

MIT Open Access Articles

*Law Is Code: A Software Engineering
Approach to Analyzing the United States Code*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Li, William, Pablo Azar, David Larochelle, Phil Hill, and Andrew W. Lo. "Law Is Code: A Software Engineering Approach to Analyzing the United States Code." *Journal of Business & Technology Law* 10(2) (2015).

As Published: <http://digitalcommons.law.umaryland.edu/jbtl/vol10/iss2/6/>

Publisher: University of Maryland School of Law

Persistent URL: <http://hdl.handle.net/1721.1/99465>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Law Is Code: A Software Engineering Approach to Analyzing the United States Code

William Li

Pablo Azar

David Larochelle

Phil Hill

Andrew W. Lo

Follow this and additional works at: <http://digitalcommons.law.umaryland.edu/jbtl>

Recommended Citation

William Li, Pablo Azar, David Larochelle, Phil Hill, & Andrew W. Lo, *Law Is Code: A Software Engineering Approach to Analyzing the United States Code*, 10 J. Bus. & Tech. L. 297 (2015)

Available at: <http://digitalcommons.law.umaryland.edu/jbtl/vol10/iss2/6>

This Article is brought to you for free and open access by the Academic Journals at DigitalCommons@UM Carey Law. It has been accepted for inclusion in Journal of Business & Technology Law by an authorized administrator of DigitalCommons@UM Carey Law. For more information, please contact smccarty@law.umaryland.edu.

WILLIAM LI,^{*} PABLO AZAR,^{**} DAVID LAROCHELLE,^{***} PHIL HILL,^{****} AND ANDREW W. LO^{*****†}

Law Is Code: A Software Engineering Approach to Analyzing the United States Code

ABSTRACT

The agglomeration of rules and regulations over time has produced a body of legal code that no single individual can fully comprehend. This complexity produces inefficiencies, makes the processes of understanding and changing the law difficult, and frustrates the fundamental principle that the law should provide fair notice to the governed. In this Article, we take a quantitative, unbiased, and software-engineering approach to analyze the evolution of the United States Code from 1926 to today. Software engineers frequently face the challenge of understanding and managing large, structured collections of instructions, directives, and conditional statements, and we adapt and apply their techniques to the U.S. Code over time. Our work produces insights into the structure of the U.S. Code as a whole, its strengths and vulnerabilities, and new ways of thinking about individual laws. For example, we identify the first appearance and spread of important terms in the U.S.

© 2015 William Li, Pablo Azar, David Larochelle, Phil Hill, and Andrew W. Lo

^{*} William Li is a Ph.D. student in the Computer Science and Artificial Intelligence Laboratory (CSAIL) and a 2012 graduate of the Technology and Policy Program at the Massachusetts Institute of Technology (MIT).

^{**} Pablo Azar is a doctoral student in Economics at the Massachusetts Institute of Technology (MIT) and a 2014 Computer Science Ph.D. graduate of MIT.

^{***} David Larochelle is an engineer at the Berkman Center for Internet & Society at Harvard University.

^{****} Phil Hill is an associate with Kirkland & Ellis LLP. He is a past Fellow at the Berkman Center for Internet & Society at Harvard University and a 2013 J.D. graduate of Harvard Law School.

^{*****} Andrew W. Lo is the Charles E. and Susan T. Harris Professor at the MIT Sloan School of Management, Principal Investigator in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at the Massachusetts Institute of Technology (MIT), and a joint faculty in the MIT Electrical Engineering and Computer Science Department.

[†] We thank seminar participants at Columbia University, Indiana University, the International Monetary Fund, MIT's Computer Science and Artificial Intelligence Laboratory, Harvard's Berkman Center for Internet & Society, the 2015 New England Database Day conference, the Pacific Institute of Mathematical Sciences, and Yale Law School for helpful comments and discussion. Research support from the MIT Laboratory for Financial Engineering is gratefully acknowledged. We also thank Jayna Cummings, the *Journal of Business & Technology Law's* editor-in-chief, Nicholas R. Rodriguez, and the *Journal's* editorial board for many helpful editorial comments.

LAW IS CODE

Code like “whistleblower” and “privacy.” We also analyze and visualize the network structure of certain substantial reforms, including the Patient Protection and Affordable Care Act and the Dodd-Frank Wall Street Reform and Consumer Protection Act, and show how the interconnections of references can increase complexity and create the potential for unintended consequences. Our work is a timely illustration of computational approaches to law as the legal profession embraces technology for scholarship in order to increase efficiency and to improve access to justice.

I. INTRODUCTION

LAWS AND REGULATIONS ARE THE RULES BY WHICH SOCIETIES OPERATE. Beginning with the Code of Ur-Nammu more than 4,000 years ago,¹ societies have often formalized laws and regulations by recording them in written form.² Over time, this simple custom evolved, producing some of the most significant innovations in the history of civilization, including replacing the rule of monarchs with the rule of law.³

With the rule of law flourishing in modern societies, subtle challenges have emerged as unintended consequences of an unwieldy system of laws. The agglomeration of rules and regulations over time and across the many facets of social, political, and economic interactions has produced a body of legal code that no single individual can fully comprehend. Despite the fact that laws now apply to virtually every aspect of daily life, the sheer volume of code requires citizens to have a certain degree of faith in the experts with whom they have entrusted the responsibilities of creating, managing, analyzing, and ultimately applying that code.

The increasing complexity of the legal system has several important implications. First, it produces inefficiencies.⁴ The time, money, and other human resources associated with the rule of law in modern society are substantial and growing.⁵ Second, as the legal code expands in size, interactions between provisions will

1. See J.J. Finkelstein, *The Laws of Ur-Nammu*, 22 J. CUNEIFORM STUD. 66 (1968–69), available at <http://www.jstor.org/discover/10.2307/1359121?uid=3739696&uid=2&uid=4&uid=3739256&sid=21104628208137> (reprinting Ur-Nammu’s legal code).

2. See H. PATRICK GLENN, *LEGAL TRADITIONS OF THE WORLD: SUSTAINABLE DIVERSITY IN LAW* 100–01, 135 (4th ed. 2010).

3. See, e.g., Russell Fowler, *The 800th Anniversary of Magna Carta: A Time for Lawyers to Remember*, 50 TENN. B.J. 23 (2014) (commemorating the Magna Carta, which limited the British monarch’s power).

4. Susan Hayes Stephan, *Blowing the Whistle on Justice as Sport: 100 Years of Playing a Non-Zero Sum Game*, 30 HAMLINE L. REV. 587, 588 (2007) (analogizing litigation to an inefficient “game”).

5. See LAWYERS FOR CIVIL JUSTICE ET AL., *LITIGATION COST SURVEY OF MAJOR COMPANIES 2–6* (2010), available at <http://www.uscourts.gov/uscourts/RulesAndPolicies/rules/Duke%20Materials/Library/Litigation%20Cost%20Survey%20of%20Major%20Companies.pdf>.

quickly outstrip humans' ability to manage them using traditional methods.⁶ Third, if one purpose of a legal code is to provide notice to the governed, then that purpose is frustrated when the code becomes increasingly opaque to the vast majority of citizens.⁷

In this Article, we propose new quantitative methods for understanding and managing the system that comprises the entire legal code. We start from the well-trodden premise that legal code is in many respects similar to computer source code.⁸ We take a computational approach to studying the full text of the United States Code ("U.S. Code") from its first edition in 1926 to the present day. Our approach adopts techniques that software engineers use to analyze the evolution and structure of large software codebases, which are often millions of lines in length.⁹ In particular, we examine the rise and fall in usage of specific words and phrases in the U.S. Code, quantify the amount of change over time, and present metrics and visualizations of its cross-reference structure. Our work leads to novel and provocative analyses of the U.S. Code's systemic structure, insights into its strengths and weaknesses, and new ways of thinking about the nature of individual laws. For example, we identify the first appearance and spread of important terms like "whistleblower" and "privacy."¹⁰ Also, we visually represent laws' network structures to show how certain laws that introduce substantial reform, including the Patient Protection and Affordable Care Act ("PPACA") and the Dodd-Frank Wall Street Reform and Consumer Protection Act ("Dodd-Frank")¹¹ differ from other long pieces of legislation, such as appropriations bills.¹²

We structure this Article as follows: Part II summarizes past and current codification efforts in the United States.¹³ Part III describes our U.S. Code dataset,

6. See Dru Stevenson, *Costs of Codification*, 2014 U. ILL. L. REV. 1129, 1129 (2014) (identifying downsides to codification, including "legislative borrowing," over-criminalization, unmanageable legal-information costs, and judicial over-emphasis on statutory text rather than policy).

7. See BRADFORD J. WHITE & PAUL W. EDMONDSON, *PROCEDURAL DUE PROCESS IN PLAIN ENGLISH: A GUIDE FOR PRESERVATION COMMISSIONS* (3d ed. 2008) (lamenting that lay persons must rely on lawyers to understand the law); see also *Mathews v. Eldridge*, 424 U.S. 319, 348–49 (1976) (delineating procedural due process requirements, including notice).

8. See LAWRENCE LESSIG, *CODE AND OTHER LAWS OF CYBERSPACE* 6 (1999). Lines of software code are to cyberspace what the laws of physics are to the non-virtual world; they determine what is possible and, in turn, what can be regulated. Dan Orr, *Book Review: Code and Other Laws of Cyberspace*, RES. CENTER FOR CYBERCULTURE STUD. (Aug. 2000), <http://rccs.usfca.edu/bookinfo.asp?BookID=79&ReviewID=79>. In this Article, we conclude that "law is code." See *infra* Part VII. While Lessig likens software code to the laws of nature, this Article analogizes source code to legal code and then proposes legal reforms. See *infra* Part III.

9. See *infra* Part III.

10. See *infra* Part IV.B.4.

11. Patient Protection and Affordable Care Act, Pub. L. No. 111-148, 124 Stat. 119 (2010); Dodd-Frank Wall Street Reform and Consumer Protection Act, Pub. L. No. 111-203, 124 Stat. 1376 (2010).

12. See *infra* Parts III.C.3.b, IV.C, V.

13. See *infra* Part II.

provides an overview of key software engineering principles that we adopt, and outlines the analytics and algorithms that we use.¹⁴ Part IV applies these tools to explore the U.S. Code's *evolution*; for example, we quantify the rise and fall of key terms and the percentage change in selected titles' content.¹⁵ In Part V, we focus on recently passed laws' impact on the U.S. Code.¹⁶ Finally, Part VI applies software engineering metrics to specific titles of the current U.S. Code, focusing on Title 12 (Banks and Banking) and Title 26 (Internal Revenue Service) as examples.¹⁷ In the appendices, we define our network-analysis metrics and visually represent different kinds of laws' network structures.

II. THE UNITED STATES CODE

Given the frequency with which legal practitioners and scholars cite the U.S. Code, many facts about the U.S. Code may seem shocking. For example, Congress did not authorize the official collection of federal statutes until 1926,¹⁸ meaning that as of 2012, around 5 million living U.S. citizens were born before the U.S. Code was first published.¹⁹ Until 1947, the U.S. Code was merely *prima facie* evidence of the statutes reproduced within the U.S. Code;²⁰ only after 1947 did Congress begin the slow, piecemeal process of converting the U.S. Code into controlling law (known as "positive law codification").²¹ Further, the U.S. Code is *still* only *prima facie* evidence of the law for 26 of the U.S. Code's 52 titles.²²

The goal of the U.S. Code is simple enough: to provide "the laws of the United States, general and permanent in their nature."²³ The project of codification, however, has been wrought with difficulty from the beginning.²⁴ This Part outlines the goals that lawmakers have aspired to address with codification and the troubles they have encountered along the way. The techniques proposed in this Article use

14. See *infra* Part III.

15. See *infra* Part IV.

16. See *infra* Part V.

17. See *infra* Part VI.

18. CODE OF LAWS OF THE UNITED STATES OF AMERICA OF A GENERAL AND PERMANENT NATURE IN FORCE DECEMBER 7, 1925; see Will Tress, *Lost Laws: What We Can't Find in the United States Code*, 40 GOLDEN GATE U. L. REV. 129, 136 (2010).

19. See U.S. CENSUS BUREAU, CURRENT POPULATION SURVEY, ANNUAL SOCIAL AND ECONOMIC SUPPLEMENT, 2012 (2013), available at http://www.census.gov/population/age/data/files/2012/2012gender_table1.xlsx.

20. See Tress, *supra* note 18, at 137–38.

21. Act of July 30, 1947, ch. 388, 61 Stat. 633, 633; *Positive Law Codification*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/codification/legislation.shtml> (last visited Jan. 27, 2015); see Tress, *supra* note 18, at 137.

22. See *Positive Law Codification*, *supra* note 21.

23. 1 U.S.C. § 204(a) (2012).

24. See Tress, *supra* note 18, at 133–38.

modern computer scientific methods to analyze and remedy issues that have plagued U.S. lawmakers for centuries.

A. Early Federal Codification Problems

In 1795, Congress authorized the first compilation of federal statutes, which included all public laws and treaties to date.²⁵ But until 1845, the annual session laws were not published on a regular basis; rather, federal statutes were published in newspapers.²⁶ In the early 1820s, individual states began to debate the idea of codification, with the New York Revised Code of 1829 leading the way, followed by newly admitted western states.²⁷ Recognizing the value of codification, private publishers produced chronological, bound volumes of U.S. public laws.²⁸

The first federal solution came when Little, Brown & Co., a Boston-based private publisher, proposed the creation of the Statutes at Large in 1845.²⁹ This collection of laws, as updated, remains the authoritative collection for half the U.S. Code titles today.³⁰ The Statutes at Large contain a chronological set of laws which Congress passed and the President signed into law.³¹ Each volume of the Statutes at Large covers one congressional session.³² The Government Printing Office—created in 1861—replaced Little, Brown & Co. as the entity responsible for publishing the Statutes at Large until 1950, when the Office of the Federal Register in the National Archives took over.³³

While the Statutes at Large improved matters by providing a definitive collection of laws, the chronological, session-based presentation, among other sundry conventions, made it difficult for lawmakers to determine what was current U.S. law in any given subject area.³⁴ In 1848, the chairman of the House Judiciary Committee proposed a bill to revise the session laws.³⁵ The accompanying House Report outlined a litany of issues, including that the session laws may have been “enacted under the pressure of momentary emergency; if not inconsistent, they are

25. Act of Mar. 3, 1795, ch. 50, 1 Stat. 443.

26. See Ralph H. Dwan & Ernest R. Feidler, *The Federal Statutes—Their History & Use*, 22 MINN. L. REV. 1008, 1010 (1938); Tress, *supra* note 18, at 133.

