In-domain or out-domain word embeddings? A study for Legal Cases

Milagro Teruel and Cristian Cardellino

Universidad Nacional de Córdoba, Argentina
milagro.teruel@gmail.com ccardellino@famaf.unc.edu.ar

Abstract. In this paper we explore the contribution of word embeddings for domain adaptation, more precisely, for Named Entity Recognition and Classification in the legal domain. We compare two different kinds of models: obtained from a portion of Wikipedia, and obtained from a very small but in-domain corpus. Wikipedia models can be trained with a large corpus without further annotation efforts, but the examples used are out-of-domain. In contrast, in-domain models require new annotated examples and are expected to be more accurate, but also more prone to overfitting. In this setting, we expect that word embeddings will be useful because they provide a smoothed representation of the data. We compare different kinds of word embeddings with models trained with traditional linguistic features, and find that word embeddings decrease overall performance, but improve recall. This behavior is specially beneficial for legal applications, where coverage is more important than precision.

Keywords: domain adaptation, word embeddings, legal named entity recognition and classification

1 Introduction

Information Extraction (IE) systems can have a high impact on many tasks, and the legal domain is no exception [1]. Legal records are stored in natural language documents, unstructured or semi-structured, and constitute an important factor in law interpretation, legal reasoning and decision making. In particular, the identification of relevant jurisprudence allows law practitioners to build sounder argumentations for new cases, and many semi-automatic solutions are heavily used nowadays [2]. The task of Named Entity Recognition and Classification (NERC) is a building block on IE systems and can also influence the performance of other tasks related to the legal domain, such as argument mining, claim identification or automatic reasoning [3].

Although IE and NERC systems have been very popular over the last two decades and many systems are available, the processing of legal documents is special in several aspects. Legal documents are more structured than general text, often recurring to very formulaic expressions, the vocabulary is used with precise and special meaning, etc. Moreover, Named Entities are not only names of people, places or organizations, as in general-purpose NERC. Names of laws, of typified procedures and even of concepts are also Named Entities in legal cases, as can be seen in the example in Figure 1.

Only very few annotated legal corpora exist, so it is difficult to train a Named Entity Recognizer. In the legal domain, the effort of annotation is specially high because only trained annotators can produce the corpora, given the very technical and precise semantics of those documents. A usual workaround consists in obtaining a model from general-domain documents, and then applying techniques of Domain Adaptation [4] improve the performance using the small available data in the target domain.

In this work, we propose to apply an automatic classifier for NERC in judgments of the European Court of Human Rights (ECHR). To cope with the lack of labeled examples, we use a portion of the English Wikipedia relevant to the legal domain as the starting training dataset. The selection of documents is obtained through the alignment of two ontologies: the legal ontology LKIF [5] which delimits the legal content of interest to us, and the YAGO ontology[1], which provides a way to link concepts to Wikipedia documents containing them. The use of a big number of Wikipedia documents allows us to have a baseline classifier for the legal domain without the need of an extensive annotation process. We were able to extract 102,000 entities and 4,5 million mentions.

For evaluation purposes, we have annotated a small set of cases of the ECHR. With this labeled dataset, we can compare the results of the classifier trained with a big number of Wikipedia documents against a classifier trained with a small amount in-domain data. Both classifiers, along with state of the art classifiers and simple heuristics, are applied to previously unseen judgments. Results are analyzed in Section 5, after the description of the dataset (Section 3) and the NERC system (Section 4).

2 Related work

A good review of data mining and information extraction methods applied to the legal domain can be found in the book by Stranieri (2005)[1]. It covers general Natural Language tools, as well as Machine Learning techniques.

Dozier et al. (2010) [2] approach the same problem as us, but with several differences. First, they consider fewer categories of Named Entities, leaving out abstractions and procedural entities. Second, their approach is rule-based, complemented with statistical methods, while we propose a fully data-driven method. Last, they use a different dataset: legal cases from United States Courts.

Quaresma and Gonçalves (2010) [6] also work in the NERC task over a set of European Union law documents in several languages, applying an automatic SVM classifier. They identify only locations, organizations, dates and references to other documents. Our proposal differs in the machine learning method and the use of out-of-domain training data.

