Automated Directive Extraction from Policy Texts

Karl Branting
Jim Finegan
David Shin
Stacy Petersen
The MITRE Corporation
McLean, VA, USA
lbranting,jfinegan,hshin,spetersen@mitre.org

Carlos Balhana
Language Technology Lab
University of Cambridge
Cambridge, UK
ceb81@cam.ac.uk

Alex Lyte
The MITRE Corporation
Bedford, MA, USA
alyte@mitre.org

Craig Pfeifer
The MITRE Corporation
Ann Arbor, MI, USA
cpfeifer@mitre.org

ABSTRACT

Federal agencies must comply with directives expressed in documents issued by authoritative sources elsewhere in the government. To automate identification of these directives, the ADEPT (Automated Directive Extraction from Policy Texts) system exploits the observation that directive sentences are usually characterized by deontic modality (e.g. "must", "shall", etc.) permitting the open-ended task of summarizing obligations to be reduced to a well-defined and circumscribed linguistic analysis task. ADEPT comprises a linearizer, which converts deeply nested sentences into a form that can be handled by standard parsers, a deontic sentence classifier trained on an annotated corpus of sentences drawn from representative policy documents, a semantic role analyzer, and other analytic tools for extracting and analyzing the deontic content of policy documents.

1 INTRODUCTION

Modern administrative states are regulated by statutes, regulations, and other authoritative legal sources that are expressed in complex, interconnected texts. Compliance with these rules is challenging for agencies, citizens, rule-drafters, and attorneys alike. For agencies, compliance requires understanding changes in federal laws, executive orders, and authoritative directives, policies, regulations, and standards. Simply identifying and summarizing these changes, which often originate from a multitude of sources, can be a burdensome drain on staff resources. The diversity of authoritative sources imposing requirements of a given nature is typified by the proliferation of cybersecurity requirements on U.S. federal agencies. Directives can be expressed in Executive Orders, Office of Management and Budget (OMB) circulars and memoranda, Department of Homeland Security (DHS) Binding Operational Directives (BODs), National Institute of Standards and Technology (NIST) Federal Information Processing Standards (FIPS), and Special Publications (SPs). Each agency must devote staff to monitor and review multiple streams of publications to identify changes affecting their cybersecurity profile (i.e., policies, practices, procedures, standards, and/or guidance).

A similar monitoring task is required for all other areas within an agency where compliance is compulsory, such as privacy, health policy, and processing of sensitive information. An algorithmic process that automated the identification of sentences expressing obligations incumbent upon a given agency could significantly reduce the burden on staff having to review a large stream of documents. Such automated processes could provide agencies with early warnings of pending obligations, enabling them to better plan for implementation once the obligation is finalized.

A key observation of human performance on the document-monitoring task is that the summaries produced by staff typically focus on sentences that express *obligations*, i.e., that are characterized by *deontic modality*. This suggests that the tasks of monitoring and extracting directive sentences depend critically on the identification of such deontic sentences. We hypothesize that exploiting this observation will permit an important portion of the open-ended task of summarizing obligations to be reduced to a well-defined and circumscribed linguistic analysis task.

The remainder of this paper describes the design of a system for automated extraction of directives, ADEPT, and the evaluation of the critical deontic-sentence classification component. Section 2 presents examples of directives and describes the characteristics that distinguish directives from non-directives and different types of directives from one another. Section 3 discusses prior related work on modality classification, and the handling of nested directives, that is, sentences where dependent clauses or sentential complements share a common root clause is discussed in Section 4. Section 5 sets forth ADEPT's approach to identifying and classifying directive sentences, and Section 6 describes the use of semantic role labeling and frame instantiation to extract structured knowledge from sentences identified as directives. The implemented ADEPT architecture is described in Section 7, and Section 8 summarizes and outlines future efforts.

In: Proceedings of the Workshop on Artificial Intelligence and the Administrative State (AIAS 2019), June 17, 2019, Montreal, QC, Canada.