27. See CHARLES COOK, AMERICAN CODIFICATION MOVEMENT 158–59 (1981); EDWIN C. SURRENCY, A HISTORY OF AMERICAN LAW PUBLISHING 90–91 (1990).

28. Richard J. McKinney, *Basic Overview on How Federal Laws Are Published, Organized and Cited*, LAW LIBRARIANS’ SOC’Y WASH. D.C. 2 (Jan. 12, 2006), <http://www.llsdc.org/assets/sourcebook/federal-laws.pdf>.

29. *Id.*

30. See *Positive Law Codification*, *supra* note 21.

31. See McKinney, *supra* note 28, at 2.

32. *Id.* at 3.

33. *Id.*

34. *Id.*

35. See H.R. 535, 30th Cong. (1st Sess. 1848). No record of the bill remains. Tress, *supra* note 18, at 133.

LAW IS CODE

obscure; sometimes involved in statutes dissimilar in title and object, and always scattered over different parts of a broad surface, in the numerous hiding places of which they are concealed.”³⁶ The report admonished, with some prescience, that “enactments defining the duties of a particular office should naturally be so united as to furnish all needful information in one comprehensive body. That which seems to be complete in its enumeration should be so in reality.”³⁷

In 1866, Congress created a commission tasked “to revise, simplify, arrange, and consolidate all statutes of the United States, general and permanent in their nature.”³⁸ Two years into their task, the commission reported several insurmountable difficulties, noting, “[w]here several statutes relating to the same subject modify each other, it has been impossible to state their united effect without writing a new statute.”³⁹ In 1872, the commission presented its proposed revisions, which Congress deemed too extreme a departure from the language of existing laws, and delegated the draft to a special reviser charged with reversing much of the commission’s proposals.⁴⁰ Ultimately, this process yielded the Revised Statutes of the United States, containing 70 titles, which revised, reorganized, and consolidated all permanent and general U.S. laws, and was enacted in 1874 and published in 1875.⁴¹

The Revised Statutes repealed all general acts “embraced in any section” of the revisions, replacing them as controlling authority.⁴² Shortly after publication, however, numerous mistakes and omissions were identified.⁴³ Congress addressed these errors in an amended and updated 1878 revision.⁴⁴ Sensitive to the debacle that these errors and omissions produced, the 1878 Revision provided that it would not “preclude reference to, nor control, in case of discrepancy, the effect of any original act passed by Congress since” December 1, 1873.⁴⁵

Problems arising from the Revised Statutes dealt a blow to the codification movement. The ensuing 50 years saw several proposals to update or replace the Revised Statutes, but Congress did not issue another code until 1926.⁴⁶ In the

36. H.R. REP. NO. 30-671, at 1 (1848).

37. *Id.* at 2.

38. Act of June 27, 1866, ch. 140, 14 Stat. 74.

39. WILLIAM JOHNSTON & CHARLES P. JAMES, REPORT OF THE COMMISSIONERS APPOINTED UNDER ACT OF JUNE 27, 1866, S. Misc. Doc. 101, 40th Cong. (2d Sess. 1868).

40. Dwan & Feidler, *supra* note 26, at 1013.

41. REVISED STATUTES OF THE UNITED STATES PASSED AT THE FIRST SESSION OF THE FORTY-THIRD CONGRESS 1873-74 (2d ed. 1878); *see* McKinney, *supra* note 28, at 3.

42. Sec. 559, 1 Rev. Stat. 1091 (1873); *see* Tress, *supra* note 18, at 135.

43. INACCURACIES AND OMISSIONS IN REVISED STATUTES, H. Exec. Doc. 36, 44th Cong. (1st Sess. 1876).

44. Revised Statutes (1878).

45. *Id.* at iii (preface).

46. CODE OF LAWS OF THE UNITED STATES OF AMERICA OF A GENERAL AND PERMANENT NATURE IN FORCE DECEMBER 7, 1925.

interim, private publishers again shouldered the collection and organization of laws passed since the 1878 Revisions.⁴⁷

B. Early Problems with the U.S. Code

Perhaps still stinging from prior codification efforts, Congress undertook several measures to forestall similar issues. First, Congress enlisted the professional expertise of two private code publishers, West and Edward Thompson, to oversee the new edition of the Code.⁴⁸ Second, Congress was careful to note that the 1926 Revisions were an “official restatement in convenient form” of U.S. law, but “[n]o new law is enacted and no law repealed. It is *prima facie* the law. It is presumed to be the law. The presumption is rebuttable by production of prior unrevoked Acts of Congress at variance with the Code.”⁴⁹

Third, this overly cautious, mostly redundant preface was the product of legislative compromise. The original bill, as passed by the House,⁵⁰ provided that the U.S. Code would remain *prima facie* evidence until June 30, 1927, at which time it would become controlling law.⁵¹ Lawmakers hoped that window would allow sufficient time to correct any new errors.⁵² Fearing the prospect of errors, however, the Senate amended the bill to prevent the U.S. Code from becoming the controlling statement of the law.⁵³ True enough, 537 errors were later found and corrected, 88 of which were substantive errors.⁵⁴

Identifying those errors also presented difficulty. The aforementioned preface cautiously limited the U.S. Code to *prima facie* evidence of U.S. law, but it failed to identify which published laws *could be* cited to rebut the presumption.⁵⁵ Ultimately, the 1878 Revision controlled for statutes enacted before December 1, 1873, and although the 1878 Revision also contained statements of the law from 1874 to 1878, the Statutes at Large were the authoritative text for all statutes from 1873 to date.⁵⁶

47. SURRENCY, *supra* note 27, at 107–10; Dwan & Feidler, *supra* note 26, at 1016–21.

48. Tress, *supra* note 18, at 136.

49. Preface, U.S.C. (1926).

50. H.R. 10000, 69th Cong. (1926).

51. See Richard J. McKinney, *Unraveling the Mysteries of the U.S. Code*, LAW LIBRARIANS' SOC'Y WASH. D.C. 1 (Aug. 2009), <http://www.llsdc.org/assets/sourcebook/usc-mysteries.pdf>.

52. *Id.*

53. See Act of June 30, 1926, ch. 712, 44 Stat. 777.

54. See McKinney, *supra* note 51, at 1.

55. Tress, *supra* note 18, at 137. Private publishers like West and Lexis filled the gap by providing annual updates, and today Congress annually archives electronic versions. *Id.* at 137 n.42.

56. *Id.* at 137.

C. The U.S. Code: 1926 to Today

After its first publication in 1926, the U.S. Code was replaced by a new edition in 1934, followed by new editions every six years.⁵⁷ The U.S. Code remained only *prima facie* evidence of the law until 1947, when Congress began the process of converting the U.S. Code to the controlling statement of the law.⁵⁸ That year, U.S. Code Title 1 (General Provisions) was positively codified, along with Title 4 (Flag & Seal, Seat of Government, and the States), Title 6 (Official & Penal Bonds), Title 9 (Arbitration), and Title 17 (Copyrights).⁵⁹ Interestingly, one congressman noted the intent to begin with “the more important titles and those urgently needing codification,” including, for example, Title 28 on the Judiciary.⁶⁰ Despite that lofty initial goal, the first few positively codified titles were “low-hanging fruit” that required little editing to prepare⁶¹—a volte-face that was likely motivated by Congress’s prior track record with positive law codification.

The original version of the U.S. Code organized then-existing federal laws into 50 titles within a single bound volume; today, the U.S. Code contains over 47,000 pages, 51 titles, and spans several volumes.⁶² In 1974, Congress created the Office of the Law Revision Counsel of the U.S. House of Representatives (“OLRC”) to prepare and publish the U.S. Code.⁶³ Among other things, the OLRC (1) periodically reviews enacted laws and makes recommendations for repealing obsolete, superfluous, and superseded provisions; (2) determines whether and how new laws should be incorporated into the Code; (3) classifies the newly enacted provisions so that they may be incorporated into the relevant titles of the U.S. Code; (4) makes the necessary revisions to each title within the U.S. Code; and (5) recommends certain titles for positive law codification.⁶⁴

While the OLRC’s task of incorporating new law into the U.S. Code can be simple when the laws are small and narrow in subject-matter (though not necessarily so), the task is more complicated when the laws are large, cover a multitude of subjects, and/or contain a complicated mixture of amendatory and freestanding provisions, general specific provisions, and permanent and temporary

57. *Id.*; see 1 U.S.C. § 202(c) (2012).

58. Tress, *supra* note 18, at 137; see Act of July 30, 1947, ch. 388, 61 Stat. 633, 638.

59. Tress, *supra* note 18, at 137–38 (citing William Chamberlain, *Enactment of Parts of the United States Code into Positive Law*, 36 GEO. L.J. 217 (1947)).

60. 93 CONG. REC. 8384 (1947) (remarks of Rep. John M. Robison).

61. Tress, *supra* note 18, at 138.

62. Peter LeFevre, *Positive Law Codification Will Modernize U.S. Code*, THE HILL (Sept. 28, 2010, 5:33 PM), <http://thehill.com/blogs/congress-blog/judicial/121375-positive-law-codification-will-modernise-us-code>.

63. Act of Dec. 27, 1974, ch. 3, Pub. L. No. 93-554, 88 Stat. 1771, 1777 (codified as amended at 2 U.S.C. § 285 (2012)).

64. 2 U.S.C. § 285b (2012); *About the Office; Contact Information*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, http://uscode.house.gov/about_office.xhtml (last visited Jan. 22, 2015).

provisions.⁶⁵ The OLRC often must make impressionistic determinations about how to incorporate a new piece of legislation into the U.S. Code.⁶⁶ Moreover, while the OLRC can editorially add new sections, chapters, and statutory notes to 23 non-positive law titles, only Congress can add new sections and chapters to the 27 positive law titles, and only by amendment.⁶⁷

D. Criticisms and Aspirations for the U.S. Code

Commentators cite numerous problems with the U.S. Code, including: (1) many of the new laws passed since 1926 are often shoehorned awkwardly into pre-existing titles; (2) Congress often pays little or no attention to existing laws when enacting new legislation, which makes it difficult for U.S. Code editors to keep statutes that relate to similar subjects together; and (3) the increasingly voluminous body of legislation since 1926 has produced many obscure, obsolete, and redundant provisions, archaic and inconsistent language, and statutory errors.⁶⁸

In touting the benefits of positive law codification, the OLRC has identified and, in some cases, reaffirmed the U.S. Code's deficiencies. For example:

Improved organization. Provisions that are closely related by subject may be scattered in different places in the . . . Code. Such provisions may have been enacted many years apart and incorporated . . . at different times. Positive law codification affords an opportunity to revisit the organizational structure of statutory material. Thoughtful regrouping of provisions often yields a statutory product that is easier to use and that fosters a more comprehensive understanding of the law.

Elimination of obsolete provisions. Obsolete provisions are frequently identified in the course of preparing a positive law codification bill . . . [and] are eliminated from the law after appropriate vetting of proposed changes. Although such changes seem small and innocuous when viewed individually, the cumulative effect of removing all obsolete provisions can be profound, resulting in a much more compact and comprehensible text.

Precise statutory text. The process of positive law codification promotes public access to the precise text of Federal statutory law. Provisions set out

65. *About Classification of Laws to the United States Code*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, http://uscode.house.gov/about_classification.xhtml (last visited Feb. 2, 2015).

66. *Id.*

67. *Id.*

68. See LeFevre, *supra* note 62; see also *Positive Law Codification*, *supra* note 21 (noting that revisers seek to reorganize existing provisions, conform style and terminology, modernize obsolete language, and correct drafting errors); *About Classification of Laws to the United States Code*, *supra* note 65.

LAW IS CODE

in non-positive law titles . . . may vary slightly from the precise language enacted into law; cross references are adapted and stylistic changes are made in order to facilitate the integration of Federal Statutory provisions. . . .

***Cleaner amendments.** Positive law codification promotes accuracy and efficiency in the preparation of amendments. . . . [s]pecifying words to be struck or the place where new words are to be inserted or simplified[; u]nderstanding the impact of proposed amendments is easier[; d]rafting errors are reduced. In addition, compliance with congressional rules requiring comparative prints (showing proposed omissions and insertions) is facilitated.⁶⁹*

With the foregoing in mind, this Article's remaining parts describe the development and application of computer scientific techniques to assess and remedy problems that have plagued, and often continue to plague, the U.S. Code.

III. SOFTWARE ENGINEERING APPROACHES TO ANALYZING THE LAW

A. Analogizing Legal Code to Software Code

Many analogies between software code and legal code apply at both general and specific levels. At a general level, both forms of code consist of a collection of rules that govern certain operations: human transactions in the case of legal code, and computer transactions in the case of computer code. The main difference—that humans interpret and implement laws whereas machines interpret and implement software—is more a matter of degree than kind. Because humans are more flexible and intelligent, laws need not be as explicit and precise as software. This lack of precision, however, is not without cost, as evidenced by the fundamental debate over “rules versus standards.”⁷⁰ At a functional level, software and legal code share common features, functions, and frailties, irrespective of whether they are meant for or interpreted by humans or machines; hence, methods that have been developed in one domain should be relevant in the other. For example, based on concerns raised about the understandability of the law, we adopt four approaches

69. See *Positive Law Codification in the United States Code*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE 5, http://uscode.house.gov/codification/positive_law_codification.pdf (last visited Jan. 27, 2015).

70. See Louis Kaplow, *Rules Versus Standards: An Economic Analysis*, 42 DUKE L.J. 557, 559–63 (1992).

from good software design practices—conciseness,⁷¹ change,⁷² coupling,⁷³ and complexity⁷⁴—that should also have implications for good legal coding practices.

Software engineers apply a range of techniques to analyze computer code that may be relevant to analyzing the U.S. Code. First, conceptually, the similarity between software code and the U.S. Code in terms of function is an important parallel that exists between computer programming and lawmaking. For example, software code often is written to compute some kind of output upon receiving certain inputs, *e.g.*, a computation module receives numerical values to perform arithmetic, and a search engine—like Google—receives a search query and returns a list of results. Similarly, the U.S. Code is a collection of laws that describes the inputs that determine when the authority of the federal government is to be applied and the outcomes that result; *e.g.*, how the salaries of members of Congress are determined (contained in Title 2) and the role of the U.S. Patent and Trademark Office (contained in Title 35).⁷⁵

Second, the internal structure and composition of both software code and legal code also matter. Laws should be easy to read and comprehend so that individual citizens can understand their rights and obligations, and lawyers, legislators, judges, and jurors can more efficiently perform their jobs. Consonant with this concern, the U.S. Senate’s Legislative Drafting Manual emphasizes readability.⁷⁶ Section 107, entitled “Focus on Reader,” states: “A draft must be understandable to the reader. The rules in this manual should be applied in a manner that makes the draft clearer and easier to understand.”⁷⁷ Similarly, the manual for the U.S. House of Representatives states, “[d]raft should be clear and understandable – In almost all cases, the message has a better chance of accomplishing your client’s goal if it is

71. Harry H. Porter III, *Designing Programming Languages for Reliability* 2 (Oct. 16, 2001) (unpublished paper) (on file with the Journal of Business & Technology Law), *available at* <http://web.cecs.pdx.edu/~harry/musings/RelLang.pdf> (“Most programming languages tend to emphasize conciseness.”).