Domain adaptation techniques have been also used for NERC in Social Media text by [7]. The authors gather unlabeled documents from several sources in

[1] www.yago-knowledge.org/

similar domains, together with a classifier pre-trained on a different domain. Next, they apply a bootstrapping method to select instances labeled with highest confidence by the classifier for further training.

The use of Wikipedia links in documents for Entity Identification is explored in [8], with successful results over in-domain and general documents.

3 Dataset descriptions

3.1 Wikipedia dataset

Wikipedia has been used as a corpus for NERC because it provides naturally occurring text where entities are manually tagged. Moreover, there is an explicit connection between Wikipedia URIs and nodes in the YAGO ontology. We aim to build a training dataset for NERC taking advantage of the vast amount of information already available in this resource. This process has two steps: the identification of entity mentions in documents, and the selection of relevant entities to include in the training dataset.

For the first task, we downloaded an XML dump of the English Wikipedia from March 2016. Then, we consider as mentions of an entity every anchor text of hyperlinks pointing to the entity's Wikipedia page, as in [8]. This results in accurate examples, but with a high number of false negatives, provided that usually only the first mention of an entity in a document is labeled.

As we mentioned before, to select the relevant entities to train the model we rely on the legal ontology LKIF, which we aligned to the Wikipedia-linked ontology YAGO. From this ontology, we obtain 122 populated YAGO classes aligned to LKIF and all entities included in these classes. We extracted all articles that contained at least one link to an entity in this set, obtaining a total of 4,5 million mentions, corresponding to 102,000 unique entities. Then, we kept only sentences that contained at least one mention of a named entity.

For the NERC problem we consider each word as a training instance. Given a sentence, each word is labeled independently as a Named Entity if it is contained in the anchor of an entity mention. More than 90% of the words (instances) were not inside a mention. This imbalance in the classes results in largely biased 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.

3.2 Abstraction levels

Once instances were obtained, we defined labels to assign them. There are multiple possible levels of abstraction for Named Entities. To assess the performance of a classifier in several abstraction levels, we established some orthogonal divisions in the LKIF-YAGO ontology, organized hierarchically. The final levels and the number of labels in each of them we use for classification are listed below:

1. NERC (6 labels): Instances are classified as: Abstraction, Act, Document, Organization, Person or O (Non-Entity).

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]$_{abstraction}$

LKIF

The [Court]$_{Public\_Body}$ is not convinced by the reasoning of the [combined divisions of the Court of Cassation]$_{Public\_Body}$, because it was not indicated in the [judgment]$_{Decision}$ that [Eğitim-Sen]$_{Legal\_Person}$ had carried out [illegal activities]$_{Crime}$

YAGO

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

Fig. 1. An example of legal entities annotated at different levels of granularity.

2. LKIF (21 labels): Instances are classified as belonging to an LKIF node, for example Legal_Role, Public_Body, Right, etc.
3. YAGO (122 labels): Instances are classified as belonging to the most concrete YAGO node possible, for example: wordnet_prosecutor_110484858, wordnet_human_right_105176846.

In Figure 1, we show an example annotated at these different levels of abstraction.

3.3 ECHR judgments dataset

From the ECHR website[2] we downloaded 5 random documents and annotated the sections describing the relevant Law and the Court's reasoning, leaving out the description of the facts. From a total of 19,000 words, we identified 1,500 entities and 3,650 instances.

The annotation was carried by four researchers involved in this work, following specific guidelines inspired in the LDC guidelines for annotation of NEs [9]. They were instructed to assign labels only at the YAGO level (the most specific), and the labels of the remaining levels were inferred using the hierarchical structure of the LKIF-YAGO ontology. Each annotator worked on at least two documents. To assess the agreement between annotators, four documents were annotated by two different people, achieving an inter-annotator from $\kappa = .4$ to $\kappa = .61$ using Cohen's kappa coefficient [10]. Most of the disagreement between annotators was found for the recognition of concepts, not for their classification. We are working on developing the guidelines to enhance consistency among

[2] http://hudoc.echr.coe.int/eng

annotators. We will also apply automatic pre-processing and post-edition to annotated texts, in order to spot and correct errors. We randomly selected one of these duplicated annotations as correct for the final experimentation dataset.

