Finding Contextually Consistent Information Units in Legal Text

Dominic Seyler*
dseyler2@illinois.edu
University of Illinois at Urbana-Champaign

Paul Bruin
Pavan Bayyapu
paul.bruin@regology.com
pavan.bayyapu@regology.com
Regology

ChengXiang Zhai
czhai@illinois.edu
University of Illinois at Urbana-Champaign

ABSTRACT

Terms in the laws of a legislature can be highly contextual: especially for corpora of codified laws and regulations where the reader has to be aware of the correct context when the corpus lacks a single level of hierarchy. The goal of this work is to assist professionals when reading legal text within a codified corpus by finding contextually consistent information units. To achieve this, we combine NLP and data mining techniques to develop novel methodology that can find these information units in an unsupervised manner. Our method draws on expert experience and is modeled to emulate the "contextualization process" of experienced readers of legal content. We experimentally evaluate our method by comparing it to multiple expert-annotated datasets and find that our method achieves near perfect performance on four state corpora and high precision on one federal corpus.

KEYWORDS

information units, logical document organization, legal text mining

1 INTRODUCTION

Within a corpus, the same term can have different meanings. For instance, in the United States Code (USC) 26 USC § 7701(a)(1) provides that for purposes of *Title 26* "The term 'person' shall be construed to mean and include an individual, a trust, (...) or corporation." However, 42 USC § 2000e(a) provides that for purposes of the *subchapter* it belongs to "The term 'person' includes one or more individuals, governments, (...) or receivers." Thus, the context is not identical across the corpus.

This work's goal is to inform professionals who read a section of legal text about the continuous and contextually consistent information unit the section belongs to. We coin such a unit "root context" and it commonly represents an individual law on a specific topic. Root context is used where a codified corpus' hierarchical structure does not designate a single level for individual laws. Our root context method is modeled to emulate the "contextualization process", where experienced readers use references in the text to help contextualize what they are reading. The contextualization process can be broken down into three steps: 1) The reader notices that definition of 'person' is not in the current section. 2) The reader needs to find the definition of 'person' that is applicable to the current section. 3) Once the definition for 'person' is found

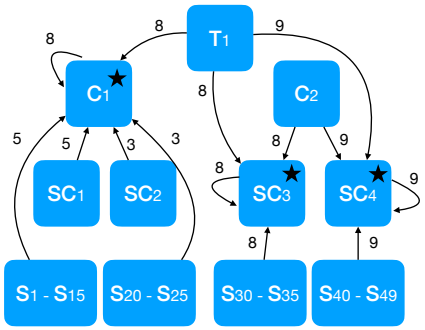

Figure 1: An example of a hierarchy-reference graph. Identified root contexts are highlighted with ⋆.

the reader needs to understand which other sections the definition applies to. For instance, Figure 1 shows Chapter 1 as root context. If the reader is reading Section 12 she might find the mention of the word 'person'. From our algorithm the reader is informed that the definition of 'person' is applicable to Chapter 1, which encompasses Section 1 through Section 25.

One challenge is to optimize the existing hierarchy within legal documents. For instance, the USC has 53 titles that are broken down in lower levels of hierarchy such as chapters, parts, etc. We address the aforementioned challenge by combining the natural language text and the existing document hierarchy to find higher-level logical groupings, which are not present in the original document hierarchy (we call these groupings "root contexts"). To achieve this, our method makes use of NLP techniques to extract hierarchy references from the text and automatically builds an hierarchy-reference graph (an example is depicted in Figure 1 with hierarchy references shown as arrows in the figure). We then create an algorithm that follows the graph references to automatically identify a point in the hierarchy that is a root context.

We evaluate our approach on five U.S. law corpora (one federal and four state corpora). To build our test datasets, we have a domain expert annotate every root context in each of the corpora. In our experiments, we compare these annotations with the predictions of our algorithm and find that our method achieves near perfect performance on our state corpora and high precision on our federal corpus. Summarizing, our work makes the following contributions:

(1) We introduce and define the novel problem of root context identification.
(2) We combine NLP and data mining techniques to develop novel methodology to identify root contexts.
(3) We show the effectiveness of our approach on five corpora.

*This work was done during an internship at Regology.