72. *The Importance of Writing Good Code*, GNOME DEVELOPER, <https://developer.gnome.org/programming-guidelines/stable/writing-good-code.html.en> (last visited Jan. 21, 2015) (“General-purpose code is easier to reuse and modify than very specific code with lots of hardcoded assumptions.”).

73. Kailash Patidar et al., *Coupling and Cohesion Measures in Object Oriented Programming*, 3 INT’L J. ADVANCED RES. IN COMPUTER SCI. & SOFTWARE ENG’G 517, 517 (2013) (“[C]oupling is an important aspect in the evaluation of reusability and maintainability of components or services.”).

74. NEIL D. JONES, *COMPUTABILITY & COMPLEXITY: FROM A PROGRAMMING PERSPECTIVE*, at vii (1997).

75. 2 U.S.C. § 31 (2012); 35 U.S.C. § 2 (2012) (delineating the U.S. Patent and Trademark Office’s powers and duties).

76. OFFICE OF THE LEGISLATIVE COUNSEL, U.S. SENATE, *LEGISLATIVE DRAFTING MANUAL* (1997), *available at* [http://www.law.yale.edu/documents/pdf/Faculty/SenateOfficeoftheLegislativeCounsel_LegislativeDraftingManual\(1997\).pdf](http://www.law.yale.edu/documents/pdf/Faculty/SenateOfficeoftheLegislativeCounsel_LegislativeDraftingManual(1997).pdf) [hereinafter *SENATE DRAFTING MANUAL*].

77. *Id.* at 7.

readable and understandable. It should be written in English for real people.”⁷⁸ Unnecessarily complicated laws can interfere with commerce, economic growth, and access to justice.⁷⁹

Because the form—not just the function—of the legal code is important, a software engineering approach can yield several new insights when applied to the law. In particular, software developers are deeply invested in making their code readable and easy to understand. Software engineering teams often need to integrate new team members, fix bugs, and refactor existing code, which are all tasks that require a deep understanding of code written by others who are often no longer available to provide support or clarification.⁸⁰ These requirements suggest that the tools used by software engineers to track progress, monitor potential vulnerabilities, or simply gain an understanding of an existing software codebase may be useful for serving the same functions when applied to legal code.

B. U.S. Code Datasets for Analysis

We use two datasets for our analyses:

1. We obtained complete text versions of the U.S. Code from 1926 to 2006 under license from William S. Hein & Co.⁸¹ This dataset includes the editions of the U.S. Code from 1926, 1934, 1940, 1946, 1952, 1958, 1964, 1970, 1976, 1982, 1988, 1994, 2000, and 2006. The text in these U.S. Code editions is split up only per title, meaning that most of our analyses and visualizations are done on a title level. Collectively, we refer to the U.S. Code editions from Hein as our “historical dataset.”

78. THE OFFICE OF THE LEGISLATIVE COUNSEL, U.S. HOUSE OF REPRESENTATIVES, HOUSE LEGISLATIVE COUNSEL’S MANUAL ON DRAFTING STYLE 5 (1995), available at <http://legcounsel.house.gov/pdf/draftstyle.pdf> [hereinafter HOUSE DRAFTING MANUAL].

79. See *Over-Regulated America*, ECONOMIST, Feb. 18, 2012, available at <http://www.economist.com/node/21547789> (contending that the Dodd-Frank Act’s complicated provisions constrict economic growth); Michael Burgess, *Death Is Much Less Complicated Than the U.S. Tax Code*, FORBES (Apr. 15, 2013, 8:00 AM), <http://www.forbes.com/sites/realspin/2013/04/15/death-is-much-less-complicated-than-the-u-s-tax-code/> (arguing that the federal tax code confuses taxpayers and unfairly advantages certain groups).

80. ROBERT SEDGEWICK & KEVIN WAYNE, INTRODUCTION TO PROGRAMMING IN JAVA 8 (2008) (“[Bugs] are the bane of a programmer’s existence: the error messages can be confusing or misleading, and the source of the error can be very hard to find.”).

81. See *U.S. Code*, HEINONLINE, <http://heinonline.org/HOL/Index?collection=uscode> (last visited Feb. 2, 2015).

2. The official current U.S. Code is available for free download from the Office of the Law Revision Counsel (“OLRC”).⁸² This version is in Extensible Markup Language (“XML”) format, which means that headings, sub-headings, and cross-references are annotated within the document.⁸³ As a result, in addition to analyzing the text of the U.S. Code on a per-title basis, we can also use algorithmic approaches on a per-section basis. We also used data from the Cornell Legal Information Institute (“LII”), which is similarly structured, for data mining the cross-references from the U.S. Code.⁸⁴ We refer to the current U.S. Code edition from the OLRC as our “current dataset.”

C. Choosing Software Engineering Approaches and Metrics

The software engineering industry uses a wide range of frameworks, principles, and metrics in its work.⁸⁵ Some materials focus on “design patterns,” which describe solutions to common programming tasks;⁸⁶ others emphasize project-management techniques to monitor a software engineering project’s progress;⁸⁷ and still others educate coder-readers with examples of poorly written code.⁸⁸ There are entire treatises on subtypes of software engineering, such as refactoring, in which a codebase is re-organized so that it is cleaner and easier to understand.⁸⁹ For the purposes of this Article, we focus on four categories of issues that affect the understandability of the legal code: conciseness, change, coupling, and complexity.

82. *Current Release Point*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, http://uscode.house.gov/download/releasepoints/us/pl/113/290not235not287/xml_uscAll@113-290not235not287.zip (last visited Jan. 21, 2015).

83. *United States Legislative Markup: User Guide for the USLM Schema*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/download/resources/USLM-User-Guide.pdf> (last visited Feb. 2, 2015) [hereinafter *USLM User Guide*].

84. *U.S. Code: Table of Contents*, CORNELL LEGAL INFO. INST., <http://www.law.cornell.edu/uscode/text> (last visited Feb. 22, 2015).

85. See generally STEVE MCCONNELL, *CODE COMPLETE: A PRACTICAL HANDBOOK FOR SOFTWARE CONSTRUCTION* (2d ed. 2004).

86. See, e.g., ERICH GAMMA ET AL., *DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE* 2–4 (1995) (defining a design pattern as a solution to a common problem that one can use “a million times over, without ever using it the same way twice”).

87. See, e.g., BARBARA KITCHENHAM, *SOFTWARE METRICS: MEASUREMENT FOR SOFTWARE PROCESS IMPROVEMENT* 5 (1996).

88. See, e.g., ROBERT C. MARTIN, *CLEAN CODE: A HANDBOOK OF AGILE SOFTWARE CRAFTSMANSHIP* 285–314 (2009) (listing common coding problems).

89. See, e.g., MARTIN FOWLER, *REFACTORED: IMPROVING THE DESIGN OF EXISTING CODE*, at xvii (1999) (explaining how to refactor without “introduc[ing] bugs into the code”).

1. Conciseness

According to the U.S. Senate Legislative Drafting Manual, brevity is desirable: “Use short, simple sentences rather than complex or compound sentences. If a shorter term is as good as a longer term, use the shorter term.”⁹⁰ Laws that are long and verbose require more time to read, interpret, and revise. Despite being a simple and limited metric, length is a reasonable starting point for quantifying legal code.

In software engineering, the size of a software codebase, usually measured by lines of code (“LOC”), is a common metric for evaluating the effort required to develop and maintain it.⁹¹ Each line of code has the potential to contain errors or unnecessary complexity. Large amounts of code, therefore, correspond to larger, more complicated software, which might have a greater number of bugs. In practice, while imperfect, counting lines of code is a simple, reasonable starting point to start characterizing a codebase’s complexity and potential problems.⁹²

In software code, the number of LOCs is typically used as the rough approximation of complexity; since most programming languages generally require line breaks, this provides a rough indication of the number of “instructions” in the program.⁹³ Turning to the U.S. Code, to measure conciseness, we use the number of words as our measurement unit because each clause or sentence is not necessarily a new line in the document.

We count words in two ways for different datasets. First, for our historical dataset, we visualize the length of different titles of the U.S. Code at different snapshots, namely every six years when a new complete edition of the U.S. Code is released. Second, for the most current edition of the U.S. Code, we compare the lengths of different bills and titles.

2. Change

Revisions to the law require interested parties to understand what has changed and what has remained the same. Changes may also introduce unexpected or unintended effects. Over the span of many decades, the U.S. Code has become more difficult to read and understand due to the changes made by many Congresses.⁹⁴

90. SENATE DRAFTING MANUAL, *supra* note 76, at 4.

91. MCCONNELL, *supra* note 85, at 725–26.

92. See Jarrett Rosenberg, *Some Misconceptions About Lines of Code*, in PROCEEDINGS FOURTH INTERNATIONAL SOFTWARE METRICS SYMPOSIUM 137 (1997), available at <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=637174>.

93. Graylin Jay et al., *Cyclomatic Complexity and Lines of Code: Empirical Evidence of a Stable Linear Relationship*, 2 J. SOFTWARE ENG'G & APPLICATIONS 137, 137 (2009), available at <http://www.scirp.org/Journal/PaperDownload.aspx?paperID=779> (finding a “practically perfect linear relationship” between lines of code and cyclomatic complexity).

94. See Daniel Martin Katz & Michael James Bommarito II, *Measuring the Complexity of the Law: The United States Code 5* (Aug. 1, 2013) (unpublished working paper) (on file with the Journal of Business &

Quantifying what has actually changed with each new edition could be useful for understanding how the law has evolved and as a first step for understanding how the law might be designed better. For example, measuring changes reveals what sections have been modified by Congress frequently and what sections have withstood the test of time.

For large software codebases, because many programmers are working on the code at the same time, mechanisms that ensure that changes do not break functional code or create conflicts are needed. The practice of software engineering has adopted “version control systems” to handle these problems.⁹⁵ Given that the U.S. Code is the product of many individual members of Congress over time, similar mechanisms are needed. Thus, a software engineering-inspired version control approach to the law could be a reasonable future method of managing legislative changes.

For this metric, we focus on the goal of developing software engineering-inspired tools for visualizing and communicating changes to the Code. We quantify two types of changes: (1) the aggregate number of words added or deleted, and (2) the appearance and spread of words over time and to different titles of the U.S. Code.

a. Addition-and-deletion metrics

When working on a codebase, software engineers routinely add, delete, revise, reorder, or restructure LOCs. For the purpose of analyzing the existing U.S. Code, a key insight from software engineers is how they communicate and visualize changes to a document. Specifically, although editing software code can involve many high-level thought processes, they can be communicated through two operations: the addition and deletion of LOCs. In software, such a comparison of two versions of the same document is called a “diff” operation.⁹⁶ Revising an existing line of code is simply the deletion of the existing line and the addition of a new line. When a team member makes changes, the rest of the team can easily identify where changes were made to a document and what those changes were, essentially by viewing redline comparisons. Further, the number of lines changed may suggest whether the

Technology Law), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2307352 (The U.S. Code “contains hundreds of thousands of provisions and tens of millions of words”).

95. Christopher Menegay, *Using Source Code Control in Team Foundation*, MICROSOFT DEVELOPER NETWORK (Sept. 2005), <http://msdn.microsoft.com/en-us/library/ms364074%28v=vs.80%29.aspx> (noting that version control systems can “manage files through the development lifecycle, keeping track of which changes were made, who made them, when they were made, and why”). Popular version control systems include: Concurrent Versions System (“CVS”), Apache Subversion (“SVN”), Mercurial, and Git. See *Concurrent Versions System*, NONGNU.ORG, <http://www.nongnu.org/cvs/> (last visited Feb. 10, 2015); *Apache Subversion*, APACHE SOFTWARE FOUND., <https://subversion.apache.org/> (last visited Feb. 10, 2015); *MERCURIAL SCM*, <http://mercurial.selenic.com/> (last visited Feb. 10, 2015); *GIT-SCM.COM*, <http://git-scm.com/> (last visited Feb. 10, 2015).

96. *GNU Tools*, UNIXHELP FOR USERS (Sept. 22, 1993), <http://unixhelp.ed.ac.uk/CGI/man-cgi?diff>.

revision was large or small. The act of comparing two versions of a document is a fundamental operation that programmers use regularly.⁹⁷

Each historical edition of the U.S. Code is a snapshot of the laws at one instance in time. Using text-matching techniques, we can also apply the “diff” concept to two versions of a legal document; the only difference is that, instead of computer instructions, the law is written in English. Text matching is simply the task of detecting whether a sequence of words in one version of a document exists in a previous version. When applied to an entire document, it is possible to calculate what percentage of a document is new and what percentage previously existed.

b. Word-based metrics

Software engineers also use a number of other text-based tools in their daily work routines. Similar to how Internet search engines help users find relevant documents online or “find files” programs help computer users locate documents on their own computer, software engineers might search for specific terms in a codebase or their frequency to help them do their work.⁹⁸ Looking for the existence of terms throughout a codebase might, for example, help the software engineer determine whether a feature has already been implemented or assess the design conventions that the team has used. In principle, these search techniques could also be applied to snapshots of the codebase over time to identify changes.

Understanding changes in the law, of course, requires going beyond simple length measurements. It is interesting, for example, to detect the first appearance of particular words in the U.S. Code; given our historical dataset, doing so is quite straightforward. In addition, we count the number of times that each word appears in each edition of the U.S. Code. Similar efforts have been employed by Google to count the appearance of terms in all English literature,⁹⁹ the *New York Times* for its news coverage,¹⁰⁰ and by other researchers for U.S. Supreme Court opinions.¹⁰¹ In our case, these measurements are useful because they reflect the extent to which the U.S. Code covers different concepts.

97. See *id.*

98. See *How to: Programmatically Search for and Replace Text in Documents*, MICROSOFT DEVELOPER NETWORK, <http://msdn.microsoft.com/en-us/library/f1f367bx.aspx> (last visited Jan. 31, 2015) (explaining how to use Microsoft Word’s “find” function).