To obtain high quality annotations independent of the current task, annotators were allowed to add new classes to the existing label list. Considering all final annotations, 14 new classes were added in the LKIF level and 73 on the YAGO level. This indicates the classes available in the LKIF-YAGO ontology are not covering the semantic domain of ECHR judgments, as can be expected in a domain adaptation problem. The main factor is that the LKIF ontology does not contain classes for the subdomain of *Procedural Law* or *Crime*, which are very frequent in a legal case document.

To assess performance in a realistic scenario, classifiers are evaluated in a single hold out document from the ECHR annotated dataset. The hold out document can contain previously unseen classes, as in a real application. All classifiers were trained using the remaining set of documents, separating 90% of instances for training and 10% for validation (parameter tuning). The evaluation process was repeated leaving out a different document in each iteration, and finally averaging the results of all iterations.

4 Building the NERC systems

4.1 Representation of instances

Word embeddings have been shown to help in cross-domain classification problems [11, 12] because they capture latent properties of words that are less dependent on the domain. This can also be viewed as a smoothing of the resulting representation, which should be specially adequate to address overfitting. 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. Therefore, we trained three kinds of embeddings: obtained from Wikipedia documents alone (a very big corpus), obtained from the judgments alone (an in-domain corpus), and obtained with a mixed corpus. The mixed corpus is composed of all the available judgments of the ECHR, and a similar quantity of text from Wikipedia (an augmented in-domain corpus).

To train the word embedding vectors we used the Word2Vec's skip-gram algorithm [13]. For the Wikipedia dataset, we use the documents described in Section 3.1. Words with less than 5 occurrences were filtered out, resulting in a 2.5 million unique tokens, where the capitalization of words is preserved. 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, summing up to 9,161 documents with 70 millions tokens and 131,000 unique words. The trained embeddings were vectors with 200 elements, and taking them we generated a matrix where each instance was represented by the vectors of the instance word and the vectors of the surrounding words by a symmetric window of 3 words at each size. If the word was near the beginning or the end of a sentence, or if any word was not in the Word2Vec model, the vector was padded with zeros.

We compared the performance of word embeddings with the standard set of features proposed by Finkel *et al.* [14] for the Stanford Parser CRF-model. For each instance 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 presence of a word in as the total or as part of an entity in a gazetteer. To reduce the dimensionality of the final feature vector due to memory limitations, we applied a simple feature selection technique filtering out all features with variance less than 2e-4. We call this representation handcrafted features, in contrast with automatically obtained word embeddings.

4.2 Classifiers

Using the corpus described in the previous section, we trained a Multilayer Perceptron (MLP) classifier for each abstraction level with a similar architecture. We experimented with one, two and three hidden layers, but it resulted that a single hidden layer performed better. We select an architecture with a hidden layer of size 8000 for the Wikipedia dataset with handcrafted features, of 5000 for the ECHR dataset with handcrafted features, and 2000 for all experiments using word embeddings.

To better assess the performance of these classifiers, we compare them with a sequential classifier: a Conditional Random Field model. We use the Stanford CRF for NERC [15] implementation. We retrained this classifier for all abstraction levels with the ECHR dataset, but the YAGO level had too many classes to be trained with the Wikipedia dataset. The representation used is the same as for MLP classifiers, except for the presence in gazetteers and the PoS tags of surrounding words. The second baseline proposed is a K-Nearest Neighbors classifier trained using the current, previous and following word tokens over the ECHR dataset. This is a very simple approach, equivalent to checking the overlap of the terms in the entity. We consider this baseline appropriate for the evaluation of the ECHR documents, where entity mentions tend to be more regular, such as ”the applicant” or ”the Court”. However, the bigger vocabulary and higher number of entities made this approach unfeasible in the Wikipedia dataset.

5 Analysis of results

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 not weighted by the population of the class (macro-average).

In total, we have trained four different MLP classifiers varying the representation of the instances used. The different representations are: handcrafted

	Wikipedia trained				ECHR trained			
Classifier **NERC** task	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
MLP	**.76**	**.56**	**.24**	**.25**	**.80**	**.69**	.41	.47
MLP+WordEmb wiki	.73	.34	.21	.21	.78	.54	**.58**	.55
MLP+WordEmb mix	.75	.42	.23	.23	.77	.48	.50	.48
MLP+WordEmb echr	.75	.38	.24	.24	.77	.52	.54	.52
CRF	.73	.36	.17	.16	.79	.67	.51	**.56**
K-NN	-	-	-	-	.73	.54	.49	.50

Table 1. Performance of evaluation in ECHR holdout documents for the NERC task for classifiers trained with Wikipedia or ECHR documents. The metrics presented are averaged using a macro strategy. Classifiers are MultiLayer Perceptron (MLP), Conditional Random Field (CRF) and K-Nearest Neighbors (K-NN). The MLP classifier is also combined with Word Vectors (+WordEmb) representations from different datasets.