NLLP @ KDD 2020, August 24th, San Diego, US

2 RELATED WORK

The identification of information units is common in library and information sciences [18] and web information retrieval [12, 16, 19]. There, the purpose is to find a set of documents or articles that form a logically and contextually coherent unit. In the legal domain, it is also common to create some sort of abstraction of legal text. Mostly this is achieved through information extraction, such as text summarization [9, 11], argument mining [13, 17], named entity extraction [6, 7], citation resolution [15] and visualization [10]. Combining these information extraction techniques as building blocks, another line of work builds structured knowledge resources with the intend of abstraction and logical organization of legal content [8, 14]. Our work is bridging the gap between information units in the sense of information retrieval, where we combine content on the logical level, and the legal domain. In contrast to other work in the legal domain, we leverage information units as a concept for guiding the user when navigating legal text. Rather than extracting content from an existing document collection, we augment the data with contextually consistent information units, which we call root contexts. Thus, root context is a novel application of the concept of information units to organize large amounts of semi-structured, legal text.

3 PROBLEM DEFINITION

Definition 3.1. Hierarchy instance: A concrete instantiation of a part of the hierarchy, e.g. "Title 12", denoted as T_{12}. Each hierarchy instance is the union of all its sub-parts. For example, T_{12} is made up of chapters $C_1, ...C_{18}$, which transitively are made up of subchapters SC_i, etc.

Definition 3.2. Hierarchy level: The union of all hierarchy instances for a certain level. For instance, the "title" hierarchy level $\mathcal{T}$ of USC encompasses all 53 title instances T_i: $(T_1, T_2, ..., T_{53}) \in \mathcal{T}$. The lowest level we consider for this work are Sections S, which contain all the textual information.

Definition 3.3. Hierarchy branch: A unique identifier for a specific part in the corpus hierarchy. The hierarchy branch is a tuple of hierarchy instances that need to be "followed" from the hierarchy root. For example, the tuple ('USC', 'T_{12}', 'C_2', 'S_{221}'), identifies Section 221 in Chapter 2 ("Federal Reserve System"), within Title 12 ("Banks and Banking") in the United States Code.

Definition 3.4. Hierarchy reference: The hierarchy branch of a reference made to the hierarchy in legal text. For example, if the text in ('USC', 'T_{12}', 'C_2', 'S_{12}') references its chapter, then the resulting hierarchy reference is: ('USC', 'T_{12}', 'C_2'). We define custom regular expressions to identify hierarchy references in the text.

Definition 3.5. Hierarchy-reference graph: A weighted, directed graph where the nodes are hierarchy branches and the edges are hierarchy references between them. The edge weight is the number of references between a pair of nodes.

Definition 3.6. Root context: A specific hierarchy instance, which makes up a contextually consistent information unit.

Definition 3.7. Root context identification: Given a document collection C (i.e., a law corpus), extract all hierarchy references E from the text and combine them with the existing document hierarchy V to form a hierarchy-reference graph $G = (V, E, w)$. More specifically, V is the set of all hierarchy branches, $E \subseteq \{(x, y)|(x, y) \in V^2\}$ are all directed hierarchy references and $w : E \rightarrow \mathbb{R}$ the edge weights. The goal is to find a set $R \subseteq V$ with all hierarchy branches (i.e., nodes in the graph) that are root contexts.

4 APPROACH

Algorithm 1: Root Context Identification

```
Data: C, a document collection (i.e., a law corpus), V a set of all hierarchy
      branches
Result: R, a set of all root contexts with R ⊆ V
begin
    G = (V, E, w) ← extract hierarchy references and build graph;
    D ← identify hierarchy references in definition and purposes sections;
    R ← ∅;
    MAX_HOPS ← 10;
    for v ∈ V do
        hops ← 0;
        cur_v ← v;
        while hops < MAX_HOPS do
            hops ← hops + 1;
            if cur_v ∈ D then
                R ← R ∪ cur_v;
                break;
            counts ← ∅;
            for (x, y)∀x = cur_v do
                count ← count ∪ (y, w((x, y)));
            if |count| > 0 then
                max_y ← max_{w((x,y))} count;
                if cur_v = max_y then
                    R ← R ∪ cur_v;
                    break;
                cur_v ← max_y;
            else
                cur_v ← parent(cur_v);
```

Our approach consists of three phases: 1) extract hierarchy references and build the hierarchy-reference graph 2) find root context indicators within definitions and purposes sections 3) perform multi-hops following the graph's edges with the highest weight until a root context is identified. Algorithm 1 describes our methodology for identifying root contexts. The input to the algorithm is a document collection C and a set of all hierarchy branches V. The output is a set of root contexts R.