99. See Ben Zimmer, *Google’s Ngram Viewer Goes Wild*, ATLANTIC (Oct. 17, 2013, 9:17 AM), <http://www.theatlantic.com/technology/archive/2013/10/googles-ngram-viewer-goes-wild/280601/>.

100. Alexis Lloyd, *Chronicle: Tracking New York Times Language Usage over Time*, N.Y. TIMES (July 23, 2014), <http://blog.nytlabs.com/2014/07/23/chronicle-tracking-new-york-times-language-use-over-time/>.

101. Daniel Martin Katz et al., *Legal N-Grams?: A Simple Approach to Track the “Evolution” of Legal Language* (Dec. 13, 2011) (unpublished paper) (on file with the Journal of Business & Technology Law), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1971953.

3. Coupling

The U.S. Code is not simply a long passage of text; the legislative drafting manuals of both houses of Congress state that individual sections of legislation should be organized into titles, sections, sub-sections, sub-clauses, and other subdivisions.¹⁰² Moreover, these subdivisions often reference each other.¹⁰³ For instance, one part of the U.S. Code might refer to definitions in another part, creating dependencies between them. The coupling of various parts of the U.S. Code creates nonlinearities that can make the code more challenging to parse and revise. In particular, a reader must now explore different “pathways” of references to fully understand a certain domain of law. Furthermore, revisions to any part of a chain of references could contribute to unknown, unintended downstream effects. Mapping the large-scale structure of cross-references in the U.S. Code, therefore, may reveal potential vulnerabilities in the law.

a. Modularity in software

In the software context, good software systems are easy to separate into different modules, with the interface between modules being kept relatively sparse and simple.¹⁰⁴ The notion of “modularity” is the central idea behind “object-oriented programming,” which is a fundamental design pattern in programming large software systems today and the focus of many seminal computer science papers and textbooks.¹⁰⁵

Object-oriented programming has become a dominant paradigm in software because it leverages the power of abstraction and modularity.¹⁰⁶ For example, a powerful word processor application like Microsoft Word has many functions, including formatting, citation management, checking spelling and grammar, and document printing options.¹⁰⁷ To manage this complexity, large software systems are split into modular subsystems. Smaller, more agile teams of software engineers

102. See HOUSE DRAFTING MANUAL, *supra* note 78, at 23–24; SENATE DRAFTING MANUAL, *supra* note 76, at 8–10.

103. See Katz & Bommarito, *supra* note 94 (citing 11 U.S.C. § 101 as an example of a statute that contains both “within-Title” references and “cross-Title” references).

104. MCCONNELL, *supra* note 85, at 38.

105. Ola Berge et al., Learning Object-Oriented Programming (Nov. 23, 2007) (unpublished paper) (on file with the Journal of Business & Technology Law), available at https://telearn.archives-ouvertes.fr/file/index/docid/190184/filename/Berge_2003.pdf.

106. See, e.g., Leslie Kaelbling et al., *Introduction to Electrical Engineering and Computer Science I: Syllabus*, MIT OPENCOURSEWARE, <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/Syllabus/> (last visited Jan. 21, 2015) (setting a goal to teach “the fundamental design principles of modularity and abstraction in a variety of contexts from electrical engineering and computer science”).

107. See *Word Object Model Overview*, MICROSOFT DEVELOPER NETWORK, <http://msdn.microsoft.com/en-us/library/kw65a0we.aspx> (last visited Feb. 10, 2015).

are responsible for each of these modules, and each team only needs to understand the input-output behavior of other modules with which its module interacts.¹⁰⁸ Object-oriented design, therefore, leads to clearer lines of responsibility, both from a software standpoint and a human team management standpoint. This modularity results in more efficient coding, debugging, and, ultimately, more robust software.

One way to study modularity is to interpret the system as a network (also known as a graph), where each function or variable corresponds to a node in the network, and there is an “edge” (a connection) from component A to component B if component B references component A.¹⁰⁹ A rich body of algorithms and techniques has been developed to characterize the properties of these networks.¹¹⁰ Continuing with the Microsoft Word example, when a user decides to print a document, the “user interface” module connects to the “print” module.¹¹¹ Any major software system involves multiple references among its component modules; good object-oriented design suggests that cross-references should be used only when they are necessary to avoid needless dependencies and complexity.

For any given software codebase, it is possible to construct and analyze its nodes and edges in aggregate. The resulting network structure can provide insights into the nature of the software system, such as how robust it is and where its vulnerabilities likely reside. The network map can also provide a sense of the different categories of modules that exist in a software system. Previous work, for example, has examined the core-periphery architecture common to many large software systems.¹¹² The portion of the network to which a certain module belongs can provide information about how the module relates to the rest of the system.¹¹³

b. Modularity in the U.S. Code

The same modularity principles can be applied to the law. We can interpret each section of the U.S. Code as a node of the network, with citations to sections as the network’s edges. We can then analyze the graph structure for novel insights into the structure of the U.S. Code.

As a concrete example, 37 U.S.C. § 329, which describes an incentive bonus for retired or former members of the military, cites exactly two other sections, 37

108. MCCONNELL, *supra* note 85, at 21–22.

109. UDAY P. KHEDKER ET AL., DATA FLOW ANALYSIS 234 (2009).

110. See, e.g., THOMAS H. CORMEN ET AL., INTRODUCTION TO ALGORITHMS 587 (3d ed. 2009) (describing how computer scientists can use algorithms and graphing techniques to solve computational problems).

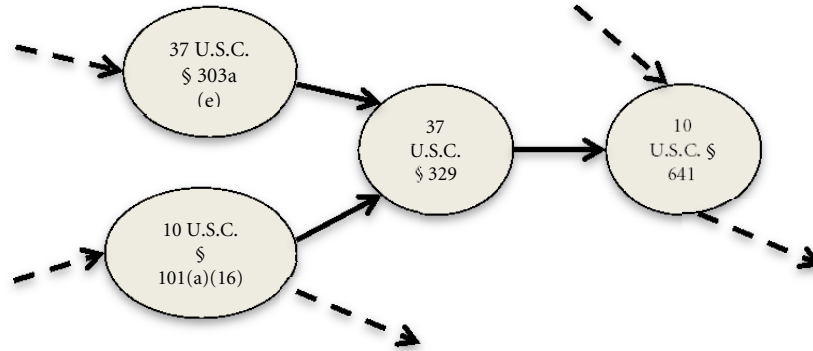
111. See Beth Melton, *Organizing Your Macros*, MICROSOFT WORD MVP (Nov. 1, 2002, 9:52 PM), <http://word.mvps.org/faqs/macrosvba/OrganizeMacros.pdf>.

112. Alan MacCormack et al., *The Architecture of Complex Systems: Do Core-Periphery Structures Dominate?* 1 (Jan. 19, 2010) (unpublished paper) (on file with author), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1539115.

113. *Id.* at 7.

U.S.C. § 303a(e) (general provisions of special pay in the military), and 10 U.S.C. § 101(a)(16) (a definition of “congressional defense committees”).¹¹⁴ Meanwhile, 37 U.S.C. § 329 is cited by one other section, 10 U.S.C. § 641, which notes that other laws in Title 10 of the U.S. Code do not apply to the officers to whom the bonus in 37 U.S.C. § 329 applies.¹¹⁵ Locally, the network is shown in Figure 1 (with dashed arrows representing links to and from other parts of the U.S. Code).

Figure 1: Network representation of references to and from 37 U.S.C. § 329



This simple representation immediately shows a chain of citations in which modifying 37 U.S.C. § 303a(e) could have ramifications for 10 U.S.C. § 641.¹¹⁶ Now, imagine a longer chain with multiple branches, some of which could refer back to the section being modified. These chains can be used to identify complex sequences of legal implications that even the most knowledgeable and intelligent human cannot fully comprehend without technological assistance.

The entire U.S. Code comprises a large network with many references. This network can be analyzed in many ways; previous work, for instance, has sought to identify important U.S. Code sections by following references and determining which sections are encountered most often.¹¹⁷ In our work, we examine the U.S. Code network in the following three ways.

First, for the historical dataset, we examine how sections from bills passed by Congress map to sections in the U.S. Code.¹¹⁸ This data is available from the OLRC

114. See 37 U.S.C. §§ 303(a), 329 (2012); 10 U.S.C. § 101(a) (2012).

115. See 37 U.S.C. § 329 (2012); 10 U.S.C. § 641 (2012).

116. See *supra* Figure 1.

117. See, e.g., Katz & Bommarito, *supra* note 94, at 1, 6.

118. *About the Table III Tool*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/table3/table3explanation.htm> (last visited Jan. 22, 2015). The Office of the Law Revision Counsel provides tables that “show where recently enacted laws will appear in the United States Code and which sections of the Code have been amended by those laws.” See *United States Code Classification Tables*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/classification/tables.shtml> (last visited Jan. 22, 2015).

for every bill ever passed by Congress (including, interestingly, public laws before the U.S. Code came into being in 1926).¹¹⁹ Specifically, for selected recent legislation, we find previously enacted laws that have the most overlapping number affected U.S. Code sections. This method allows us to find groups of similar laws by domain. For instance, the laws most similar to the Dodd-Frank Wall Street Reform and Consumer Protection Act tend to be finance and banking laws.¹²⁰

Second, for the current law, we apply concepts from recent work on the network architecture of software codebases to describe the structure of U.S. Code titles and selected bills passed by Congress.¹²¹ This analysis is based on finding the network's "core," which is the largest interconnected collection of nodes in the network.¹²² More precisely, we define the core as the largest "strongly connected" component of the network.¹²³

Third, we identify important sections by using the structure of the network of cross-references in the current U.S. Code.¹²⁴ Specifically, we use a link analysis algorithm very similar to PageRank, popularized by Google as their method of ranking the importance of individual webpages.¹²⁵ The idea of PageRank is that each section in the U.S. Code has references to and from other sections, and a section that has many references to it is likely more important.¹²⁶ Further, if an important section refers to other sections, those sections may also be important. Using this intuition, the relative importance of all sections in the U.S. Code can be calculated. Previous work has applied this approach to academic literature¹²⁷ and, in the domain of law, the social network of the U.S. law professoriate.¹²⁸

119. *Table III Tool*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/table3/table3years.htm> (last visited Jan. 22, 2015).

120. See *infra* Part IV.C and notes 168–69.

121. See, e.g., Carliss Baldwin et al., *Hidden Structure: Using Network Methods to Map System Architecture* (Harv. Bus. Sch. Working Paper, No. 13-093, May 2013), available at <http://dash.harvard.edu/handle/1/1064642> 2 (describing an operational methodology to characterize complex technical system architecture); Alan Grosskurth & Michael W. Godfrey, *Architecture and Evolution of the Modern Web Browser 1–2, 5, 18* (June 20, 2006) (unpublished paper) (on file with author), available at <http://plg.uwaterloo.ca/~migod/papers/2006/jss-browserRefArch.pdf> (presenting a reference architecture for web browsers).

122. See Baldwin et al., *supra* note 121, at 2, 8.

123. See *infra* App. A (defining mathematical terms that appear in this Article's network analysis, including "core" and "strong connectedness").

124. See *USLM User Guide*, *supra* note 83; *U.S. Code*, *supra* note 84.

125. See Sergey Brin & Lawrence Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, 30 J. COMPUTER NETWORKS & ISDN SYS. 107, 109–10 (1998).

126. See *id.* at 109–10, 117; see also LAWRENCE PAGE ET AL., *THE PAGERANK CITATION RANKING: BRINGING ORDER TO THE WEB* (1998), available at <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.1768>.

127. Carl T. Bergstrom et al., *The Eigenfactor™ Metrics*, 28 J. NEUROSCIENCE 11433 (2008).

128. Daniel Martin Katz et al., *Reproduction of Hierarchy? A Social Network Analysis of the American Law Professoriate*, 61 J. LEGAL EDUC. 76 (2011).

4. Complexity

The law is riddled with conditional statements, exceptions, and special cases.¹²⁹ Applying different rules to different situations is not inherently bad; however, such “balancing tests” make it more challenging to fully appreciate the consequences of a given piece of legislation.¹³⁰ Further, an excessive number of conditional statements might suggest that the underlying rule is faulty, requiring many special cases and exceptions. For these reasons, methods to count the number of statements that exist in the law might be useful for analyzing the U.S. Code.

Analogously, software code often contains conditional statements of the following form:

```
IF (condition)
    (execute subroutine A)
ELSE
    (execute subroutine B)
```

If a condition is met, then some subroutine A is executed, and if the condition is not met, some other subroutine B is executed.¹³¹ Each time a conditional statement appears, the possible execution of the software forks into two paths. Further, conditional statements can be nested (there can be conditional statements inside subroutines), which can lead into exponentially many possible execution paths for a given input.¹³² The complexity that conditional statements introduce is formalized in software engineering as “cyclomatic complexity” (sometimes known as “McCabe’s complexity”), which is the number of times a piece of code has to make a decision, *i.e.*, the number of paths in software.¹³³ It can be computed by assigning a score to each conditional statement that a piece of software encounters.¹³⁴

To create an analogous metric for the U.S. Code, we count the number of conditional terms in a passage of text. We count the occurrences of the following conditional terms in a law or a section of the U.S. Code: “if,” “except,” “but,”

129. See, e.g., RESTATEMENT (SECOND) OF TORTS §§ 314, 314A (1965) (providing that a person has no duty to act when another person requires the first person’s aid or protection, unless a special relationship exists between them); U.C.C. § 2-207 (2002) (prescribing that a “definite and seasonable expression of acceptance” forms a contract even if the offeree’s terms differ from the offeror’s, unless the offeree expressly conditions acceptance on the offeror’s assent to the different terms).

130. Patrick M. McFadden, *The Balancing Test*, 29 B.C. L. REV. 585, 636–42 (1988), available at <http://lawdig.italcommons.bc.edu/bclr/vol29/iss3/2/>.

131. See MCCONNELL, *supra* note 85, at 355–56, 358–59; see also *Conditional (Computer Programming)*, WIKIPEDIA, [http://en.wikipedia.org/wiki/Conditional_\(computer_programming\)](http://en.wikipedia.org/wiki/Conditional_(computer_programming)) (last visited Jan. 22, 2015).

132. See MCCONNELL, *supra* note 85, at 445, 453, 456–58, 460.

133. Thomas J. McCabe, *A Complexity Measure*, 2 IEEE TRANSACTIONS ON SOFTWARE ENG’G 308 (1976).

134. *Id.* at 308–10, 318–19.

