Generating Original Structure in Regulatory Documents

Steven O. Kimbrough, Thomas Y. Lee, Balaji Padmanabhan, Yinghui Yang
University of Pennsylvania, 3730 Walnut Street, Suite 500, Philadelphia, PA 19103
{kimbrough, thomas.lee, balaji, yiyang}@wharton.upenn.edu

1.0 Introduction

As technology and society continue to evolve, the size of the corpus of government policies and procedures continues to more than keep pace. The U.S. Federal Tax Code today consumes over 2.8 million words or 6000 pages. There are more than 20,000 cross-references both within the code itself and to external regulations. Navigating the sea of information is a daunting task for the IRS let alone a well-intentioned tax payer, or policy-maker seeking to eliminate redundancies, inconsistencies, or loopholes. While tools for tasks such as compliance checking or query answering have long held promise, automated reasoning, however intelligent, needs something to reason upon, a formalized knowledge base of some kind. In other domains people may be the primary targets of knowledge engineering; in the policy realm, much of the requisite knowledge resides in legal and regulatory documents.

In general, we may say that there have been three main approaches to extracting formalized knowledge from documents of legal and regulatory interest. (1) *Manually symbolize a relevant corpus*. The approach here is to pick an appropriate formal representation language and manually (perhaps with computerized support) symbolize the essential information from the relevant collection of documents, typically statutes, administrative rules, and legal cases. (2) *Accept minimally structured documents*. Under this approach, the documents in the chosen corpus are formalized only in a very weak sense, e.g., by creating inverted files of their terms and limiting inference to what Information Retrieval techniques can produce. (3) *Automatically structure a relevant corpus*. Under this approach, pattern-finding programs extract structure from documents in a target corpus and create derived documents, such as in XML, which present their structures more transparently.

The three approaches define a solution space that trades cost for inferential acuity. Manual symbolization affords the best prospect for deep and detailed automated inferencing and information recovery, yet it is the most labor intensive option and presents serious problems of maintenance. Accepting minimally structured documents is the least expensive alternative and the least powerful in terms of potential to support inferencing. Automated structuring lies more or less between the other two approaches.

Our objective is to expand the solution space rather than to seek the best compromise for a given application. We propose to shift the operating frontier by governing how documents are *originally created*: beginning with the initial draft, write the documents in such a way that, from the initial draft, they have the requisite structure to support automated inferencing. We seek to create documents that are equivalent in their expressiveness to those which are manually coded as per (1) using semistructured models that afford more complex reasoning than automated approaches as in (3).

This project is therefore about a fourth approach to formalizing the knowledge in legal and regulatory documents: *create* documents in a structured form rather than attempting to impose structure later. The intuition is to write using formal sublanguages so that structure emerges as a *by-product* of the normal, policy-making process. More specifically, in restricted domains of import to systems of law and government, it may be feasible to draft policies and regulations using formal, special-purpose (artificial sub-) languages and vocabularies. We proceed with a brief introduction to artificial sub-languages and then turn to their potential in drafting fully-structured and semistructured documents that support automated reasoning.

2.0 Artificial Languages

Artificial languages, *constructed languages*, and *planned languages* are all terms used to describe languages that are designed for ease of use and that are sufficiently expressive for particular purposes. intended application. An artificial *sublanguage* focuses on those elements "used by a particular community of speakers, say, those concerned with a particular subject matter or those engaged in a specialized occupation (Sager, 1986 p. 2)." The U.S. Navy employs the simple vocabulary and constrained grammar of a telegraphic sublanguage that exemplifies those that are in wide use today. Examples from the U.S. Navy telegraphic sublanguage include "72 manhours expended," "work request submitted," and "no parts required (Fitzpatrick et al., 1986 p. 45)."

Our challenge is to formalize artificial sublanguages so that machines may productively conduct inferences using them. ACLs (agent communication languages) provide a theoretical approach to the problem. Artificial languages reveal techniques and requirements for ACLs. From a practical point of view, formalizing artificial sublanguages could afford human-machine and machine-human communication, including language translation and error detection, as well as machine-machine communication. Perhaps of greatest immediate benefit, however , are opportunities for automated recovery and discovery of information. We focus on this form of application below.

