# A Review of Implementing Natural Language Processing in Legal Texts

**Yiming Gu**
yimingg7@illinois.edu

## I Introduction

Text is one of the key components in the field of law. Ordinally, all texts in this field are expressed in natural language. Natural Language Processing (NLP) provides methods for converting texts into formal representation and could then be used for further analysis and other applications. Meanwhile, law is in many ways particularly conducive to the application of AI and machine learning as the legal reasoning is based on the formal logic and the logic-orientated methodology is exactly the type of activity where the AI can be applied to. Actually, this is a rapidly evolving area and a lot of startups are working in this field [2].

There are millions or billions of legal documents around the world. Generally speaking, these documents could be divided into two categories, the statutory law and legal solutions or opinions, even though someone may have different criterion regarding the document categorization in some particular jurisdictions [1]. The statutory law ordinarily includes law codes and administrative rules, policies. The characteristic of such kind of documents is that they provided a relatively structured and organized texts, although the ambiguity of its content or expression is relatively higher. The legal solutions or opinions are the documents produced by lawyers and judges generally, including the contacts, the decisions, opinions. Such texts are generally unstructured and they don't require a standard structure or expression and the ambiguity of its content is comparatively lower due to the fact that they are generally an interpretation of the statutory law.

The aforementioned classification is crucial because the characteristic of a particular document is closely related to the goals or methods of NLP. For example, for statutory law, it is hard to conduct a prediction its outcome because of the ambiguity of words in such documents.

Generally, NLP techniques can be implemented on legal text in two ways, information extraction and outcome prediction.

## II Information Extraction

Information extraction is a major task of implementing NLP technique directly on legal texts. The primary goal of this task is to retrieve useful, structured or particular information from large amounts of unorganized or unstructured documents. Actually, as the number of documents is increasing rapidly, the need for retrieving documents that could precisely meet the requirement of the searchers is becoming more and more urged. Fortunately, a lot of different ways are proposed by scientists in response to such needs.

1. Semantic Annotation [3]

This solution provides the sort of implicit information that human readers are naturally be able to get during their reading. The *semantic annotation* refers to the process of labeling texts by tags indicating its semantic content.

A SALEM framework is proposed in this solution and it has two tasks: 1) to assign each paragraph of law to a given legislative provision type; 2) to automatically tag the parts of the paragraph with domain-specific semantic roles identifying the legal entities (such as the actors, actions, etc.) referred to in the provision. In sum, the SALEM takes a law paragraph as an input and outputs a semantic tagging of the text, including its classification and semantic roles, in XML tags.

This approach is achieved by taking two steps, the syntactic pre-processing and semantic annotation. The syntactic pre-processing produces the data structures. The text is first tokenized and normalized and then, analyzed and lemmatized using an Italian lexicon. Then, the text is POS-tagged and shallow parsed into non-recursive "chunks", which includes the type of information, the lexical head and any intervening modifier, causative or auxiliary verb and preposition.

The second stage of this approach is semantic annotation, which further contains two steps. The first step involves assigning each paragraph to a provision type frame, which includes all instances of provision types. The slots of the frame are then turned to an extraction template and filled with information extracted from texts. Both the recognition of provision types and information are based on the combination of both syntactic and lexical criteria, such as the model verb 'shall/must' are indicators of obligation and its combination with identification of the type of dependency relations.

This solution actually facilitates the reading and analysis of the content of the text. Actually, it is more closely related to the classification of text itself. It uses pre-defined pattern to help understanding of texts. This is a case that is representative in the earlier stage of the development of the NLP in law. By using part-of-speech tagging and other useful tools, such as name-entity recognition, syntactic parsing, texts could be divided into different categories and text content could be applied to different parts of framework. It reflects a preliminary step of content analysis. However, this solution cannot recognize the paragraph that does not fit in the pre-defined framework and cannot recognize the similarity between two different expressions.

2. Gov2Vec [4]

Gov2Vec provides a method that investigate the differences or similarities between different institutions through the legal corpus of each institution. By doing so, similar words could be added to the query and the information retrieval could be conducted more precisely and encompassed.

The solution is based on a common method for learning vector representations of words, which is to use a neural network to predict a target word with the mean of its context word's vectors. After iterating over many contexts, words share similar meaning could be embedded in similar locations in vector space. Inspired by this method, the method embeds institutions and corresponding corpus into a shared vector space. This is achieved by averaging a vector unique to an institution with context word vectors and back-propagation and stochastic gradient decent.