LAW IS CODE

“provided,” “when,” “where,” “whenever,” “unless,” “notwithstanding,” “in no event,” and “in the event.”¹³⁵ This list is not exhaustive, and we do not expand these root words, but it provides an indication of the exceptions and special cases that are found throughout the U.S. Code. Section 308 of the Senate Legislative Drafting Manual, entitled “Conditional Provisions and Provisos,” offers guidelines on what words to use: it recommends “if” instead of “when” or “where” to indicate a condition, and “except that,” “but,” or “if” instead of phrases involving the word “provided.”¹³⁶ We include both the recommended and non-recommended terms because laws are not obligated to follow these guidelines—an online search of the current U.S. Code shows that all of these terms still exist in the U.S. Code to describe a conditional statement.¹³⁷

5. Summary

Table 1 summarizes the previous four sections. The first column provides brief definitions of these four principles, and the second and third columns identify metrics borrowed from the software engineering community for the historical and current datasets, respectively. In the remainder of this Article, we apply our metrics and show the results of our analyses and visualizations.

Table 1: Description of Principles and Metrics for U.S. Code

Principle	Proposed Metrics (Evolution of U.S. Code)	Proposed Metrics (Current Laws and Titles)
Conciseness: Good code should be as long as it needs to be, but no longer.	Change in total number of words	Total number of words
Change: Code that exhibits large or frequent changes may suggest defects. Large, untested changes can also produce new defects.	Number of words added or deleted Counts of specific words and terms versus time First appearance of words in U.S. Code by title	N/A

135. See *infra* Part IV.D.

136. SENATE DRAFTING MANUAL, *supra* note 76, at 69.

137. See *Search the United States Code*, OFFICE OF THE LAW REVISION COUNSEL: U.S. CODE, <http://uscode.house.gov/search.xhtml> (last visited Jan. 22, 2015).

<p>Coupling: Modular code is more robust and easier to maintain than code with unnecessary cross-dependencies.</p>	<p>Bills affecting similar sections</p>	<p>Size of cross-reference “network core” versus “network periphery” Google PageRank-inspired methods</p>
<p>Complexity: Code with a large number of conditions, cases, and exceptions is difficult to understand and prone to error.</p>	<p>Change in number of condition statements in code</p>	<p>Total number of condition statements by section (cyclomatic complexity)</p>

IV. EVOLUTION OF THE U.S. CODE

To understand the evolution of the U.S. Code, we used the following datasets:

1. Historical U.S. Code texts under license from William S. Hein & Co., Inc.¹³⁸
2. For certain comparisons, a more recent version of the U.S. Code from the OLRC.¹³⁹
3. A document called “Table III,” published by the OLRC, which shows, on a section-by-section basis, how an enrolled bill maps to the U.S. Code.¹⁴⁰

Using our historical dataset, we analyzed and visualized changes to the U.S. Code since 1926. For each of the four software engineering principles listed in the previous section, we comment on insights that emerge from studying the U.S. Code through these metrics.

A. Conciseness: Evolution of the Size of the U.S. Code

Figure 2 is a stacked area graph of the size of the U.S. Code, organized by title and measured in the number of words. Consistent with the popular conception of

138. See *U.S. Code*, *supra* note 84.

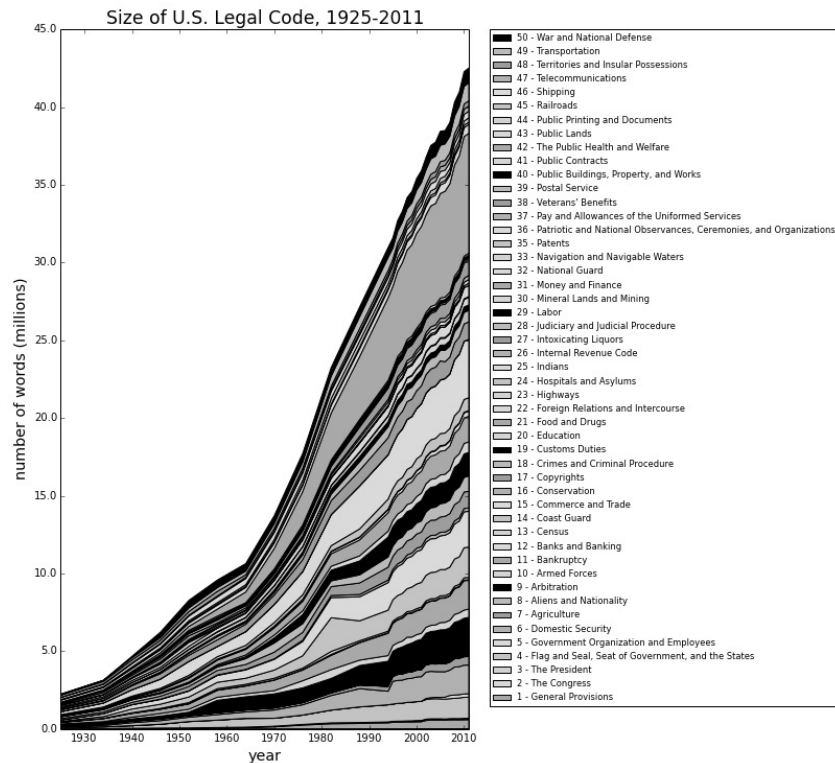
139. See *Current Release Point*, *supra* note 82; *USLM User Guide*, *supra* note 83.

140. *Table III Tool*, *supra* note 119. The authors have written software that parses data from the Table III Tool into machine-readable form for network-based analyses. See *uscode/table 3*, GITHUB, <https://github.com/unitedstates/uscode/tree/master/table3> (last visited Jan. 22, 2015). This free software is available as part of the @unitedstates project. See *id.* An “enrolled bill” is “[t]he final copy of a bill or joint resolution which has passed both chambers in identical form.” See *Glossary*, U.S. SENATE, https://www.senate.gov/reference/glossary_term/enrolled_bill.htm (last visited Jan. 22, 2015).

LAW IS CODE

federal laws, the size of the U.S. Code has grown continuously since 1926.¹⁴¹ Moreover, the rate of growth is increasing.¹⁴²

Figure 2: Number of Words in the U.S. Code by Title



This simple length-based analysis also illustrates that different titles of the U.S. Code are different sizes. For example, Title 42 (Public Health and Welfare) is the longest.¹⁴³

B. Change: Evolution of Content in the U.S. Code

The inadequacy of assessing the U.S. Code by length alone is apparent when analyzing changes over time. For example, despite the increasing size of the U.S. Code, the length is actually the net result of numerous laws enacted and repealed. Thus, the “diff” function allows us to more accurately assess the quality and

141. See *infra* Figure 2.

142. See *infra* Figure 2.

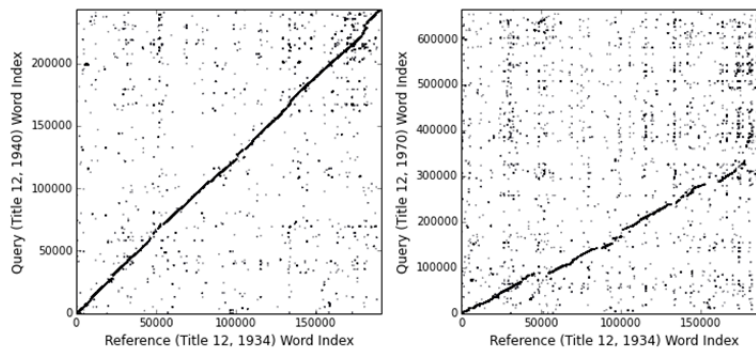
143. See *supra*, Figure 2.

quantity of change. Moreover, the first appearance and frequency of terms in the U.S. Code are also informative for studying its evolution.

1. Addition and Deletion of Words

As described *supra* in Part III, the U.S. Code is not neatly organized into individual lines like software, so we treat each title as a sequence of words to find matches between sequences. Figure 3 shows the output of the document comparison process. We chose Title 12 (Banks and Banking) to illustrate our approach. When there is a matching sequence of words, a black dot is drawn on the plot; when there is a mismatch, no dot is drawn. The following insights can be gained by examining the two plots:

Figure 3: Title 12 (Banks and Banking) Comparisons Between 1934 and 1940 Editions (Left) and 1934 and 1970 Editions (Right).



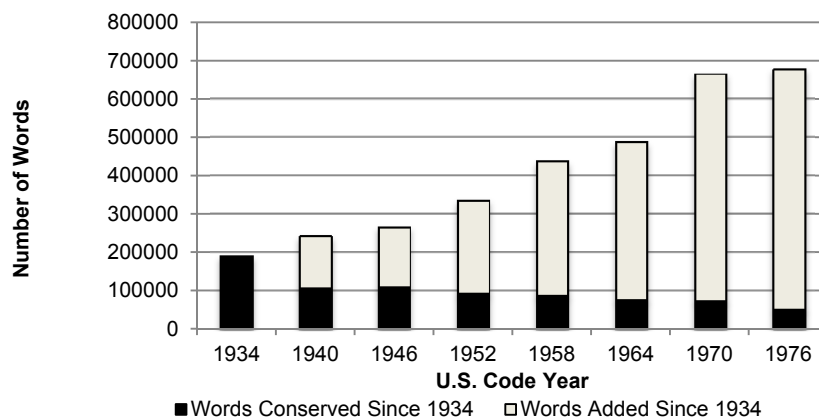
The dark diagonal line from the bottom-left to the top-right of the left plot in Figure 3 indicates that the 1934 and 1940 versions of the Title 12 are largely the same. The relatively small breaks in this dark diagonal line indicate there were relatively few changes between 1934 and 1940. In contrast, the diagonal line is much less intact in the comparison between the 1934 and 1970 versions of the U.S. Code. This pattern indicates that there are large differences between the two documents; that is, there were far more changes between 1934 and 1970 than there were between 1934 and 1940. In particular, large amounts of text were added to the end of Title 12 sometime prior to 1970.

Using the text comparison technique shown above, we can go beyond simply counting the number of words and determine how many words were added and deleted with each subsequent edition of the U.S. Code. Figure 4 summarizes these changes between 1934 and 1976 for Title 12. For graphing purposes, instead of showing “words deleted,” we show “words conserved” in order for the stacked bar graph to show the total number of words in each edition of Title 12. This graph illustrates that the length changes in Title 12 are the product of both the addition of

LAW IS CODE

new passages of text and the deletion of passages of text that existed in 1934, though the vast majority of the length comes from words added.

Figure 4: Words Conserved and Added to Title 12



2. Term Frequency Counts

As a first step toward understanding the content of the U.S. Code, we built a U.S. Code “term-count viewer.” Figure 5 presents some screenshots of our tool to count the frequency of terms by year. Figure 5a illustrates the rise of legislation related to the telephone, and the slow decline of the telegraph.¹⁴⁴ Figure 5b shows how “homeland security” entered the discourse between 2000 and 2006, after September 11, 2001.¹⁴⁵ Figure 5c corresponds to the invention of the credit card and laws related to consumer protection in the 1960s and beyond.¹⁴⁶ These term frequency plots illustrate the attention that legislators and society devoted to new domains of law in different decades.

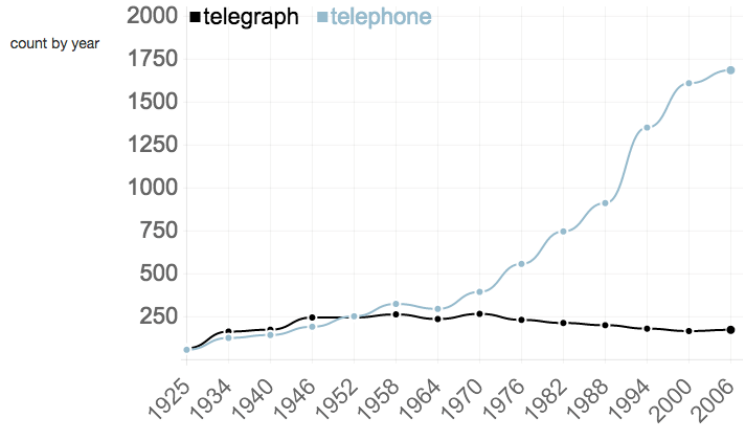
144. See Derek Thompson, *The 100-Year March of Technology in 1 Graph*, ATLANTIC (Apr. 7, 2012, 1:08 PM), <http://www.theatlantic.com/technology/archive/2012/04/the-100-year-march-of-technology-in-1-graph/255573/> (graphing increased adoption of the word “telephone” from 1900–2005).

145. The word “homeland” appears less than twenty times in U.S.C. (2000), but more than 1,700 times in U.S.C. (2006). *Search the United States Code*, *supra* note 137.

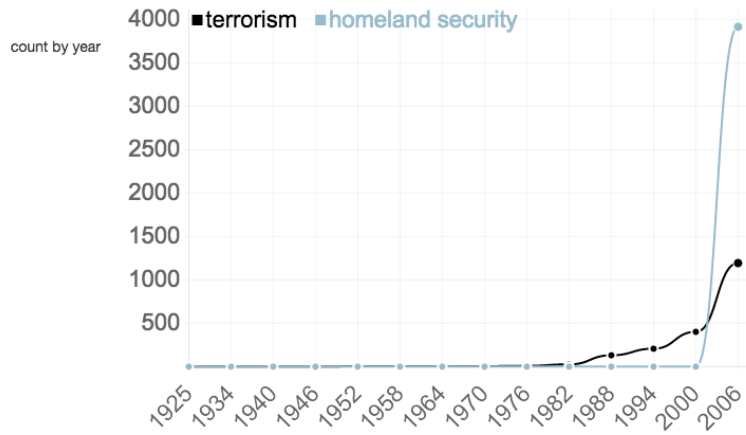
146. Although revolving debt credit cards first appeared in the 1950s, significant regulation did not occur until the 1960s. See John T. Finley, *Consumer (Bankcard) Debt and Regulation—Are Things Working?*, 17 PROC. ACAD. LEGAL, ETHICAL & REG. ISSUES 7 (2013), available at <http://www.alliedacademies.org/public/proceedings/Proceedings32/ALERI%20Proceedings%20Spring%202013.pdf>.

