## Supplementary Information

In this Supplementary Information, we detail the structured profiles of the example law smells introduced in Section 3.2.

### Duplicated Phrase

*Description* A *duplicated phrase* is a phrase above a specified *length* that has more than a specified number of *occurrences* in a legal text. Here, a *phrase* is a nonempty sequence of terms, where a term is either a token (i.e., a sequence of non-whitespace characters roughly corresponding to a word) or a placeholder for an argument that is itself a phrase.

*Problem* From a lawmaker's perspective, a duplicated phrase heightens the risk of inconsistencies when the phrase needs to be changed or is used to mean different things in different places. From a lawtaker's perspective, a duplicated phrase reduces the information density of the text, making it harder to read, and it increases the risk of confusion if the phrase is used with divergent meanings.

*Detection* Via scalable variants of *n*-gram search, possibly after some text preprocessing (e.g., to replace named entities).

*Mitigation* By introducing named variables and definitions. Wherever a duplicated phrase occurs, it can be replaced by its variable name and a reference to its definition (which might be visible in the text or embedded in the markup).

*Example* The phrase "information within the scope of the information sharing environment, including homeland security information, terrorism information, and weapons of mass destruction information" occurs 26 times in the 2019 version of Title 6 of the United States Code. From a parliamentary lawmaker's perspective, if $X$ is information that they consider "within the scope of the information sharing environment" (*ISE information*) but a court decides that $X$ is *not* ISE information, the lawmaker will need to add $X$ to the list of *includes* after the first comma to get their will. In its current structure, this requires 26 changes to the text, whereas a single change will suffice if $X$ is maintained as a variable. From a lawtaker's perspective, *ISE information*, once defined by the duplicated phrase and then used consistently throughout, would be much more readable and no less precise. In the digital realm, the term could then be expandable to its full meaning, e.g., on click or on hover.

### Long Element

*Description* A *long element* is an element containing legal text that is long as assessed by some absolute measure (e.g., number of tokens) or relative measure (e.g., quantile of a token length distribution).

*Problem* A long element may indicate a lack of structure in the legal text. This increases the cognitive load for lawtakers reading the text, and it complicates maintenance on the part of lawmakers.

*Detection* Via outlier detection using automatically computed length measures and (potentially nested) distributions for different types of elements (e.g., Titles, Chapters in a Title, or Sections in a Chapter of the United States Code).

*Mitigation* By moving (with the appropriate adjustments) some of their text to new elements or to shorter elements that already exist, i.e., by adding or altering structure.

*Example* In its 2019 version, 5 U.S.C. § 552 contains more than 8 000 tokens, i.e., more than 16 normally typeset pages of text (or, more precisely: 22.5 pages according to the page break markup in the official XML of the United States Code). Its heading already indicates that the Section is a mixed bag: *Public information; agency rules, opinions, orders, records, and proceedings*.[10] This Section contains a large collection of rules related to public information, organized in up to six levels of substructure (small Latin alphabets, Arabic numbers, large Latin alphabets, small Roman numerals, large Roman numerals, small double Latin alphabets), sometimes with more than 15 substructure items on the same level. From a lawmaker's perspective, a section of this length is hard to maintain in an orderly fashion. For example, lawmakers seeking to add content related to 5 U.S.C. § 552 have incentives to simply append it to the section—there is not much order to ruin anyway, and integrating the new content into the section where it fits best would necessitate much more work. This will make the section even longer, and even harder to maintain. Since the substructure items have no headings, lawtakers understandably unwilling to read 5 U.S.C. § 552 linearly are left to navigate it mostly by keyword search, or—if they have the necessary background knowledge—by memory. Breaking up 5 U.S.C. § 552 into several sections with separate headings would allow them to restrict their search to potentially relevant text passages by making related, but relatively independent content directly accessible.

### Ambiguous Syntax

*Description Ambiguous syntax* is the use of logical operators (e.g., *and*, *or*, and *no(t)*), control flow operators (e.g., *if*, *else*, or *while*), or punctuation (e.g., commas and semicolons) in a way that leaves room for interpretation.

*Problem* The intention of the legislator is communicated ambiguously, which leads to legal uncertainty for lawtakers. Eliminating that uncertainty further creates social costs (e.g., through lawsuits).

*Detection* Via pattern matching with regular expressions, followed by a manual assessment of the potentially problematic instances.

[10] Computer scientists might be reminded of functions with long names that include connectors, such as *do_x_and_y*, which strongly indicate that these functions should be refactored.

*Mitigation* By using logical operators with their precise mathematical meaning, introducing *xor* as a shorthand for the exclusive *or* (to be clearly distinguished from the inclusive *or*), and adding brackets as syntax (e.g., round brackets to clarify operator binding, curly brackets to denote sets, rectangular brackets to denote lists).