After training the model, the cosine similarity of words to the particular vector

combinations could be calculated and similar words could be pursued by setting a threshold.

By implementing this method, information retrieval on legal text through query could return documents more precisely.

This solution provides a more in-depth NLP than the solution provided in the first part of the section. Actually, it is an application of a general solution to a particular kind of document, that is, legal text. It is based on the assumption that the context is closely related to the part being discussed. Even though this assumption holds true in the situation, it may be more preferable if this method could be combined with the logic relations embedded in legal context. This is because the logics and meaning of the same words in different context could vary from each other greatly, which is an area that needs further investigation by both the legal experts and the artificial intelligence experts.

## III Outcome Prediction

Another major task of implementing NLP in the legal field is to assist lawyers or counsels to predict the outcome of particular case, such as the probability of an action that violets an article. A lot of researches have been conducted in this direction with positive feedbacks.

1. Predicting the Judicial Decisions of ECHR [5]

In this article, the author proposes a solution to predict whether a particular Article of European Convention on Human Rights has been violated based on the textual evidence extracted from a case, which comprises of facts, applicable law and the arguments presented by the parties involved. It makes an assumption that the textual content and different parts of a case are important factors that could influence the outcome. Furthermore, the assumption that there is enough similarity between certain chunks of the text of published judgements must also be made.

The author defines this solution as a binary classification task, which means the solution is to predict whether a particular case violate or not violate a specific Article of the Convention. The solution focuses on Articles 3, 6 and 8 in particular because there are a large number of cases involved these articles. The solution uses published cases in HUDOC, a legal repository, and excludes any sections on operative provisions of the court in order not to use information pertaining the outcome of a case.

The solution derives N-gram textual features from the documents. For each set of cases, it compute the top-2000 most frequent N-grams, where $N \in \{1,2,3,4\}$. Each feature represents the normalized frequency of a particular N-gram in a case. This solution also creates topics for each article by clustering together N-grams that are semantically similar by using C feature matrix, where each column vector of the matrix represents a N-gram and each row represents a case. Then the author computes N-gram similarity using the cosine metric and create an N-gram by N-gram similarity matrix. Then, spectral clustering, which performs graph partitioning [6] on the similarity matrix, is applied to get the 30 clusters of N-grams.

For classification, the solution uses N-grams and topics to train Support Vector Machine (SVM) classifiers [7]. The violation cases are labelled as +1 and no violation cases are denoted by -1. Therefore, features assigned with positive weights are more indicate of violation. The models are also trained and tested by applying 10-fold cross validation.

This solution provides an example of predicting the violation of specific provision in

the condition of facts. Generally, it provides some good on the application of NLP in predicting the outcome of a case. However, due to the fact that it employs only part of the cases, it is still not as encompassed as we expect because there are actually a lot of cases that are rejected at administrative stages. Therefore, this solution could still be improved. Meanwhile, the solution focuses on only one particular aspect of the process of determination. That is, the determination of violation or non-violation. Actually, there are still a lot of questions yet to be answered, such as whether a party hold liability for the violation, etc. Considering that, improvement could still be made.

2. TOPJUDGE [8]

This article provides a solution to address of multiple tasks in legal judgement. There are a lot of different but related subtasks in legal judgement, such as the application of law, the liabilities of parties involved. Such subtasks are often related to each other as the judges will provide reasoning one after each other. This solution proposes a multi-task learning framework, TOPJUDGE, to exploit the dependencies among the subtasks in judgements.

Basically, the framework is comprised of two parts, the Encoder part and the DAG Predictor part. The Encoder part process the facts of cases and produce a fact representation vector. Then the solution will predict the judgement based on the fact representation vector over DAG.

A fact encoder is employed to generate a fact description's representation vector as an input of TOPJUDGE. There are three layers in this stage. Firstly, each word $x_i$ in a word sequence x, which is the fact description of a case, is converted into its word embedding $x_i \in R^k$. Then, a convolution operation is applied to produce a feature map by a convolution matrix $W \in R^{m\times(h\times k)}$, where feature map $c_i = W \cdot x_{i:i+h-1} + b$ and b is the bias vector. Finally, a pre-dimension max-pooling over c is implemented and the fact representation d is obtained.