Figure 5: Term Frequency Counts for Selected Phrases.

a. “telegraph” versus “telephone”

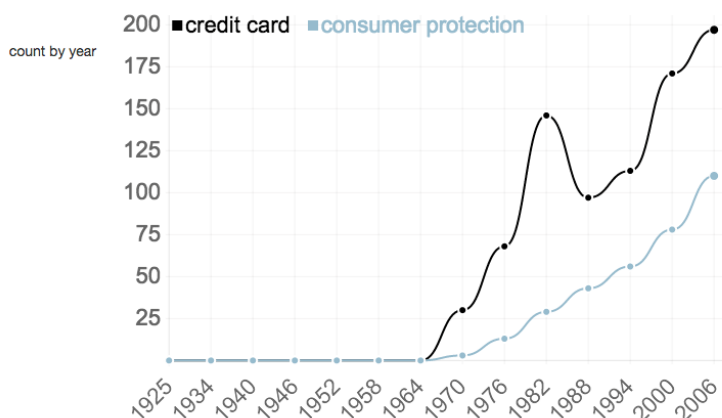


b. “terrorism” versus “homeland security”



LAW IS CODE

c. “credit card” versus “consumer protection”



3. First Appearance of Words

Along with examples of term frequency patterns, we can also examine when words first appeared in the U.S. Code. Table 2 shows new terms that appeared in each edition of the U.S. Code between 1952 and 2006. The top 10 words in terms of their total count in the 2006 edition of the U.S. Code are shown in order to show words that first appeared in a given year and have now become commonplace in the U.S. Code. For example, in our dataset, the term “television” first appears in the 1952 edition of the U.S. Code and can be found 1,297 times.¹⁴⁷

Some terms (such as “Palau” and “Mariana”) reflect routine bookkeeping changes to the U.S. Code, such as changes corresponding to entities that signed Compacts of Free Association with the United States.¹⁴⁸ The timing of other words, such as “television,” “telecommunications,” “pesticide,” or “privacy,” reflect when these concepts and entities first received the attention of federal law. Meanwhile, other terms reflect a change in language usage: the appearance of the term “servicemember(s)” indicates a move away from gender-specific terms.¹⁴⁹

147. See U.S.C. (1952); e.g. 18 U.S.C. § 1343 (“Whoever, having devised or intending to devise any scheme or artifice to defraud, . . . transmits or causes to be transmitted by means of interstate wire, radio, or *television* communication. . . .” (emphasis added)); 26 U.S.C. § 3403(c) (“Parts or accessories (other than tires and inner tubes and other than radio and *television* receiving sets) for any of the articles enumerated in subsection (a) or (b), 8 per centum, except that on and after April 1, 1954, the rate shall be 5 per centum.” (emphasis added)).

148. Compact of Free Association, U.S.-Marshall Islands, Apr. 30, 2003. T.I.A.S. No. 04-501.

149. See, e.g., National Defense Authorization Act for Fiscal Year 2012, Pub. L. No. 112-81, § 631(a), 125 Stat. 1298, 1452 (2011) (“Recognizing the complexities and the changing nature of travel, the amendments made by this section provide the Secretary of Defense and the other administering Secretaries with the authority to prescribe and implement travel and transportation policy that is simple, clear, efficient, and flexible, and that

Table 2: First Appearance of Terms in the U.S. Code

1952	1958	1964	1970	1976
television (1297)	infrastructure (1341)	environmental (5811)	medicare (2553)	chairperson (1112)
operational (1210)	rulemaking (1062)	guidelines (3477)	subclause (1598)	terrorist (865)
pipeline (1199)	pesticide (918)	technologies (2111)	expertise (1149)	medicaid (683)
terrorism (1193)	enhancement (915)	providers (1859)	strategies (964)	update (615)
workforce (1159)	micronesia (830)	computer (1341)	outreach (889)	digital (528)
telecommunications (1068)	inpatient (829)	mariana (1318)	ensuring (781)	methodology (524)
victim (1013)	elderly (791)	environ (1150)	innovative (741)	software (459)
reconciliation (975)	confidentiality (716)	monitor (1099)	oceanic (722)	amtrak (455)
satellite (912)	statewide (700)	evaluations (907)	affordable (719)	syndrome (432)
significantly (823)	global (685)	privacy (788)	initiatives (684)	underserved (377)

Table 2: continued

1982	1988	1994	2000	2006
palau (566) ¹⁵⁰	database (397)	internet (754)	tricare (331) ¹⁵¹	servicemember (161)

meets mission and *servicemember* needs. . . .” (emphasis added)); Servicemembers Civil Relief Act, Pub. L. No. 108-189, 117 Stat. 2835 (2003).

150. The Republic of Palau signed a Compact of Free Association with the United States in 1982. Sophie Foster, *Palau*, ENCYCLOPEDIA BRITANNICA, <http://www.britannica.com/EBchecked/topic/439255/Palau/54026/History#ref513942> (last updated Mar. 27, 2014); see *Approval of Compact of Free Association*, 48 U.S.C. § 1931 (2012).

151. TRICARE is a healthcare program for active duty service members, National Guard and Reserve members, retirees, and their families. *Welcome, TRICARE*, <http://www.tricare.mil/> (last visited Jan. 27, 2015).

LAW IS CODE

nonproliferation (266)	noncustodial (135)	countervailable (145)	cbtpa (83) ¹⁵²	atpdea (61) ¹⁵³
targeted (454)	affordability (206)	nafta (504)	website (199)	pdp (153) ¹⁵⁴
saharan (415)	remic (194) ¹⁵⁵	stalking (370)	y2k (127)	cafta (137) ¹⁵⁶
assistive (330) ¹⁵⁷	kg (188)	geospatial (232)	biobased (126)	darfur (100)
swap (316)	privatization (177)	mentoring (210)	hubzone (112) ¹⁵⁸	restyling (94)
hospice (300)	servicemembers (163)	biodiesel (160)	bliley (108) ¹⁵⁹	nanotechnology (77)
competitiveness (289)	alzheimer's (154)	nonoriginating (151)	vento (103) ¹⁶⁰	safetea (75) ¹⁶¹
nonattainment (272)	mammography (148)	databases (148)	telehealth (100)	katrina (67)
fueled (267)	forensic (142)	empowerment (148)	hass (93)	pandemic (63)

152. Caribbean Basin Trade Partnership Act, Pub. L. No. 106-200, 114 Stat. 251 (2000) (codified in scattered sections of 19 U.S.C.).

153. Andean Trade Promotion and Drug Eradication Act, Pub. L. No. 107-210, 116 Stat. 933, 1023 (2002).

154. "PDP" stands for "prescription drug plan." See, e.g., Patient Protection and Affordable Care Act, Pub. L. No. 111-148, § 3209, 124 Stat. 119, 460 (2010); Medicare Prescription Drug, Improvement, and Modernization Act of 2003, Pub. L. No. 108-173, § 101, 117 Stat. 2066, 2073 (2003).

155. "Remic" is an acronym for "real estate mortgage investment conduit." See INTERNAL REVENUE SERV., U.S. DEP'T OF THE TREASURY, PUB. NO. 938, REAL ESTATE MORTGAGE INVESTMENT CONDUITS (REMICs) REPORTING INFORMATION 1 (2014).

156. "CAFTA" is an acronym for the Central America Free Trade Agreement. See *CAFTA-DR (Dominican Republic-Central America FTA)*, OFFICE OF THE U.S. TRADE REPRESENTATIVE, <https://ustr.gov/trade-agreements/free-trade-agreements/cafta-dr-dominican-republic-central-america-fta> (last visited Jan. 22, 2015).

157. Assistive Technology Act of 2004, Pub. L. No. 105-394, 112 Stat. 3627 (codified as amended at 29 U.S.C. §§ 3001-3058 (2012)).

158. The Historically Underutilized Business Zones ("HUBZone") program helps small business in urban and rural communities gain preferential access to federal procurement opportunities. See *HUBZone Project*, U.S. SMALL BUS. ADMIN., <https://www.sba.gov/category/navigation-structure/contracting/contracting-support-small-businesses/small-business-cert-0> (last visited Jan. 27, 2015).

159. Gramm-Leach-Bliley Act, Pub. L. No. 106-102, 113 Stat. 1338 (codified at 15 U.S.C. §§ 6801-6809 (2012)).

160. McKinney-Vento Homeless Assistance Act of 1987, Pub. L. No. 100-77, 101 Stat. 482 (codified as amended at 42 U.S.C. §§ 11301-11489 (2012)).

161. Safe, Accountable, Flexible, Efficient Transportation Equity Act, Pub. L. No. 109-59, 119 Stat. 1144 (2005).

4. Trajectories of Terms in U.S. Code Titles

Finally, instead of merely listing the appearance of new terms in the U.S. Code, we can also examine the contexts in which they are used. In particular, our historical dataset makes it possible to track terms of interest across different titles of the U.S. Code. In Figure 6, we show that the term “whistleblower” first appeared in Titles 5 (Government Organization and Employees), 42 (Public Health and Welfare), 31 (Money and Finance), and 10 (Armed Forces) in 1994.¹⁶² It now is mentioned in a total of 11 U.S. Code titles. Meanwhile, “privacy” is mentioned throughout the U.S. Code. Interestingly, its first appearance was in 1964 in Title 39 (Postal Service).¹⁶³ This visualization reveals when discourse framed around whistleblowers or privacy entered different titles of the U.S. Code.

Figure 6: Appearance of “Whistleblower” in U.S. Code by Year and Title

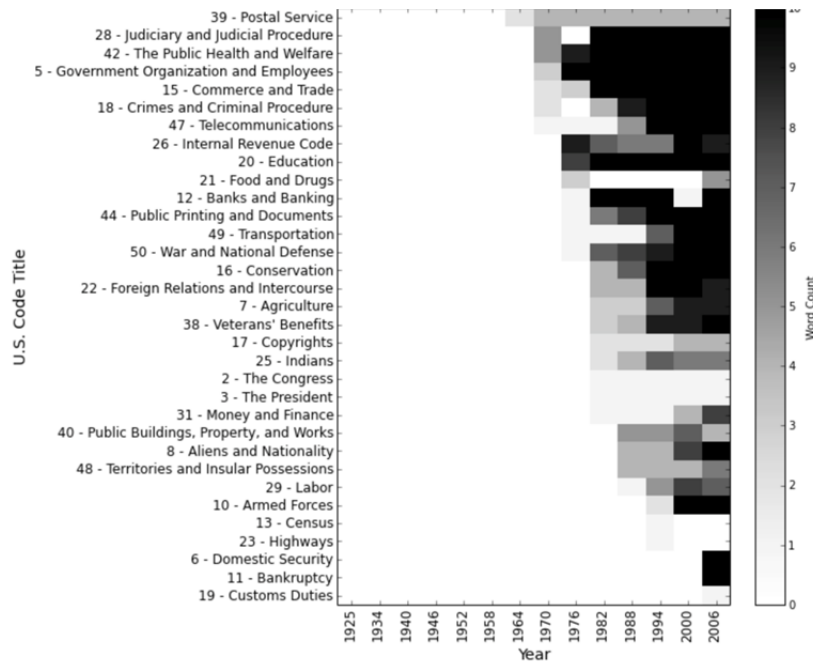


162. See *Search the United States Code*, *supra* note 137.

163. Privacy of Accounts, 39 U.S.C. § 5212 (1964).

LAW IS CODE

Figure 7: Appearance of “Privacy” in U.S. Code Titles by Year and Title



C. Coupling: Evolution of Structure of U.S. Code

When Congress passes a bill and the President signs it into law, the OLRC incorporates the new law into the U.S. Code.¹⁶⁴ The OLRC keeps an online record of the mapping of every bill section to its corresponding section in the U.S. Code.¹⁶⁵ This mapping of bill sections to U.S. Code sections forms a network connection map (a “graph” in computer science terms).

One application of this graph is to determine similar bills in terms of the overlap of the U.S. Code sections that they affect. If newly enacted laws are like new additions to software, then we can determine quantitatively which existing laws were changed or impacted most. We use the Jaccard similarity,¹⁶⁶ a mathematical measure of overlap of sets of entities, to calculate how similar two laws are in terms of sections affected: two bills that affect the exact same U.S. Code sections would

164. *About Classification of Laws to the United States Code*, *supra* note 65.

165. *See Table III Tool*, *supra* note 119.

166. *See* Sheetal A. Takale & Sushma S. Nandgaonkar, *Measuring Semantic Similarity Between Words Using Web Documents*, INT’L J. ADVANCED COMPUTER SCI. & APPLICATIONS, Oct. 2010, at 78, 82; R. Real, *Tables of Significant Values of Jaccard’s Index of Similarity*, 22 MISCELLANEA ZOOLOGICA 29, 30 (1999).

have a Jaccard similarity of 1.0, while two bills that affect completely different sets of sections would have a Jaccard similarity of 0.0.¹⁶⁷

As an illustration of this method, Table 3 shows the most similar bills to the Dodd-Frank Wall Street Reform and Consumer Protection Act while Table 4 shows similar bills to the Patient Protection and Affordable Care Act.¹⁶⁸ Notably, the bill ranked most similar to Dodd-Frank is Public Law 101-73 (the Financial Institutions Reform, Recovery, and Enforcement Act of 1989), which was the legislative response to the Savings and Loan Crisis in the late 1980s.¹⁶⁹ The list also contains other landmark pieces of legislation related to the financial sector at different points in the 20th century.

Table 3: Bills with Highest Similarity to Dodd-Frank Wall Street Reform and Consumer Protection Act

Rank (by Jaccard Similarity)	Public Law No.	Bill Name	Number of Sections in Bill	Jaccard Similarity
1	101-73	Financial Institutions Reform, Recovery, and Enforcement Act of 1989	284	0.113
2	90-321	Consumer Credit Protection Act	188	0.112
3	73-291	Securities Exchange Act of 1934	87	0.072
4	102-242	Federal Deposit Insurance Corporation Improvement Act of 1991	173	0.071
5	103-325	Riegle Community Development and Regulatory Improvement Act of 1994	245	0.052
6	95-630	Financial Institutions Regulatory and Interest Rate Control Act of 1978	166	0.051
7	96-221	Depository Institutions Deregulation and Monetary Control Act of 1980	125	0.05
8	106-102	Gramm-Leach-Bliley Act	155	0.049
9	102-550	Housing and Community	558	0.049

167. See Takale & Nandgaonkar, *supra* note 166, at 82; Real, *supra* note 166, at 30.

168. See Dodd-Frank Wall Street Reform and Consumer Protection Act, Pub. L. No. 111-203, 124 Stat. 1376 (2010); Patient Protection and Affordable Care Act, Pub. L. No. 111-148, 124 Stat. 119 (2010).

169. See Paul T. Clark et al., *Regulation of Savings Associations Under the Financial Institutions Reform, Recovery, and Enforcement Act of 1989*, 45 BUS. LAW. 1013, 1013 (1990).