Published at http://ceur-ws.org

2 DIRECTIVE SENTENCES IN POLICY DOCUMENTS

ADEPT is based on an analysis of the work products of subject matter experts engaged in monitoring federal policy documents originating from the authoritative sources such as those listed in Section 1. Analysis of these sentences revealed that directives typically consist of expressions of obligations on the part of an agency or other government entity to perform or refrain from some specified actions, such as:

(1) Agencies must establish performance goals.

(2) Agencies are required to provide narrative responses regarding their risk management decision process.

(3) Each agency business owner is directed to ensure that 3DES and RC4 ciphers are disabled on mail servers.

(4) Chief Information Officers are to submit a report within 180 days.

These directive sentences can be viewed as illocutionary [3] or performative texts [22] that make a given action compulsory for a given government entity (i.e., the agency or a holder of a role within the agency). Frequently, as in sentence 1 above, directive sentences use modal verbs, such as “must”, “shall”, “may”, and “should”, as auxiliaries [20]. However, sentences 2–4 illustrate that obligations can be expressed without the use of modal verbs.

In addition to these *absolute*, i.e., *unqualified*, sentences, there are two other types of sentences that are important for some, but not all, applications.

First, some directives are *qualified* in the sense of expressing either *permission* or *weak necessity*, as in the following two sentences:

(5) Senior executives may consider delaying awarding new financial assistance obligations (permission).

(6) Agencies should establish and report other meaningful performance indicators and goals (weak necessity).

Second, some sentences merely report an obligation created by a different document, rather than creating an obligation themselves, such as:

(7) Section 1 of the Executive Order requires agency heads to ensure appropriate risk management.

We term such sentences *indirect obligation sentences.*

We exclude sentences from our set of directive sentences those that specify the details of an obligation created in a different sentence, e.g., by elaborating on the requirements of a work product obligation:

(8) Reports must enumerate performance goals.

We treat these sentences as non-directives because they provide details of obligatory actions but do not in themselves create an obligation for an agency or other government entity. We defer handling of these sentences to future applications.

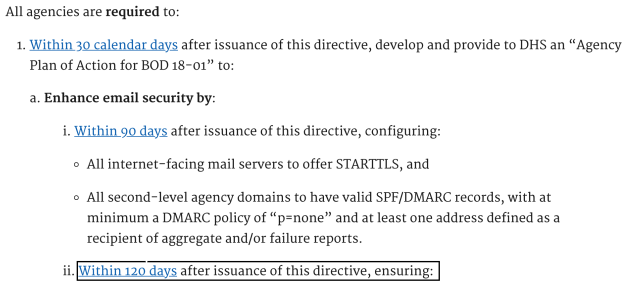
In summary, we found that directive summaries extracted from policy documents by human experts typically have deontic force, which may be absolute, qualified, or indirect, depending on the construction of the sentence. We hypothesize that summaries consisting of these deontic sentences closely match existing work products by agency personnel who currently monitor such documents and that summaries of this type could benefit agencies by enabling agency personnel to quickly identify the impact of new obligations, improving an agency's capability for complete and timely compliance.

3 RELATED WORK

Providing assistance to agencies in complying with complex regulatory and policy constraints is increasingly recognized as an important AI application. Typical examples include development of knowledge acquisition techniques to increase the agility in public administration [4] and information retrieval techniques optimized for regulatory texts [6]. Research in this area has addressed both cross-document relationships among regulatory and statutory texts, such as network structure [14], and within-document analysis, such as discourse analysis of regulatory paragraphs [5] and parsing statutory and regulatory rule texts into a computer-interpretable form [24]. The work most closely related to the objectives of the current work is [17], which addressed sentential modality classification of sentences in financial regulation texts.

A number of previous research projects have addressed the general task of modal sense disambiguation in legal and government texts. Marasović and Frank [15] developed a classifier for *epistemic*, *deontic,* and *dynamic* modal categories in English and German using a one-layer convolutional neural network (CNN) with feature maps and semantic feature detectors, reporting better results than with MaxEnt or a one-layer neural network. O'Neill et al. [17] combined a neural network with both legal-specific and more general distributional semantic model representations to distinguish among the deontic modalities *obligation*, *prohibition*, and *permission*. Wyners and Peters [19] used a rule-based approach to extract conditional and deontic rules from the U.S. Federal Code of Regulations. They found that this approach worked well for a specific set of regulatory texts, but its generality is unclear. Maat et al. [7] compared machine learning approaches to knowledge-based approaches for legal text classification in Dutch legislation, finding that while machine learning classifiers performed as well as the pattern-based model, the pattern-based approach generalized better than the machine learning model to new texts.