	Wikipedia trained				ECHR trained			
Classifier **LKIF** task	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
MLP	**.76**	**.13**	**.07**	**.08**	**.81**	.46	.24	.28
MLP+WordEmb wiki	.74	.08	.05	.05	.79	.30	**.34**	.29
MLP+WordEmb mix	.75	.10	.06	.06	.77	.28	.32	.28
MLP+WordEmb echr	.75	.11	.07	.07	.75	.27	.32	.27
CRF	.73	.07	.06	.05	**.81**	**.49**	.30	**.34**
K-NN	-	-	-	-	.73	.32	.27	.25

Table 2. Performance of evaluation in ECHR holdout documents for the LKIF task for classifiers trained with Wikipedia or ECHR documents. The metrics presented are averaged using a macro strategy. Classifiers are MultiLayer Perceptron (MLP), Conditional Random Field (CRF) and K-Nearest Neighbors (K-NN). The MLP classifier is also combined with Word Vectors (+WordEmb) representations from different datasets.

	Wikipedia trained				ECHR trained			
Classifier **YAGO** task	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
MLP	**.76**	**.06**	**.03**	**.03**	**.81**	**.33**	.18	**.21**
MLP+WordEmb wiki	.74	.03	.02	.02	.80	.24	**.22**	.19
MLP+WordEmb mix	.75	.04	.04	.03	.78	.23	**.22**	.18
MLP+WordEmb echr	.74	.04	.03	.03	.79	.22	**.22**	.19
CRF	-	-	-	-	.80	.28	.21	**.21**
K-NN	-	-	-	-	.72	.22	.18	.16

Table 3. Performance of evaluation in ECHR holdout documents for the YAGO task for classifiers trained with Wikipedia or ECHR documents. The metrics presented are averaged using a macro strategy. Classifiers are MultiLayer Perceptron (MLP), Conditional Random Field (CRF) and K-Nearest Neighbors (K-NN). The MLP classifier is also combined with Word Vectors (+WordEmb) representations from different datasets.

features, Wikipedia word embeddings, ECHR word embeddings, and word embeddings mixed from both sources. Additionally, we compare how the classifiers perform if they are trained with Wikipedia documents and ECHR documents. As a result, we evaluated eight MLP configurations. The performances of the baseline classifiers CRF and K-NN described in the previous section are also presented.

Results for the three proposed levels of abstraction: NERC, LKIF, and YAGO, are shown in Tables 1, 2, and 3 respectively. For the sake of comparison, we note that the reference tool for NERC achieves an F1 around 85-90% for in-domain training-testing with a large corpus at a level of granularity comparable to the NERC level [14]. In our approach performance never goes over 60% F1, probably due to a small training corpus, with few representatives of some of the classes. Indeed, using this same reference tool on our dataset yields only 56% performance at the NERC level.

We can observe from the three result tables that MLP classifiers outperform the K-NN baseline and perform comparably or better than the CRF classifier. Wikipedia-trained classifiers obtain lower figures in general, which could be expected because many of the classes in the ECHR corpus are not in Wikipedia and they could not be learned. This sets a ceiling to the performance of Wikipedia-trained models, which is a maximum accuracy of 0.866 for the LKIF level and 0.804 for the YAGO level. The obtained accuracy is then only 10 points below this ceiling of performance. However, even if the accuracy of Wikipedia-trained classifiers is not bad, the macro precision, recall and F1 score clearly show that they are not recognizing most of the classes. What they are actually doing is recognizing a small number of classes which have a big number of examples.

Focusing on ECHR-trained classifiers, we can see they achieve a better performance for all levels of abstraction, which is expected as we are training and evaluating on the same domain. However, developing a specific annotated corpus is costly, while Wikipedia provides a huge amount of annotated examples of a similar domain, for free.