LAW IS CODE

		Development Act of 1992		
10	100-181	Securities and Exchange Commission Authorization Act of 1987	63	0.049

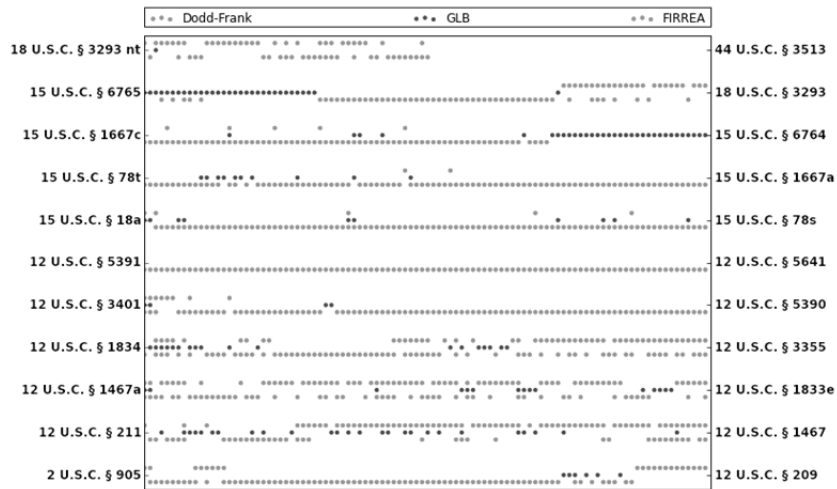
Table 4: Bills with Highest Similarity to Patient Protection and Affordable Care Act

Rank (by Jaccard Similarity)	Public Law No.	Bill Name	Number of Sections in Bill	Jaccard Similarity
1	74-271	Social Security Act of 1935	538	0.129
2	108-173	Medicare Prescription Drug, Improvement, and Modernization Act of 2003	234	0.122
3	94-437	Indian Health Care Improvement Act	156	0.12
4	78-410	Public Health Service Act of 1944	1227	0.103
5	105-33	Balanced Budget Act of 1997	424	0.086
6	102-573	Indian Health Amendments of 1992	141	0.081
7	111-152	Health Care and Education Reconciliation Act of 2010	116	0.074
8	110-275	Medicare Improvements for Patients and Providers Act of 2008	84	0.071
9	101-239	Omnibus Budget Reconciliation Act of 1989	706	0.062
10	101-508	Omnibus Budget Reconciliation Act of 1990	938	0.06

We can also visualize this similarity. Figure 8 shows show the sections of the U.S. Code affected by Dodd-Frank and similar laws, while Figure 9 shows laws similar to PPACA. For each of the bills, each dot represents a section of the U.S. Code. These dots are ordered by U.S. Code section number. Because these bills have a very large number of sections, they need to be shown in multiple rows. Only sections affected by at least one of the bills are represented, and the notations on the side indicate the sections corresponding to the first and last dots on each row. Stacked dots in the

same row indicate multiple bills affected those sections. For example, in Figure 9, as shown by the annotations, all three bills affected 42 U.S.C § 1395yy, but only PPACA affected sections in 42 U.S.C. § 280.¹⁷⁰ In the case of both PPACA and Dodd-Frank, it is worth noting that these laws, in addition to amending many existing sections related to other key bills, also created entirely new sections in the U.S. Code, which may explain why they did not overlap more with previous bills.¹⁷¹

Figure 8: Comparisons of Sections of the U.S. Code affected by Dodd-Frank Wall Street Reform and Consumer Protection Act, Financial Institutions Reform, Recovery, and Enforcement Act of 1989 (FIRREA), and Gramm-Leach-Bliley Act (GLB)

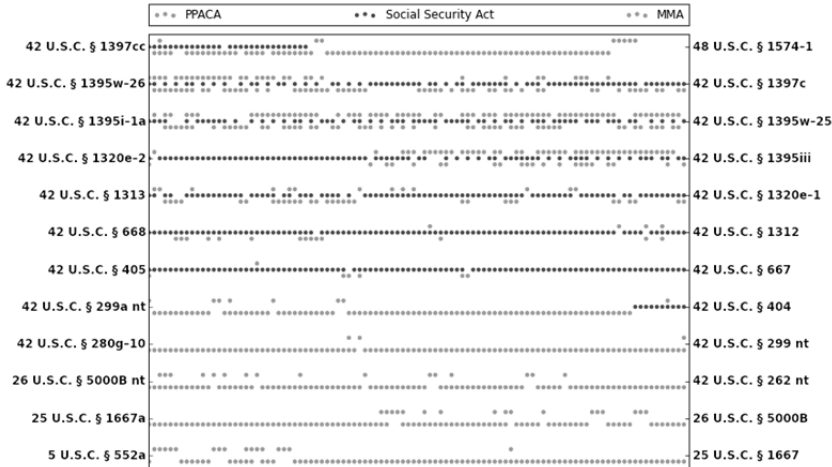


170. 42 U.S.C. §§ 280, 1395yy (2012).

171. E.g., Dodd-Frank Wall Street Reform and Consumer Protection Act § 748 (codified at 7 U.S.C. § 26 (2012)) (setting rewards for whistleblowers); Patient Protection and Affordable Care Act § 3021(a) (codified at 42 U.S.C. § 300jj-51 (2012)) (establishing the Center for Medicare and Medicaid Innovation).

LAW IS CODE

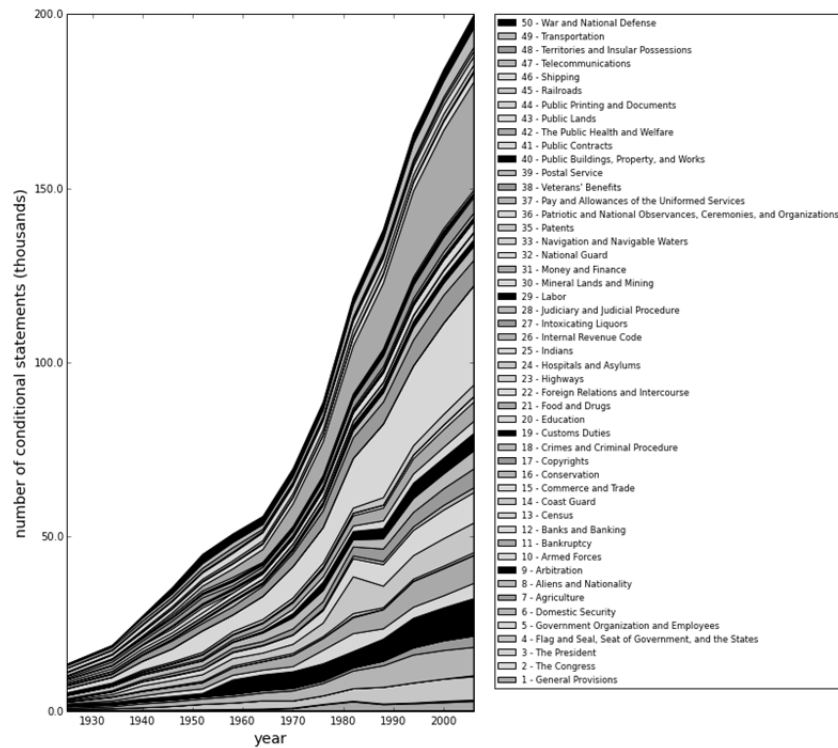
Figure 9: Comparison of sections of the U.S. Code affected by Patient Protection and Affordable Care Act (PPACA), Social Security Act of 1935, and Medicare Prescription Drug, Improvement, and Modernization Act of 2003 (MMA).



D. Complexity: Evolution of Conditional Statements in the U.S. Code

Similar to measuring length, we can count the number of conditional statements by title in the U.S. Code over time. The results are shown in Figure 10. As with the length measurement, the number of conditional statements has also grown substantially over time. In the next two sections of this Article, we identify and explore titles and specific laws with particularly high cyclomatic complexity, which indicate parts of the U.S. Code that are particularly difficult to understand.

Figure 10: Cyclomatic complexity (Number of Conditional Statements) in U.S. Code



V. STRUCTURE OF CURRENT LAWS: 111TH CONGRESS

This Section examines laws passed by the 111th Congress to determine whether our software engineering approaches can help identify the most complex laws that may be, consequently, prone to unintended consequences. The 111th Congress spanned the period from January 3, 2009 to January 3, 2011.¹⁷² Some notable laws that it passed included Public Law 111-5, the American Recovery and Reinvestment Act of 2009 (“ARRA,” or, informally, the “stimulus bill”);¹⁷³ Public Law 111-148, the Patient Protection and Affordable Care Act (“PPACA,” or, informally, “Obamacare”);¹⁷⁴ and Public Law 111-203, the Dodd-Frank Wall Street Reform and

172. *Past Days in Session of the U.S. Congress*, CONGRESS.GOV, <https://congress.gov/past-days-in-session> (last visited Jan. 22, 2015).

173. American Reinvestment and Recovery Act of 2009, Pub. L. No. 111-5, 123 Stat. 115.

174. Patient Protection and Affordable Care Act, Pub. L. No. 111-148, 124 Stat. 119 (2010).

LAW IS CODE

Consumer Protection Act (informally, “Dodd-Frank”).¹⁷⁵ Our goal in this Section and the next is to measure quantitatively the complexity of these laws, and use these measures to identify the effect that these laws had on the overall complexity of the U.S. Code.¹⁷⁶

Our main results lead to three conclusions. First, laws that would be classified as “complex” or “important” by a human reader, such as PPACA or Dodd-Frank, are also very complex according to our software metrics.¹⁷⁷ Second, the average law is not very complex according to our measures.¹⁷⁸ Combined with our first point, this implies that there is a level of agreement between our techniques to identify complex laws and our findings with PPACA or Dodd-Frank. Third, our coupling metric helps identify two categories of “lengthy laws.” The first type is appropriations acts, which are very long but do not have a high degree of coupling with the U.S. Code.¹⁷⁹ The second type includes laws such as PPACA, ARRA, or the extension of the Bush-era tax cuts in 2010, which show a high degree of coupling with the U.S. Code.¹⁸⁰ This coupling suggests that the content of these laws are more embedded in the “core” of the U.S. Code. Thus, our coupling measure can help quantify the extent to which laws have a more fundamental, structural effect on the U.S. Code.

While we focus on laws enacted by Congress, it is important to highlight that our techniques can be used in the future to analyze proposed laws. Our measure of coupling can give insights on how a proposed law will affect the rest of the U.S. Code. Our other measures can be used to compare two versions of a bill, and identify which sections of a bill can or should be simplified.

We show the top five laws passed by the 111th Congress according to length (Table 5), coupling (Table 6), and complexity (Table 7). The results confirm, in a quantitative way, the intuition that laws such as Omnibus Appropriations Act, PPACA, and Dodd-Frank are complex.

175. Dodd-Frank Wall Street Reform and Consumer Protection Act, Pub. L. No. 111-203, 124 Stat. 1376 (2010).

176. In Part VI, we study the years 1995–2012 and show that, while the laws passed by the 111th Congress are complex, they are not uniquely so. See *infra* Part VI. No correlation exists between complexity and the party that controls Congress or the Presidency, and no pattern associates complexity to the presence or absence of a gridlocked government. See *infra* Apps. B, C. To the contrary, most complex laws seem to correspond to the 104th Congress (1995–97), which is well known for its disagreements between the executive and legislative branches, including a government shutdown. See *1995–96 Government Shutdown*, BANCROFT LIBR., <http://bancroft.berkeley.edu/ROHO/projects/debt/governmentsshutdown.html> (last updated Oct. 2, 2013); *infra* Apps. B, C.

177. Patient Protection and Affordable Care Act; Dodd-Frank Wall Street Reform and Consumer Protection Act.

178. See discussion *infra* Part VI.C., App. D.

179. See discussion *infra* Tables 5–7, Figures 11–15.

180. See *infra* Tables 5, 6; App. B.

Table 5: Laws from the 111th Congress Ranked by Length

Public Law Number	Popular Name	Length (number of words)
111-11	Omnibus Public Land Management Act	191,864
111-8	Omnibus Appropriations Act	216,534
111-84	National Defense Authorization Act	274,329
111-203	Dodd-Frank	364,844
111-148	PPACA	384,324

Table 6: Laws from the 111th Congress Ranked by Coupling
(The coupling metric used is the number of sections in the law that also belong to the core of the U.S. Code)

Public Law Number	Popular Name	Number of Sections in Core of U.S. Code
111-84	National Defense Authorization Act	143
111-312	Tax Relief, Unemployment Insurance Reauthorization and Job Creation Act	199
111-203	Dodd-Frank	232
111-148	PPACA	251
111-5	Stimulus Act	293

Table 7: Laws from the 111th Congress Ranked According to Cyclomatic Complexity

Public Law Number	Popular Name	Cyclomatic Complexity
111-5	Stimulus Act	805
111-117	Consolidated Appropriations Act	1130
111-148	PPACA	1225

LAW IS CODE

111-203	Dodd-Frank	1384
111-8	Omnibus Appropriations Act	1414

It is reasonable to ask whether such complexity is significant. How much of an outlier are these particular laws from an average law enacted during the 111th Congress? This question is answered by examining the distributions of length (Figure 11), coupling (Figure 12), and complexity (Figure 13), which show the distributions of our metrics. These distributions are very thin-tailed, implying that the occurrence of these highly ranked laws is very low. Indeed, most laws have much lower values of these metrics.