*Example* Many sentences in the United States Code feature multiple instances of *and* or *or* as connectors. For example, the second sentence of 12 U.S.C. § 5538 (a) (1) in 2019 reads: "Such rulemaking shall relate to unfair or deceptive acts or practices regarding mortgage loans, which may include unfair or deceptive acts or practices involving loan modification and foreclosure rescue services." Shall the referenced rulemaking relate to *(((unfair or deceptive) acts) or (practices regarding mortgage loans))* or to *((unfair or deceptive) (acts or practices)) regarding mortgage loans))*? Can these loans include *(((((unfair or deceptive) acts) or (practices involving loan modification)) and (foreclosure rescue services))* or *(((unfair or deceptive) (acts or practices)) involving (loan modification and foreclosure rescue services))*? Without clarifying syntax, from the text alone, we can assign higher or lower probabilities to the individual possibilities based on our estimates of intended sentence semantics, but we cannot retrieve the accepted meaning without consulting external sources.

At the other end of the ambiguity spectrum, the widespread uses of "and/or" (e.g., 7 U.S.C. § 451: "interstate and/or foreign commerce") and "X, or Y[,] or both" (e.g., 26 U.S.C. § 9012: "shall be fined not more than $5,000, or imprisoned not more than one year or both") in the United States Code are prime examples of redundant syntax leading to unnecessary verbosity (and distressed mathematicians). Legally binding usage of *or* for inclusive and *xor* for exclusive options would allow lawmakers to replace "and/or" by "or" and do away with "or both", no ambiguities remaining.

### Large Reference Tree

*Description* A *reference tree* rooted at an element of law $r$ is a tuple $T_r = (V_r, E_r)$, where $V_r$ is the set of elements of law reachable from $r$ by following references (including $r$), and $E_r$ is a minimal set of edges (references) such that each element of $V_r$ can be reached from $r$. A reference tree is *large* if its edge set exceeds a given *size*.

*Problem* From a lawmaker's perspective, large reference trees may lead to unforeseen normative side effects, e.g., when a leaf element is changed. From a lawtaker's perspective, large reference trees increase the cognitive load involved in navigating a legal text.

*Detection* By traversing adequately preprocessed directed multigraphs that represent legal document networks.

*Mitigation* By making references as specific as possible (i.e., not referencing a higher-level container if all referenced content is captured by a lower-level container), or by restructuring the text and references contained in the tree elements (e.g., introducing lower-level containers that then can be referenced more precisely).

*Example* Many large reference trees can be found in tax law, e.g., in *Title 26—Internal Revenue Code* of the United States Code. For example, 26 U.S.C. § 62(a)(20) references 26 U.S.C. § 62(e), which points to Sections spread across multiple Titles, and 26 U.S.C. § 751(c)(2) references many Sections within Title 26. More generally, provisions listing taxable and tax-exempted sources of income often point to definitions of these sources, which then point to further definitions and make further exceptions in their own text. The resulting web of definitions, duties, and exemptions is hard to navigate for lay lawtakers, who need to rely on special-purpose applications (i.e., alternative user interfaces for parts of the United States Code) to determine their tax liability.

### Natural Language Obsession

*Description* *Natural language obsession* is the representation of typed data as natural language text, often without a standardized format that could facilitate their algorithmic reconstruction.

*Problem* When represented using inconsistent natural language, typed data is notoriously hard to parse and maintain. Lawmakers forgo the benefits of type and consistency checking (e.g., if two interest rates stated in different parts of the text should always be equal), lawtakers are left without semantic highlighting and expandable abbreviations, and systematic analysis of typed data usage becomes impossible.

*Detection* Using Named Entity Recognition (NER) methods (including pattern matching with regular expressions), potentially augmented or validated by scalable *n*-gram search techniques.

*Mitigation* By introducing a data layer that is separate from the text (representation) layer, using strong types for named entities, and associating type checking, highlighting, and data analysis rules with these types.

*Example* In the United States Code, punishments are often expressed using variants of the high-level pattern "shall be fined not more than {money} or imprisoned not more than {period}", and temporal requirements of duties to act are frequently stated through variants of the high-level pattern "not later than {period} after {date}". Reading, maintaining, and analyzing the rules following these patterns (as a human or a computer) becomes much easier if the amounts of money, time periods, and dates are unambiguously identified as such, follow a common format, and are readily available for analysis in their natural units (e.g., amounts of money as positive integers in national currency units or time periods as positive integers in units of days). This can be achieved by storing typed data in a metadata layer supplementing the legal text, using their natural scales and units.