The modality classification task addressed by ADEPT differs from this prior work in that it focuses on the deontic distinctions relevant specifically for the task of extracting and summarizing the directives from administrative and policy documents, e.g., distinguishing deontic from non-deontic sentences and distinguishing among the categories of deontic sentences relevant to a particular application (e.g., absolute and qualified obligations). As discussed below, ADEPT additionally addresses tasks both upstream from deontic sentence detection, such as linearization of nested directive sentences, and downstream, such as instantiation of obligation frames and conversion of instantiated frames into a structured form useful to agency personnel.



All agencies are **required** to:

1. Within 30 calendar days after issuance of this directive, develop and provide to DHS an "Agency Plan of Action for BOD 18-01" to:

    a. **Enhance email security by**:

        i. Within 90 days after issuance of this directive, configuring:

        - All internet-facing mail servers to offer STARTTLS, and
        - All second-level agency domains to have valid SPF/DMARC records, with at minimum a DMARC policy of "p=none" and at least one address defined as a recipient of aggregate and/or failure reports.

        ii. Within 120 days after issuance of this directive, ensuring:

Figure 1: A typical nested directive sentence. By itself, punctuation is insufficient to disambiguate whether the phrase in the box is a child of "Enhance email security ..." or "Within 30 calendar days ...". Either indentations or enumeration/itemization marks are required to resolve this ambiguity.

4 HANDLING NESTED DIRECTIVES

Authoritative administrative texts, including directives, regulations, and statutes, are often expressed in the form of nested enumerations, such as the directive set forth in Figure 1. Nested structures are characterized by multiple dependent clauses or sentential complements to common superordinate clauses. Such structures are intended to express complex rules and directives in a compact and comprehensible style by reducing textual redundancy. Human readers can easily understand the logical structure of such sentences because the relationships among clauses are signaled by hierarchical relations between varying levels of enumeration symbols, punctuation marks, and varying indentation depths.

Unfortunately, parsers trained on standard treebanks, which are generally based on articles from news sources such as the Wall Street Journal, are often unable to process sentences with nested enumerations [16]. Thus, until domain-specific treebanks have been developed for legal texts which include nested sentences, it will remain necessary to convert such sentences into a logically-equivalent representations that are more amenable to conventional parsers.

One approach to simplifying the syntactic structure of nested enumerations is to convert them into a series of unnested sentences "by starting from the root of the tree and by concatenating, for each possible path, the phrases found until the leaves are reached" [9]. Each depth-first traversal of this tree is a simple (non-compound) sentence. We refer to this process as *linearization*. For example, the first sentence in a linearization of the nested sentence shown in Figure 1 is:

(9) All agencies are required to within 30 calendar days after issuance of this directive, develop and provide to DHS an agency Plan of Action for BOD 18-01 to enhance email security by within 90 days after issuance of this directive configuring all internet-facing mail servers to offer STRTTLS.

Linearization of regulatory and statutory text can be complicated by ambiguity in the scope of logical connectives that can arise from inconsistencies in expressing conjunction and disjunction in legal texts [1]. Nested directives, on the other hand, appear to generally be implicitly conjunctive, so linearization into a set of separate individual directives, each corresponding to a path in the depth-first traversal of the tree representing the logical form of the sentence, is generally consistent with the intended semantics of the original nested form.