Then the TOPJUDGE will conduct judgement predictor based on the assumption of DAG, which represents Directed Acyclic Graph and means all subtasks are arranged in topological order. For each task $t_j$ in task list $T = \{t_1, t_2, \ldots, t_{|T|}\}$, it is to predict the judgment result $y_j$ based on the fact representation vector d. The method employed is LSTM cell, which comprised three steps, including cell initialization, task-specific representation and prediction. Considering the result of $t_j$ depends on the representation d and the outputs of all dependent tasks $y_k$, it uses the transformation matrix $W_{i,j}$, and bias vector $b_j$ to generate $h_j$, which is a task-specific representation of task $t_j$, and last cell state $c_j$ with the following formular:

$$\begin{bmatrix}\bar{\mathbf{h}}_j \\ \bar{\mathbf{c}}_j\end{bmatrix} = \sum_{t_i \in D_j} \left(\mathbf{W}_{i,j}\begin{bmatrix}\mathbf{h}_i \\ \mathbf{c}_i\end{bmatrix}\right) + \mathbf{b}_j$$

, where the fact representation d, initial hidden state $\bar{h}_j$ and initial memory cell $\bar{c}_j$ are inputs. With regard the representation $h_j$, an affine transformation of softmax is applied and obtain the final prediction as

$$\hat{\mathbf{y}}_j = \text{softmax}\left(\mathbf{W}_j^p \mathbf{h}_j + \mathbf{b}_j^p\right).$$

where $\mathbf{W}_j^P$ and $\mathbf{b}_j^P$ are both parameters specific to task $t_j$.

Overall, this solution provides a more comprehensive way to predict the outcome of a case, as compared with the previous solution. In particular, it addresses the fundamental relations between different subtasks in a judicial judgement. However, the limitation of such method is obvious as you have to define the task lists but in real world, such kind of

implementation have very restricted use as the sub tasks in different cases, even though these cases have similar context or law application, may vary greatly. Actually, such kind of method could only be used in some legal procedures with fixed structure and in a general context, it is hard to implement.

## IV Conclusion

This paper reviews technologies used when processing legal texts. Due to the nature of the legal documents, which is logic-based, and their large volume, they are suitable texts for natural language processing. Currently, there are two major directions of research in this area. One focuses on extraction of information from the dense documents and another focuses on predicting outcomes based on existing facts. Researches in both directions provide positive feedbacks and may assist legal professional to some extent. Current researches are still restricted in specific tasks and circumstances, however, in real world, the facts may be much more complex than published texts. The data bases and assumptions used in those techniques are still problematic and yet to reach completeness. Therefore, the current state-of-art NLP techniques cannot support a comprehensive analysis on legal texts and more in-depth researches in this field need to be carried out.

## References

[1] Nay, John, Natural Language Processing and Machine Learning for Law and Policy Texts (April 7, 2018). Available at SSRN: https://ssrn.com/abstract=3438276 or http://dx.doi.org/10.2139/ssrn.3438276

[2] Rob Toews, AI Will Transform The Field of Law (December 19, 2019). Available at: https://www.forbes.com/sites/robtoews/2019/12/19/ai-will-transform-the-field-of-law/#5600d16f7f01

[3] Biagioli, Carlo & Francesconi, Enrico & Passerini, Andrea & Montemagni, Simonetta & Soria, Claudia. (2005). Automatic semantics extraction in law documents. 133-140. 10.1145/1165485.1165506.

[4] Nay, John, Gov2Vec: Learning Distributed Representations of Institutions and Their Legal Text (November 5, 2016). Nay, J. (2016). "Gov2Vec: Learning Distributed Representations of Institutions and Their Legal Text." Proceedings of 2016 Empirical Methods in Natural Language Processing Workshop on NLP and Computational Social Science, 49–54, Association for Computational Linguistics., Available at SSRN: https://ssrn.com/abstract=3087278

[5] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. 2016. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. PeerJ Computer Science 2:e93 https://doi.org/10.7717/peerj-cs.93

[6] von Luxburg U. 2007. A tutorial on spectral clustering. Statistics and Computing 17(4):395–416.

[7] Vapnik VN. 1998. Statistical learning theory. New York: Wiley

[8] Zhong, Haoxi & Guo, Zhipeng & Tu, Cunchao & Xiao, Chaojun & Liu, Zhiyuan & Sun, Maosong. (2018). Legal Judgment Prediction via Topological Learning. 3540-3549. 10.18653/v1/D18-1390.