Build hierarchy-reference graph ($G = (V, E, w)$): We start with the nodes of the graph V, which are all hierarchy branches in a legal corpus. Because only the section hierarchy level $\mathcal{S}$ includes textual context, we start investigating the text of all $S_i \in \mathcal{S}$. For each S_i, we extract all hierarchy references by looking for textual references of the hierarchy. We use a set of regular expressions to find these references in text, i.e., "this (<hierarchy level>)". For instance, if the legal text in ('USC', 'T_{12}', 'C_2', 'S_{12}') says "this chapter" we extract ('USC', 'T_{12}', 'C_2'). We then add an edge between all distinct hierarchy references and the current S_i, where the edge weight is the sum of the hierarchy references for the edge's target node. Once this step is completed for $\mathcal{S}$, we begin a count aggregation step at each hierarchy level, moving "upwards" in the hierarchy. In our running example, we move from $\mathcal{S}$ to the chapter hierarchy level $\mathcal{C}$. For each $C_i \in \mathcal{C}$, we aggregate all hierarchy references that go from C_i to other parts in the hierarchy. For instance, we aggregate all hierarchy references to T_{12} within the hierarchy instance of C_2, as a single edge from ('USC', 'T_{12}', 'C_2') to ('USC', 'T_{12}'), with the combined edge weight of all its children that are pointing to T_{12}.

Target	Regular Expression
definitions	this (\w+) (?:describes|sets forth|governs|contains)
definitions	the following definitions apply to this (\w+)
purposes	the purposes? of (?:the)?.*this (\w+) (?:are|include|is)
purposes	this (\w+) sets forth

Table 1: Regexes for finding definitions/purposes sections.

Identify root context indicators within definitions and purposes sections (D): References in certain sections that contain definitions or purposes carry more weight than regular sections, as these were authored with the intend of helping the reader with contextualization. Our extraction algorithm goes through all sections that contain either "definition" or "purpose" in their title. We define a set of regular expressions that extract the hierarchy references in these sections. Table 1 shows some examples, where "(\w+)" matches a reference to the hierarchy (e.g., "chapter"). If one of the regular expressions matches the text within a section, the hierarchy branch of the extracted reference is added to a set D.

Traverse hierarchy-reference graph to generate R: Starting at every node in the graph, we perform multiple hops (in our case 10) along the graph, following always the outgoing edge with the highest weight. If a node is found that is in D, it will automatically be added as a root context[1]. If a node's highest weighted outgoing edge points to itself, then we also consider the node a root context. If a node has no outgoing edges, we move up one level in the hierarchy, to the node's parent, and continue the procedure.

5 EXPERIMENTS

5.1 Experimental Setup

Our experimental evaluation aims at measuring the efficacy of our root context identification approach outlined in Section 4. For our experiments, we run our algorithm and compare its predictions to the expert annotations. We experiment with three full law corpora: United States Code (USC) [5], California Law (CACL) [1], Texas Statutes (TXST) [4] and two partial law corpora: Illinois Compiled Statutes (ILCS) [3] (10 out of 68 chapters covered) and Consolidated Laws of New York (NYCL) [2] (7 out of 92 chapters covered). For each corpus, we extract all of its textual contents and hierarchy from the web-pages available online.

For each hierarchy branch in V, we have a domain expert annotate whether it is a root context ("1" label) or not ("0" label). Table 2 shows the statistics for each dataset, with its total number of nodes ("Total"), the number of annotated root contexts ("Number Root Contexts") and the percentage of root contexts out of all nodes in the graph ("%").

Since our experimental setup is equivalent to a classification problem, we report F_1-score, precision, recall of the class under consideration (i.e., "1" label) and classification accuracy. In this case, false positives are instances that our method identifies as root contexts, however, the expert annotated these instances as not a root context. False negatives are root contexts that were identified by the expert but not found by our method.

[1]We experimented with different weighting schemes for nodes that are in D, however, we found that "overriding" the edge weights in D works best.

Dataset	Total	Number Root Contexts	%
USC	166,086	3,040	1.83
CACL	177,862	2,887	1.62
TXST	239,259	4,419	1.84
ILCS	19,088	800	4.19
NYCL	4,201	306	7.28

Table 2: Statistics of datasets.

Dataset	F_1	Precision	Recall	Accuracy
USC	0.71	0.95	0.56	0.99
CACL	0.97	0.98	0.97	1.00
TXST	0.95	0.98	0.91	1.00
ILCS	0.98	0.99	0.97	0.99
NYCL	0.92	1.00	0.85	0.99

Table 3: Classification results of root context identification.