As a practical matter, the greatest challenge in documents published in PDF (the primary format used by the agencies that we support) is determining the nesting level of each constituent clause with respect to surrounding clauses. Text extracted using standard tools, such as Apache Tika [2] and Tesseract [23], does not reliably retain the indentation depths of the original PDF. Punctuation marks often signal the nesting level, e.g., a clause that ends with a colon is to be followed by one or more subordinate (more deeply nested) clauses, and a period usually indicates a leaf node. However, there is an inherent ambiguity in sentences that follow a leaf node, such as the sentence in the box in Figure 1: "Within 120 days after issuance of this directive, ensuring:". Without either an unambiguous indication of indentation depth relative to surrounding clauses or an enumeration mark signaling a clear relationship to other lines of enumerated text, it is impossible to determine whether this sentence is (1) at the level of the sentence that starts "Within 90 days", (2) at the level of the sentence that starts "Enhance email security by:", or (3) the start of a new nested expression.

The lack of accurate indentation depths in text extracted from PDF documents and the ambiguity of the typical punctuation conventions suggest that the enumeration and bullet symbols and punctuation must be the source of nesting information. After all, these are generally unambiguous for human readers. Unfortunately, there is no canonical hierarchical practice of enumerations and bullets; document conventions vary not just among agencies but often within the same issuing agency as well from one document to the next. Enumeration and bulleting formats are sometimes applied inconsistently even within the same document. Our strategy is therefore to make an initial traversal of each document, recording the order of occurrence of each of a standard set of possible enumeration styles and conventions to establish a given document's hierarchical structure in each section. Each nested expression is then replaced with its linearized equivalent as determined from the hierarchy determined in the initial pass. The Appendix sets forth this procedure in more detail.

Our approach differs from Dragoni et al. [9], which mapped enumerated propositions onto a legal ontology to define the domain of directives and their constituent subparts, in using a concept-agnostic approach that may be better suited for domains in which directives are frequently revised, rescinded, or recontextualized in ways that may not be amenable to previous ontologies.

The extraction tools described below are intended to remove reference footnotes, HTML links, page numbers, and other extraneous information from within the span of single extracted sentences, but remaining bits of extraneous text create challenges for NLP processes downstream in our pipeline, such as POS and dependency parsing, event extraction, and modality detection. The last step of the linearization component therefore attempts to push these remaining items to the bottom of the linearized document as standardized endnotes.

Table 1: The proportion of sentences of each of the 3 directive types and of non-directive sentences having a modal auxiliary.

Type	Ratio	Percent
Absolute	461/592	77.8%
Qualified	378/461	82.0%
Indirect	42/103	40.8%
Non-directive	346/1426	24.3%
Total	1227/2582	47.5%

5 DIRECTIVE SENTENCE CLASSIFICATION

Our working hypothesis is that policy-document summaries consisting of some or all categories of directive sentences described above can be a proxy for, assist in the creation of, or supplement manually-created compliance summaries. Thus, we focus on classifying sentences with respect to these directive sentence categories.

5.1 Directive-Sentence Corpus

Unfortunately, none of the models or corpora developed in the prior work on sentence modality classification described above are directly applicable to our task. We therefore found it necessary to develop a new annotated directive sentence corpus based on U.S. executive-branch policy directives. Our initial focus was on OMB Memoranda and DHS Binding Operational Directives, for which we had examples of agency work products. We downloaded 5 years of OMB directives from the White House website.[1]

Each of the documents in the corpus was originally published in PDF format, usually with the first page scanned and signed. Each document was converted to plain text using the Apache Tika software package [2]. In parallel, each document was processed with Grobid [11] to identify elements such as headers and footers that can interrupt text that spans from one page to the next. The elements identified using Grobid were disinterleaved from the main text and concatenated at the end of each document.[2]

As described in Section 4, policy documents often contain complex sentences, including bullet-pointed lists and enumerations, that establish multiple distinct obligations. Accordingly, each nested sentence in the corpus was converted into a set of simple sentences using the linearization process described in Section 4. Each of the resulting sentences was then annotated according to the categories set forth in Section 2 by several annotators, including a subject-matter expert and several linguists.

