We will be addressing this problem applying methods to balance classes for the learner and also by reconsidering the alignment using error analysis. We expect this will produce an improvement in performance that will also impact at the other levels of granularity.

As they are, the resources we have created are useful to pre-annotate the legal domain articles of Wikipedia, for example, in synergy with the WikiProject Law [1], which aims to better organize information in Wikipedia articles related to the law domain. We are also planning to use the NERC and NEL to speed up the manual annotations of the judgments of the ECHR. Then, from these annotations, we expect to obtain new mentions and entities to populate our legal ontology.

ACKNOWLEDGMENTS

The authors have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690974 for the project MIREL: MIning and REasoning with Legal texts.

REFERENCES

[1] 2016. WikiProject Law. https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Law. (2016).

[2] Gianmaria Ajani, Guido Boella, Luigi Di Caro, Livio Robaldo, Llio Humphreys, Sabrina Praduroux, Piercarlo Rossi, and Andrea Violato. 2016. The European Taxonomy Syllabus: A multi-lingual, multi-level ontology framework to untangle the web of European legal terminology. *Applied Ontology* 11, 4 (2016), 325–375.

[3] Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2015. LegalRuleML: Design Principles and Foundations. In *The 11th Reasoning Web Summer School*, Wolfgang Faber and Adrian Pashke (Ed.). Springer, Berlin, Germany, 151–188.

[4] Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Z. Wyner. 2015. LegalRuleML: Design Principles and Foundations. In *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany,*

July 31 - August 4, 2015, Tutorial Lectures (Lecture Notes in Computer Science), Wolfgang Faber and Adrian Paschke (Eds.), Vol. 9203. Springer, 151–188.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 41–48.

[6] Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, Piercarlo Rossi, and Leon van der Torre. 2016. Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law. *Artif. Intell. Law* 24, 3 (2016), 245–283.

[7] Guido Boella, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. 2014. Learning from syntax generalizations for automatic semantic annotation. *J. Intell. Inf. Syst.* 43, 2 (2014), 231–246.

[8] Joost Breuker and Rinke Hoekstra. 2004. Epistemology and Ontology in Core Ontologies: FOLaw and LRI-Core, two core ontologies for law. In *In Proceedings of the EKAW04 Workshop on Core Ontologies in Ontology Engineering*. Northamptonshire, UK, 15–27.

[9] Mírian Bruckschen, Caio Northfleet, Douglas da Silva, Paulo Bridi, Roger Granada, Renata Vieira, Prasad Rao, and Tomas Sander. 2010. Named entity recognition in the legal domain for ontology population. In *3rd Workshop on Semantic Processing of Legal Texts (SPLeT 2010)*.

[10] Cristian Cardellino and Milagro Teruel. 2017. In-domain or out-domain word embeddings? A study for Legal Cases. In *Student Session of the European Summer School for Logic, Language and Information (ESSLLI 2017)*.

[11] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal NERC with ontologies, Wikipedia and curriculum learning. In *The European Chapter of the Association for Computational Linguistics (EACL-2017)*. ACL, 254–259.

[12] J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational & Psycological Measure* 20 (1960), 37–46.

[13] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 363–370.

[14] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. 2002. Sweetening Ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web (EKAW '02)*. Springer-Verlag, London, UK, UK, 166–181. http://dl.acm.org/citation.cfm?id=645362.650863

[15] Aldo Gangemi, Maria-Teresa Sagri, and Daniela Tiscornia. 2005. Law and the Semantic Web. Springer-Verlag, Berlin, Heidelberg, Chapter A Constructive Framework for Legal Ontologies, 97–124. http://dl.acm.org/citation.cfm?id=2168120.2168129

[16] Younggyun Hahm, Jungyeul Park, Kyungtae Lim, Youngsik Kim, Dosam Hwang, and Key-Sun Choi. 2014. Named Entity Corpus Construction using Wikipedia and DBpedia Ontology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (26-31), Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), Reykjavik, Iceland.

[17] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer. 2007. The LKIF Core ontology of basic legal concepts. In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*.