5.2 Experimental Results

Table 3 shows the results of our method on different datasets. We find that our method generally achieves high precision (≥ 0.95), which means that if our method finds a root context one can be certain that it actually is a root context. Recall is also high for state corpora, but lesser so for our federal corpus, meaning that our method finds almost all root contexts on a state level. In our manual analysis we find that the reason for the lower recall on the federal level is that there are inherent differences in how the hierarchy is organized within the titles of USC. We see improving the recall on the federal level as an opportunity for future research. Accuracy is close to 1 for all corpora, which is not surprising since the number of root contexts is smaller than the number of nodes in the graph. Summarizing, we find that our method achieves near perfect performance for state corpora and high precision for USC.

6 CONCLUSION AND IMPACT

We presented the problem of finding contextually consistent information units to assist professionals when reading legal text and developed novel methodology to find these units. We evaluate our method and find that it achieves high precision and F_1 score on multiple datasets. The high accuracy of our method indicates that the task is not hard and the proposed method has worked well. Since the method is unsupervised, it does not require any manual work and can thus be applied broadly to all such application problems. Naturally, we may further improve performance by applying supervised learning with our method used as one feature and combine it with other features, especially text-based features, which would be future work. While we find our method to be effective, there are instances where root contexts cannot be identified due to the lack of hierarchy references. To solve this in the future, we envision to find root context indicators by looking at the evolution of legal text over time and study the differences of different versions of the law.

This work further aids Regology's[2] machine learning framework. For instance, the root context concept has been successfully utilized to build a keyword extraction algorithm; increase the performance of existing information retrieval components (e.g., grouping search results, assessing law changes, differentiating between new and amending laws), and extract topics.

[2]https://regology.com

REFERENCES

[1] [n.d.]. California Law. http://leginfo.legislature.ca.gov/faces/codes.xhtml

[2] [n.d.]. Consolidated Laws of New York. https://www.nysenate.gov/legislation/laws/CONSOLIDATED

[3] [n.d.]. Illinois Compiled Statutes. http://www.ilga.gov/legislation/ilcs/ilcs.asp

[4] [n.d.]. Texas Statutes. https://statutes.capitol.texas.gov/StatuteCodes.aspx

[5] [n.d.]. United States Code. http://uscode.house.gov/download/download.shtml

[6] 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 the 16th edition of the International Conference on Articial Intelligence and Law*. 9–18.

[7] Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. 2010. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*. Springer, 27–43.

[8] Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia. 2010. Integrating a bottom–up and top–down methodology for building semantic resources for the multilingual legal domain. In *Semantic Processing of Legal Texts*. Springer, 95–121.

[9] Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. Combining different summarization techniques for legal text. In *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data*. Association for Computational Linguistics, 115–123.

[10] Aikaterini-Lida Kalouli, Leo Vrana, Vigile Marie Fabella, Luna Bellani, and Annette Hautli-Janisz. 2018. CoUSBi: A Structured and Visualized Legal Corpus of US State Bills. In *Proceedings of the LREC 2018 "Workshop on Language Resources and Technologies for the Legal Knowledge Graph", Miyazaki, Japan.*

[11] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review* 51, 3 (2019), 371–402.

[12] Wen-Syan Li, K Selçuk Candan, Quoc Vu, and Divyakant Agrawal. 2001. Retrieving and organizing web pages by "information unit". In *Proceedings of the 10th international conference on World Wide Web*. 230–244.

[13] Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*. 225–230.

[14] Georg Rehm, Julian Moreno Schneider, Jorge Gracia, Artem Revenko, Victor Mireles, Maria Khvalchik, Ilan Kernerman, Andis Lagzdins, Mārcis Pinnis, Artus Vasilevskis, et al. 2019. Developing and orchestrating a portfolio of natural legal language processing and document curation services. In *Proceedings of the Natural Legal Language Processing Workshop 2019*. 55–66.

[15] Robert Shaffer and Stephen Mayhew. 2019. Legal Linking: Citation Resolution and Suggestion in Constitutional Law. In *Proceedings of the Natural Legal Language Processing Workshop 2019*. 39–44.

[16] Keishi Tajima, Kenji Hatano, Takeshi Matsukura, Ryoichi Sano, and Katsumi Tanaka. 1999. Discovery and Retrieval of Logical Information Units in Web.. In *WOWS*. 13–23.

[17] Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*. Springer, 60–79.

[18] Liangzhi Yu, Zhenjia Fan, and Anyi Li. 2019. A hierarchical typology of scholarly information units: based on a deduction-verification study. *Journal of Documentation* (2019).

[19] ChengXiang Zhai and John Lafferty. 2006. A risk minimization framework for information retrieval. *Information Processing & Management* 42, 1 (2006), 31–55.