Also known as ``English for maritime communications," Seaspeak is an artificial sublanguage for merchant shipping. It is a restricted, English-like language adopted in 1988 by the International Maritime Organization (IMO) of the United Nations. Seaspeak was created by specialists in maritime communications and applied linguistics for speech communication between ships and from-ship-to-shore, where clear, brief, unambiguous communications are critical (Johnson, 2002).

We are currently attempting to formalize Seaspeak based upon FLBC (Kimbrough, 1990; Kimbrough and Tan, 2000) and the *Seaspeak Training Manual* (Weeks et al., 1988). FLBC is an evolving research program aimed at developing logic-based formal languages for business communications. The central concept to encode in FLBC, is that of a Seaspeak message marker. There are seven markers that include Information, Warning, and Request (Weeks et al., 1988). Example messages include: *ADVICE: Anchor, positin: bearing: one-nin-our degrees true, from Keel Point distance: one mile* or *INSTRUCTION: Go to berth number: two-five*.

3.0 Semistructured data

As noted in the Introduction, from the perspective of information extraction, policy documents reside along a continuum from unstructured free-text to highly-structured information represented in a formal, symbolic language. Between these two poles likes the broad range of semistructured documents, for which formal theories of management and manipulation are still evolving (Abiteboul et al., 2000).

Semistructured documents are collections of data that comform to some instance of a semistructured data model. The semistructured data model is defined as self-describing, or more formally, as collections of label-value pairs (Abiteboul et al. 2000). Taken in this context, a document encoded in HTML is an instance of a semistructured data model that captures information about the structural elements with which documents are formed, certain relationships between those structural elements, and certain characteristics about how those elements are visually presented. For example, we know that the title of the document is the value associated with the HTML label (tag) "title". We know that the text associated with an ``H2" label bears a part-of relationship to the text of the "H1" label in which the H2 content nests.

A sublanguage is comprised of a lexicon and a grammar. The grammar defines how lexicon elements may be composed. In the context of semistructured data, the lexicon defines the set of possible labels and their corresponding semantics. The grammar constrains how labels relate to one another. Not all sublanguages are complete, however. The vocabulary might be incomplete. There might be concepts or content that cannot be represented in the lexicon and/or grammar.

While the eXtensible Markup Language (XML) is not itself an instance of a semistructured data model, it is a metalanguage for defining semistructured models. The Document Type Definition (DTD) for an XML instance such as ebXML, however, is one example of an incomplete language for describing the terms used in electronic business and their corresponding relationships to one another. Because even ebXML is incomplete, leaf nodes in these semistructured models allow for free text to capture concepts and relationships not encoded in the formal vocabulary and grammar.

A complete language in the semistructured context, then, is one in which all information is captured in labels and all terminal values are empty. A complete language would eliminate the need for free text. As an aside, we note that any incomplete sublanguage could be interpreted as a complete language by simply discarding free text in label values of the corresponding semistructured model. For example, Seaspeak defines seven labels (and their corresponding response markers) that can appear as the root of a semistructured document. Other terms from both specialized and general maritime use are defined. Moreover, Seaspeak is an example of an incomplete language because it does ultimately allow for the inclusion of terms from the general, English vocabulary.

4.0 Exploiting structures

Once the structure of legal documents is made explicit, be it fully formal or semistructured, there are interesting possibilities for manipulating and reasoning over the content. We might, for example, attempt to align bureaucratic regulations with the policies that they are intended to enact. By comparing policies and regulations to their intended application contexts, data mining models, such as C4.5 can be built to distinguish conditions that result in a favorable outcome with those that result in unfavorable outcomes (Quinlan, 1993). For example, the classification model built may show that if there is a specific set of actions taken, then favorable outcomes usually result.