[18] W. Hohfeld. 1919. *Fundamental Legal Conceptions*. Yale University Press.

[19] Llio Humphreys, Guido Boella, Livio Robaldo, Luigi di Caro, Loredana Cupi, Sepideh Ghanavati, Robert Muthuri, and Leendert van der Torre. 2015. Classifying and Extracting Elements of Norms for Ontology Population using Semantic Role Labelling. In *Proceedings of the Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts*.

[20] A. Lenci, S. Montemagni, V. Pirrelli, and G. Venturi. 2009. Ontology learning from Italian legal texts. In *Proceeding of the 2009 Conference on Law, ontologies and the Semantic Web: Channelling the Legal information Flood*.

[21] Linguistic Data Consortium. 2014. DEFT ERE Annotation Guidelines: Entities V1.7. http://nlp.cs.rpi.edu/kbp/2014/ereentity.pdf. (2014).

[22] Medialab University of Pisa. 2015. WikiExtractor. http://medialab.di.unipi.it/wiki/Wikipedia_Extractor. (2015).

[23] Stanford NLP Group. 2016. Stanford Named Entity Recognizer (NER). http://nlp.stanford.edu/software/CRF-NER.shtml. (2016).

[24] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 697–706.

[25] Mihai Surdeanu, Ramesh Nallapati, and Christopher D. Manning. 2010. Legal Claim Identification: Information Extraction with Hierarchically Labeled Data. In *Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts (SPLeT-2010)*. Malta.

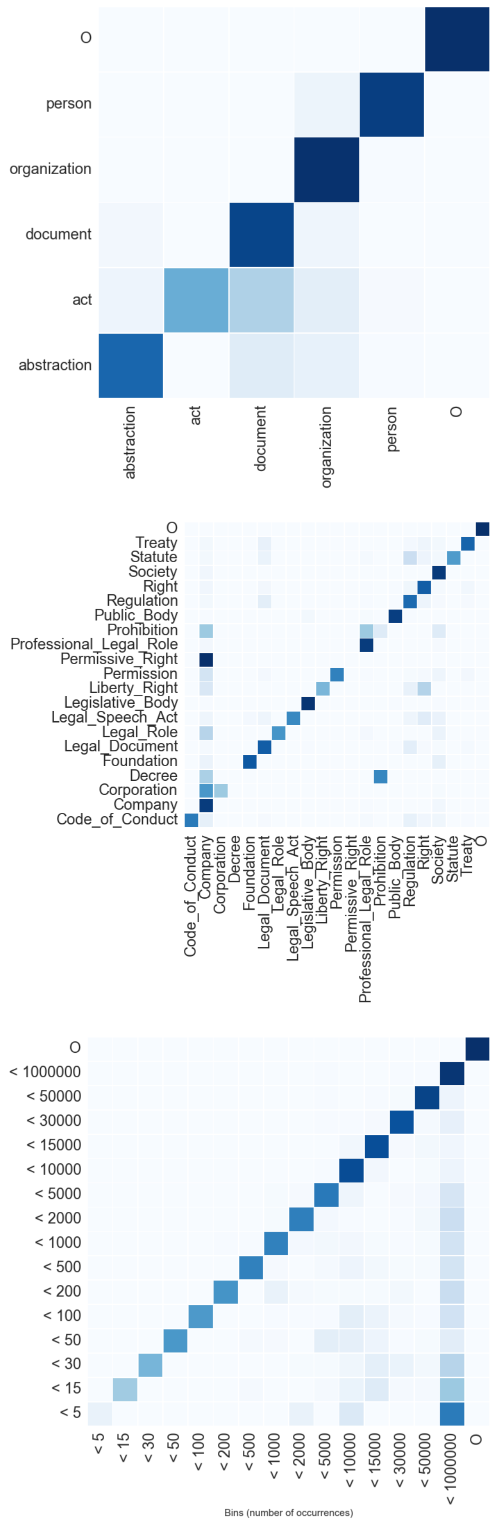

Figure 8: Confusion matrix of classification by Neural Networks with handcrafted features in different levels of granularity: NERC (top), LKIF (middle) and YAGO (bottom).