The resulting set of 2,582 labeled sentences served as ground truth in the construction of the machine learning-based models described below. The mean length of these sentences was 38 tokens. Table 1 shows the proportion of sentences of each of the 3 directive types that have a modal auxiliary.[3] These ratios illustrate that the presence of modal auxiliaries is neither necessary nor sufficient for directives in this domain.[4]

P	R	F1	ROC Area	Class
0.784	0.809	0.796	0.934	Absolute
0.795	0.720	0.755	0.854	Qualified
0.574	0.301	0.395	0.771	Indirect
0.845	0.898	0.871	0.855	Non-Directive
0.812	0.818	0.812	0.866	Weighted Avg.

Table 2: Four-category deontic sentence prediction accuracy.

5.2 Evaluation of Deontic Sentence Classification

We converted each sentence of our corpus into a vector of semantic role values using AllenNLP [10]. These vectors were converted to ARFF format[5] and evaluated in 10-fold cross-validation using the Weka [12] implementation support vector machine (SVM) (Platt's algorithm for sequential minimal optimization [13] [21]). As shown in Table 2, a mean F-score of 0.812 was achieved across all four categories. A mean F-score of 0.846 (with ROC Area of 0.689) was obtained for the binary task of distinguishing non-directives from any of the 3 types of directive sentences.

This experiment indicates that the deontic categories of relevance to our task can be distinguished by a model trained on a corpus of modest size. We anticipate that this accuracy can be improved by expanding the annotated data set size and refining the text extraction and linearization processes that provide input into the classifier.

6 SEMANTIC ROLE LABELING AND TEMPLATE INSTANTIATION

For many agency applications, the most useful representation of directives is often in the form of structured tables or spreadsheets summarizing multiple sentences. Analysis of representative work products indicated that the information of interest from each sentence includes the following:

- Actor - the agency or office to which the obligation applies
- Activity - the activity that is required of the Actor
- Object - the work product to be produced by the Activity, if any
- Time - any time-related qualification of the directed activity
- Manner - any non-time-related qualification of the directed activity
- Modal - whether the activity is obligatory, permitted, or suggested, as indicated by the particular modal or other verb used to convey the deontic character of the expression, i.e., "must" vs. "may."

For each directive, we instantiate a frame containing argument slots for each of the types of information above. For example, the instantiated frame shown in Table 3 summarizes the key information from the following directive sentence:

(10) Within 60 days of this Memorandum's publication agencies must update their list of non-governmental URLs.

[1] https://www.whitehouse.gov/omb/information-for-agencies/memoranda/

[2] Footnote texts must be retained because they sometimes contain directives.

[3] Modal verbs included can, could, may, might, must, shall, should, will, or would

[4] This annotated corpus will be made available to researchers in 2019 at http://mat-annotation.sourceforge.net/.

[5] https://www.cs.waikato.ac.nz/ ml/weka/arff.html

Table 3: An instantiated directive template.

Actor	agencies
Activity	update
Object	list of non-governmental URLs
Time	within 60 days
Modal	must

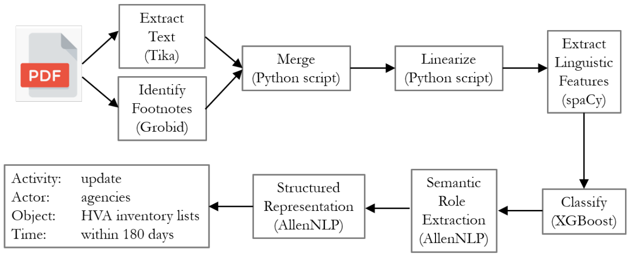

PDF
Extract Text (Tika)
Identify Footnotes (Grobid)
Merge (Python script)
Linearize (Python script)
Extract Linguistic Features (spaCy)
Classify (XGBoost)
Semantic Role Extraction (AllenNLP)
Structured Representation (AllenNLP)
Activity: update
Actor: agencies
Object: HVA inventory lists
Time: within 180 days

Figure 2: The directive sentence processing pipeline.

The slots in the directive frame are a domain-specific adaptation of standard semantic roles. We use the Semantic Role Labeling model of AllenNLP [10] to assign Propbank semantic role labels [18] to directive sentences. We then use a set of simple heuristic rules for mapping these SRLs to the slots of our frames, e.g., a Propbank “ARG0” is generally the Actor, “ARG1” is generally the Object, and “Temporal” corresponds to the Time slot. Directives expressed without a modal verb ("All agencies are required to ...") will have no entry in the "Modal" field.