Considering accuracy only, the MLP classifier performs better without word embeddings. As shown in figure 2, ECHR-trained classifiers with embeddings have a consistently higher recall, with a decrease in precision. This behavior is specially beneficial for legal applications, which are normally retrieval-related.

We also highlight that word embeddings trained with Wikipedia documents tend to perform better on models trained with the ECHR dataset, but there is no consistent difference between mixed and ECHR trained embeddings. The opposite occurs with the Wikipedia-trained models, where ECHR and mixed word embeddings improve both precision and recall. These two results show 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 improves more the classifiers. Even more, a very simple way of mixing examples for word embeddings in some cases enhances performance.

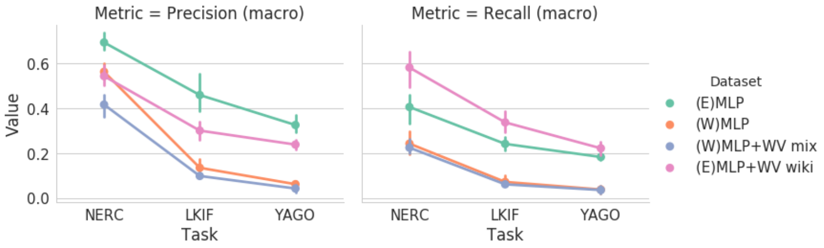

Fig. 2. Evaluation results for the most relevant classifiers over holdout ECHR documents, trained using the Wikipedia dataset (W) and the ECHR dataset (E)

6 Discussion and future work

Our results show that word embeddings are beneficial to improve the recall of a small, in-domain model for NERC in legal documents. This is specially important for legal applications, which are mostly retrieval-centered and can tolerate noise better than silence. Word embeddings trained with unlabeled in-domain documents perform better than generic embeddings when the model has been trained in a different domain.

We have found that the most naïve combination of embeddings from different domains slightly improves classifier performance. We will investigate combinations of embeddings that are specifically targeted for domain adaptation.

At the same time, we have shown that Wikipedia-trained models achieve a reasonable level of performance in the legal domain, without any annotation cost. A promising line of work is to explore techniques to select documents of Wikipedia or other sources that will produce models closer to judgment documents, including more information of procedural law.

Acknowledgements

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. Andrew Stranieri, J.Z.: Knowledge Discovery from Legal Databases. 1 edn. Law and Philosophy Library 69. Springer Netherlands (2005)
2. Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., Wudali, R.: Semantic processing of legal texts. Springer-Verlag, Berlin, Heidelberg (2010) 27–43

3. Surdeanu, M., Nallapati, R., Manning, C.D.: 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 (May 2010)
4. Sogaard, A.: Semi-Supervised Learning and Domain Adaptation in Natural Language Processing. 1st edn. Morgan & Claypool Publishers (2013)
5. Hoekstra, R., Breuker, J., Bello, M.D., Boer, A.: The lkif core ontology of basic legal concepts. In: Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007). (2007)
6. Quaresma, P., Gonçalves, T. In: Using Linguistic Information and Machine Learning Techniques to Identify Entities from Juridical Documents. Springer Berlin Heidelberg, Berlin, Heidelberg (2010) 44–59
7. Tian, T., Dinarelli, M., Tellier, I., Cardoso, P.D.: Domain adaptation for named entity recognition using crfs. In Chair), N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (may 2016)
8. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. CIKM '08, New York, NY, USA, ACM (2008) 509–518
9. Linguistic Data Consortium: Deft ere annotation guidelines: Entities v1.7. http://nlp.cs.rpi.edu/kbp/2014/ereentity.pdf (2014)
10. Cohen, J.: A coefficient of agreement for nominal scales. Educational & Psycological Measure **20** (1960) 37–46
11. Nguyen, T.H., Grishman, R.: Employing word representations and regularization for domain adaptation of relation extraction. In: ACL. (2014)
12. Yang, Y., Eisenstein, J.: Unsupervised multi-domain adaptation with feature embeddings. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, Association for Computational Linguistics (May–June 2015) 672–682
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. ArXiv e-prints (January 2013)
14. Finkel, J.R., Grenager, T., Manning, C.: 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, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 363–370
15. Stanford NLP Group: Stanford named entity recognizer (ner). http://nlp.stanford.edu/software/CRF-NER.shtml (2016)