Learning interesting patterns are a second way in which data mining methods are applicable to regulatory documents written in a formal sublanguage. By referencing documents with incident reports where the policies are applied or enforced, one might discover that certain, recommended actions have unintended effects. Unintended consequences are examples of ``interesting'' patterns that can be learned from such formalizations (Padmanabhan and Tuzhilin, 1998).

Finally, given the emergence of particular patterns or perhaps in direct response to questions regarding issues like compliance, regulatory documents in structured or semistructured form facilitate direct queries. For example, patterns relating certain policies and policy measures would have revealed the 1921 Martin Act as a vehicle now being employed by Attorney General Eliot Spizter in prosecuting Wall Street fraud. Cross references between different policies and regulations could be traced using the transitive closure. We could check for the internal consistency and quality of a single policy document by defining the concepts of well-formedness and validity with respect to the lexicon and grammar of the sublanguage. Active database concepts like event-condition-action rules could automatically update indexes to support robust querying through regulations and their periodic updates (e.g. the U.S. Federal Tax Code).

5.0 Summary and Conclusion

The present document is a position paper. It advocates a vision in the development of computerized support for reasoning over policy documents. The elements of the story are as follows.

1. Original creation of documents (and records) of legal import in either semistructured or fully formalized format offers the prospect of greatly reducing the cost and expanding the scope of knowledge engineering for legal reasoning.

2. Such origination of legal documents will likely rely on special-purpose sublanguages. If a domain is sufficiently important and well specified to support a sublanguage, even an informal one, that sublanguage becomes a potential target for full or partial formalization. Seaspeak is one such example.
3. Sublanguages will vary in their expressiveness and in their completeness along dimensions of lexicon (vocabulary including semantics) and grammar (syntax).
4. A more limited sublanguage, perhaps specified in an XML DTD, will result in a semistructured document.
5. A more complete sublanguage, perhaps specified in in a broadly logical language such as the FLBC discussed above, may result in a fully structured document.
6. We can exploit the structure of documents (the grammar) in several ways. At a high level, we can do three things: we can build specific data mining models, we can mine the corpus for interesting, unexpected facts and associations, and we can query for explicit items of interest.

References

(Abiteboul et al. 2000) S. Abiteboul, P. Buneman, and D. Suciu. *Data on the web: from relations to semistructured data and XML*. Morgan Kaufmann, San Francisco, CA, 2000.

(Fitzpatrick et al., 1986) E. Fitzpatrick, J. Bachenko, and D. Hindle. "The status of telegraphic sublanguages," In R. Grishman and R. Kittredge, editors, *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*, p. 39-51. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1986.

(Johnson, 2002) E. Johnson. "Talking Across Frontiers," In *Proceedings of the International Conference on European Cross-Border Cooperation*. http://www.prolingua.co.uk/talking.pdf, accessed December, 2002.

(Kimbrough, 1990) S. O. Kimbrough. "On Representation Schemes for Promising Electronically," *Decision Support Systems*, 6(2):99-122, 1990.

(Kimbrough and Tan, 2000) S. O. Kimbrough and Y. H. Tan. "On Lean Messaging with Unfolding and Unwrapping for Electronic Commerce," *International Journal of Electronic Commerce*. 5(1):83-108, 2000.

(Linguanet, 2003) Linguanet, *The Linguanet Project*. http://www.cbs.dk/departments/fir/linguanet/, accessed 28 February 2003.

(Padmanabhan and Tuzhilin, 1998) B. Padmanabhan and A. Tuzhilin. "A Belief-Driven Method for Discovering Unexpected Patterns," *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. p. 94-100. ACM Press, August 1998.

(Quinlan, 1993) R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

(Sager, 1986) N. Sager. "Sublanguage: Linguistic Phenomenon, Computational Tool," In R. Grishman and R. Kittredge, editors, *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*, p. 1-17. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1986.

(Weeks et al., 1988) F. Weeks, A. Glover, E. Johnson, and P. Strevens. *Seaspeak Training Manual: Essential English for International Maritime Use*. Pergammon Press, Oxford, UK, 1988.