7 SYSTEM ARCHITECTURE

As illustrated in Figure 2, ADEPT's directive extraction and analysis tasks require a series of processing steps. We have adopted a modular architecture that can accommodate a variety of alternative components.

The first stage of the pipeline consists of concurrent calls to the APIs of the Tika and Grobid services offered by their respective Docker [8] containers. Tika outputs the PDF extraction as plain text whereas Grobid outputs the footnotes embedded in XML. The **merge** stage integrates this content and outputs a text file consisting of disinterleaved page content followed by all footnotes. The **linearizer** takes this text as input and outputs a text file containing one **linearized** sentence per line. The **linguistic feature extractor** converts each sentence into a feature vector of n-grams and features derived from a dependency parse.

An API call to the Docker container of the AllenNLP service is then made with a JSON file containing all sentences identified as being of the target deontic type or types (e.g., *absolute*). The AllenNLP output is passed to the template instantiation stage. The final output consists of CSV and HTML files that can be loaded into a spreadsheet or viewed through a web browser.

8 DISCUSSION AND FUTURE WORK

ADEPT illustrates how a document analysis task that imposes a significant burden to a wide range of agencies—directive extraction—can be addressed by deontic sentence classification in combination with nested sentence disambiguation and semantic role labeling. We anticipate that an ADEPT directive-extraction pilot will take place in mid-2019 with a representative U.S. federal agency.

Future work will relax ADEPT's current simplifying assumption that the directive content of policy documents can be determined by analyzing individual sentences divorced from their surrounding context. For within-document contextual information, we plan to introduce entity resolution and link connecting sentences that elaborate on an obligation with the obligation sentence to which they apply. To improve cross-document contextual information, we plan to develop techniques to detect and classify references to other documents, particularly statements that the current document rescinds directives from other policy documents.

Automated analysis of policy documents presents a rich set of text-analytic tasks but promises very significant rewards to both agencies and citizens. ADEPT represents an initial realization of this approach to improving the administrative state through modern computational linguistics techniques.

ACKNOWLEDGMENTS

The MITRE Corporation is a not-for-profit company, chartered in the public interest, that operates multiple federally funded research and development centers. This document is approved for Public Release; Distribution Unlimited. Case Number 18-4602.

REFERENCES

[1] L. Allen and C. Saxon. More IA needed in AI: Interpetation assistance for coping with the problem of multiple structural interpetations. In *Proceedings of the Third International Conference on Artificial Intelligence and Law*, pages 53–61, Oxford, England, June 25–28 1991.
[2] Apache tika - a content analysis toolkit. https://tika.apache.org/. Accessed: 2018-11-16.
[3] J. Austin. *How to do things with words*. Oxford U. Press, New York, 1962.
[4] A. Boer and T. van Engers. An agent-based legal knowledge acquisition methodology for agile public administration. In *Proceedings of the 13th International Conference on Artificial Intelligence and Law*, ICAIL '11, pages 171–180, New York, NY, USA, 2011. ACM.
[5] A. Buabuchachart, K. Metcalf, N. Charness, and L. Morgenstern. Classification of regulatory paragraphs by discourse structure, reference structure, and regulation type. In *Proceedings of the 26th International Conference on Legal Knowledge-Based Systems JURIX*, University of Bologna, Bologna, Italy, November 2013.
[6] D. Collarana, T. Heuss, J. Lehmann, I. Lytra, G. Maheshwari, R. Nedelchev, T. Schmidt, and P. Trivedi. A question answering system on regulatory documents. In *Proceedings of the 31st international conference on Legal Knowledge and Information Systems (JURIX)*, 2018.
[7] E. de Maat, K. Krabben, and R. Winkels. Machine learning versus knowledge based classification of legal texts. In *Proceedings of the 2010 Conference on Legal Knowledge and Information Systems: JURIX 2010: The Twenty-Third Annual Conference*, pages 87–96, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
[8] DOCKER. https://www.docker.com/. Accessed: 2019-01-24.
[9] M. Dragoni, S. Villata, W. Rizzi, and G. Governatori. Combining NLP Approaches for Rule Extraction from Legal Documents. In *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*, Sophia Antipolis, France, Dec. 2016.
[10] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. E. Peters, M. Schmitz, and L. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640, 2018.
[11] Grobid (or grobid) means GeneRation of BIbliographic data. https://grobid.readthedocs.io/en/latest/. Accessed: 2018-12-18.
[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[13] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.

[14] M. Koniaris, I. Anagnostopoulos, and Y. Vassiliou. Network analysis in the legal domain: a complex model for european union legal sources. *Journal of Complex Networks*, 6(2):243–268, 2018.

[15] A. Marasović and A. Frank. Multilingual modal sense classification using a convolutional neural network. In P. Blunsom, K. Cho, S. B. Cohen, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, and S. W. Yih, editors, *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, pages 111–120. Association for Computational Linguistics, 2016.

[16] L. Morgenstern. Toward automated international law compliance monitoring (tailcm). Technical report, LEIDOS, INC, 2014. AFRL-RI-RS-TR-2014-206.

[17] J. O'Neill, P. Buitelaar, C. Robin, and L. O'Brien. Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law, ICAIL 2017, London, United Kingdom, June 12-16, 2017*, pages 159–168, 2017.

[18] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, Mar. 2005.

[19] W. Peters and A. Z. Wyner. Legal text interpretation: Identifying hohfeldian relations from text. In N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* European Language Resources Association (ELRA), 2016.

[20] The Plain Writing Act of 2010, 2010. 111th Congress H.R. 946.

[21] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[22] J. Searle. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press, Cambridge, 1969.

[23] Tesseract ocr. https://opensource.google.com/projects/tesseract. Accessed: 2018-11-16.

[24] A. Wyner and W. Peters. On rule extraction from regulations. *Frontiers in Artificial Intelligence and Applications*, (235), January 2011.

9 APPENDIX: LINEARIZATION OF NESTED DIRECTIVES

```
FOR each document ingested by the linearizer:
    Preprocess: Remove footnotes to prevent splitting of
    enumerated list elements or main body sentences during
    downstream processing later in the classification
    pipeline
EXTRACT strings matching footnote format
STORE matching strings in References array
DELETE matching strings in their original positions
DELETE all multiple (n-1) vertical and horizontal spacing
    Detect Document Section Boundaries: Identify positions
    of each document section to prevent enumerated elements
    from spanning multiple distinct lists.
MATCH list of known section headers
STORE matches in partition along with starting offset position for each section in index
READ any enumerated lists in between section boundaries
    Parse and Concatenate Enumerations: Map document
    hierarchical enumeration conventions against different
    symbol sets. Concatenate all directly subordinated sentence
    fragments with their subordinating fragments to
    form full (flat) sentences from the enumerated elements
    for downstream processing later in the classification
    pipeline.
MATCH lines in each enumerated list within each section against enumeration symbol style list delimited by punctuation cues (Uppercase Roman Numerals, Lowercase Roman Numerals, Uppercase Letters, Lowercase Letters, Number Digits, Solid Bullet Points, Hollow Bullet Points)
STORE the sequential order (i.e., layers) of enumeration styles encountered to set document convention, where each layer begins with its own closet set of enumeration symbols
FOR lower-order layers
CONCATENATE lines recursively with all parent layers
TERMINATE upon reaching new paragraph with no enumeration symbol at the start of the line
ITERATE over all sections
WRITE to [FILENAME]_paths.txt file
    Standardize Global Enumeration: Rewrite enumeration
    conventions to standard format (e.g. I.iii.B.a. → 1.3.2.1.)
FOR all enumerated lists,
REWRITE each line's enumeration symbol with its corresponding digit based on the layer order and within-layer order
WRITE to [FILENAME]_trees.txt file
    Post-Process Footnotes: Add previously extracted footnotes
    to the bottom of document
APPEND footnote elements to bottom of the [FILENAME]_paths.txt file under the new section header "Footnotes"
```