Figure 11: Distribution of Lengths of Laws Passed by 111th Congress

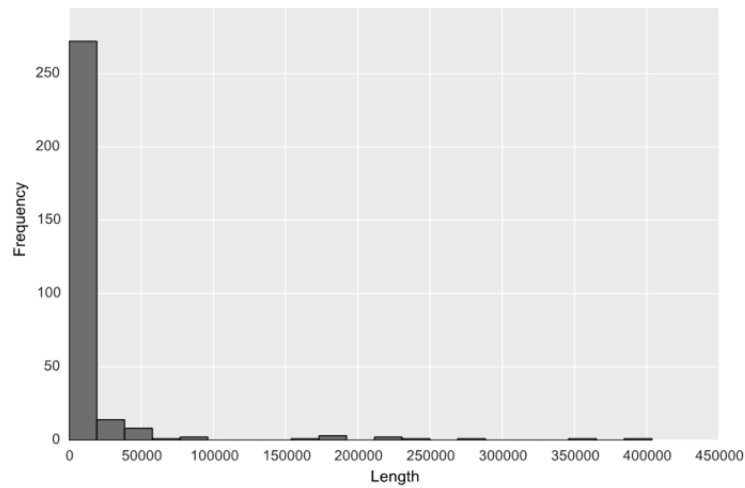


Figure 12: Distribution of Coupling Metric for Laws Passed by 111th Congress

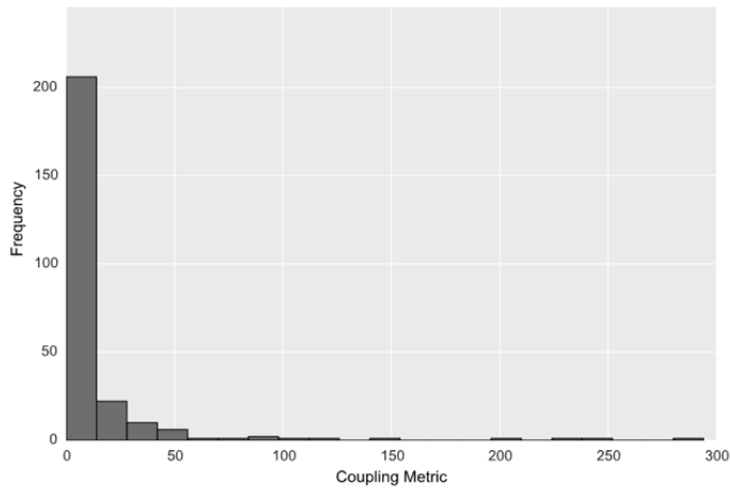
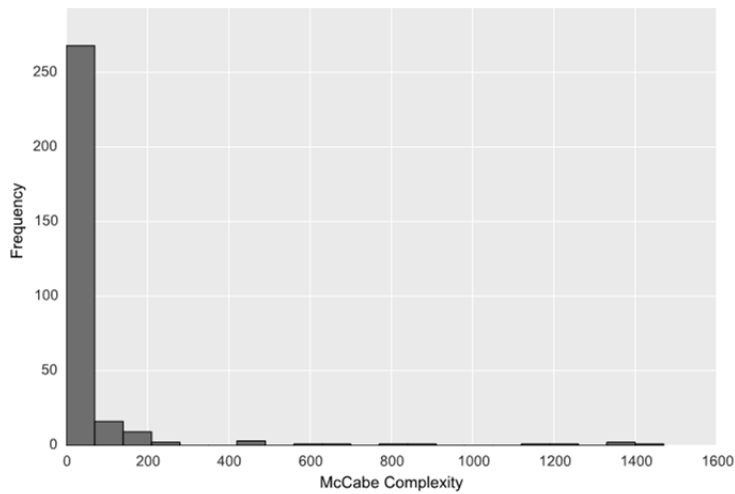


Figure 13: Distribution of Cyclomatic Complexity for Laws Passed by the 111th Congress



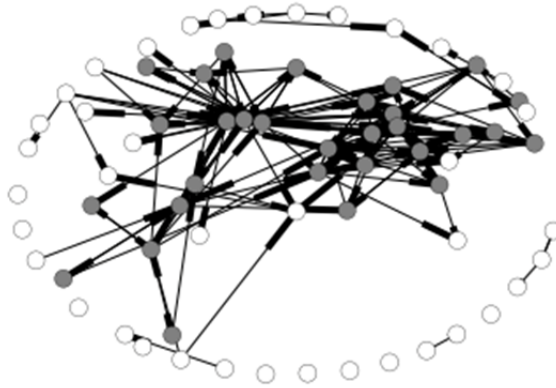
One interesting observation is that the Omnibus Appropriations Act appears highly ranked with respect to all measures of complexity except coupling. For instance, it has the highest cyclomatic complexity, which is unsurprising since the Omnibus Appropriations Act contains multiple miscellaneous funding

LAW IS CODE

authorizations that should not permanently affect other areas of the U.S. Code.¹⁸¹ In contrast, a law such as PPACA is not only complex with respect to length and cyclomatic complexity, but also has a high degree of coupling with the rest of the U.S. Code, and has a large intersection with the largest strongly connected component (the core) of the U.S. Code.¹⁸² In this respect, we can say that PPACA has a higher impact on the U.S. Code than the Omnibus Appropriations Act.

We can explore this argument further by analyzing the network structure of these laws. Each piece of legislation affects a subset of the U.S. Code. While the overall U.S. Code is too large to visualize easily, the subsets of the U.S. Code modified by individual bills are small enough that visualization is helpful. As examples, Figure 14 shows PPACA, while Figure 15 shows the Omnibus Appropriations Bill from the 111th Congress.

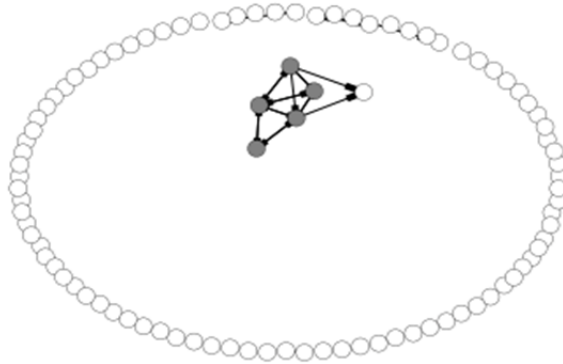
Figure 14: Sections of the U.S. Code Modified by PPACA
(Nodes in grey belong to the largest connected component in this graph, which can be interpreted as the core of PPACA)



181. Omnibus Appropriations Act, 2009, Pub. L. No. 111-8, 123 Stat. 524.

182. See Patient Protection and Affordable Care Act, Pub. L. No. 111-148, 124 Stat. 119 (2010). The U.S. Code's core contains the most interconnected sections. See *infra* App. D. Appendix A formally defines "core." See *infra* App. A.

Figure 15: Sections of the U.S. Code Modified by the Omnibus Appropriations Act of 2009
(Nodes in grey belong to the largest connected component in this graph, which can be interpreted as the core of the bill)



We highlight nodes in these networks with two shades. Nodes in grey represent the *core of the law*,¹⁸³ while nodes in white represent the remaining sections of the law. That is, the nodes in grey in Figure 14 represent the largest connected component of PPACA, while the nodes in grey in Figure 15 represent the largest connected component of the Omnibus Appropriations Bill. Our graph layout algorithm places nodes in a circular fashion, with nodes with high levels of connectivity drawn more toward the center of the graph. As the figures indicate, PPACA has many more interconnections between its sections. On the other hand, there are almost no cross-citations behind sections of the Omnibus Appropriations Bill. As Figure 15 shows, there are six sections (*i.e.* six nodes) that cite each other and form the core of the bill.¹⁸⁴ Thus, the Omnibus Appropriations Bill has a much lower degree of coupling than PPACA.

Appendices B and C show that these properties are not a fluke. Appendix B examines all appropriations bills passed since the 104th Congress. Each law corresponds to a figure in the appendix, which shows only the *core of the law*. As Appendix B shows, appropriations bills generally have very small cores. Appendix C, in contrast, shows the bills passed since the 104th Congress that have cores larger than 50.

One important conclusion from these results is that, even though appropriations bills are large, an expert reader can understand one section of it without needing to

183. See *infra* App. A.

184. See Omnibus Appropriations Act, 2009; *supra* Figure 15.

understand many other sections. In this respect, appropriations bills are simple. On the other hand, many of the sections of PPACA are coupled with other sections. To understand the impact of one section, an expert needs to understand the law as a whole, and may need to follow many levels of citations in the act. In this respect, PPACA requires nonlinear, careful reading, making it very complex and challenging to understand. This insight emerges from examining the network structure of the law.

VI. STRUCTURE OF THE CURRENT U.S. CODE: TITLES 12 (BANKS AND BANKING) AND 26 (INTERNAL REVENUE SERVICE)

In this Section, we use our techniques to perform case studies of two very complex U.S. Code titles: Title 12 (Banks and Banking) and Title 26 (Internal Revenue Code).¹⁸⁵ Using our techniques, we can identify the sections with:

1. the highest complexity, according to the cyclomatic measure of conditional statement counts; and
2. the highest degree of coupling, according to our core-periphery analysis.

Cyclomatic complexity will give us sections that have a high level of branching, and are therefore difficult to interpret without considering multiple conditional scenarios. The PageRank metric will show sections that, when modified, have a large probability of affecting other sections in their respective titles.¹⁸⁶

A. Case Study of Title 12 (Banks and Banking)

Title 12 contains laws related to banks and banking institutions.¹⁸⁷ The banking sector in the United States has, as a result of consolidation and innovation, become more complex—today's financial institutions are involved in a wide array of transactions and activities that simply did not exist a generation ago.¹⁸⁸

Along with multiple waves of financial crises and regulatory activity throughout the 20th and 21st centuries,¹⁸⁹ we argue that Title 12 can be challenging for the non-specialist to understand.¹⁹⁰ Our goal is to analyze and visualize the structure of Title 12, as well as to pinpoint areas that are especially complicated. We do this by

185. See 12 U.S.C. (2012); 26 U.S.C. (2012).

186. See *supra* notes 125–26 and accompanying text; discussion *infra* Parts VI.A, VI.B.

187. Banks and Banking, 12 U.S.C. (2012).

188. See Lisa M. DeFerrari & David E. Palmer, *Supervision of Large Complex Banking Organizations*, FED. RES. BULL., Feb. 2001, at 47.

189. See Laureen Snider, *The Conundrum of Financial Regulation: Origins, Controversies, and Prospects*, 7 ANN. REV. L. & SOC. SCI. 121, 123–28 (2011).

190. See 12 U.S.C. (2012).

computing the cyclomatic complexity of each section of Title 12, and Table 8 reports the sections with the highest complexity. Any effort to reform banking regulation should begin with a systematic refactoring and simplification of these sections.

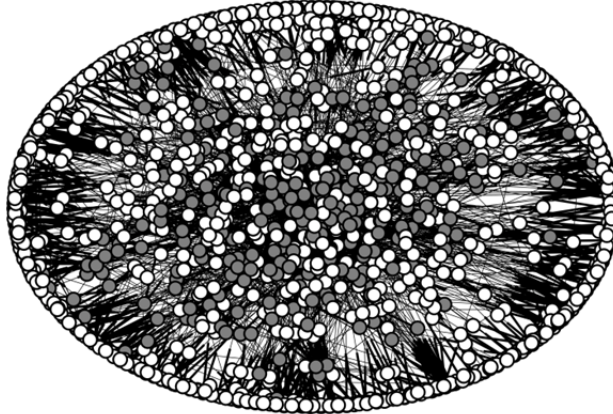
Table 8: Sections of Title 12 with Highest Cyclomatic Complexity

Section Number	Name	Number of Conditional Terms
§ 5390	Power and duties of the corporation	187
§ 1821	Insurance Funds	183
§ 1464	Federal savings associations	138
§ 1715l	Housing for moderate income and displaced families	130
§ 1467a	Regulation of holding companies	128

Another tool we can use is the network of citations that is produced by Title 12. This network can be visualized in Figure 16, which shows a very dense graph. Nodes highlighted in grey correspond to the core of this graph, and make up a significant fraction of Title 12. Thus, even a slight modification to a section of Title 12 is likely to have large repercussions across all other sections, and Figure 16 provides a systematic way to gauge such repercussions before any modification is implemented.

LAW IS CODE

Figure 16: Core-Periphery Network of Title 12 (Banks and Banking)



While it is helpful to visualize Title 12 as a network in this way, it is hard to specify the most “influential” sections of the Title just by looking at this network. It would seem, from the metrics, that all sections in the core of Title 12 would be just as influential. In order to break this tie, we introduce the PageRank metric, which is frequently used in network analysis and has been used to rank the importance of web pages for Internet search engines.

Table 9 gives the nodes in Title 12 with the highest PageRank. The one with the highest PageRank (and therefore the most influential under this metric) is 12 U.S.C. § 1481, which is the Bank Holding Company Act’s definitions section.¹⁹¹ This fact suggests that, if the definitions in this section were to be amended by a financial reform, then it would have a significant impact on the interpretation of all other sections of Title 12.

Table 9: Title 12 Sections with Highest PageRank

Section Number	Name	Beginning Excerpt
1841	Bank Holding Company Act Definitions	Except as provided in paragraph (5) of this subsection, “bank holding company” means any company which has control over any bank or over any company that is or becomes a bank holding company by virtue of this chapter.
101	Repealed	Section 101, acts Mar. 14, 1900, ch. 41, § 12, 31 Stat. 49; Oct. 5, 1917, ch. 74, § 2,40 Stat. 342, provided for delivery

¹⁹¹ 12 U.S.C. § 1481 (2012).

		of circulating notes in blank to national banking associations depositing bonds with Treasurer of United States.
1818	Termination of Status as Insured Depository Institution	<p>(a) Termination of insurance</p> <p>(1) Voluntary termination</p> <p>Any insured depository institution which is not—</p> <ul style="list-style-type: none"> (A) a national member bank; (B) a State member bank; (C) a Federal branch; (D) a Federal savings association; or (E) an insured branch which is required to be insured under subsection (a) or (b) of section 3104 of this title, <p>may terminate such depository institution’s status as an insured depository institution if such insured institution provides written notice to the Corporation of the institution’s intent to terminate such status not less than 90 days before the effective date of such termination.</p>
1709	Insurance of Mortgages	<p>(a) Authorization</p> <p>The Secretary is authorized, upon application by the mortgagee, to insure as hereinafter provided any mortgage offered to him which is eligible for insurance as hereinafter provided, and, upon such terms as the Secretary may prescribe, to make commitments for the insuring of such mortgages prior to the date of their execution or disbursement thereon.</p>
1813	Federal Deposit Insurance Act Definitions	<p>(a) Definitions of bank and related terms</p> <p>(1) Bank</p> <p>The term “bank”—</p> <ul style="list-style-type: none"> (A) means any national bank and State bank, and any Federal branch and insured branch; (B) includes any former savings association.

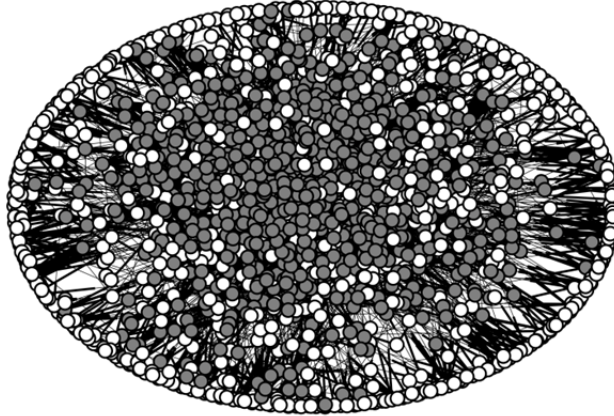
B. Case Study of Title 26 (Internal Revenue Code)

We apply the same analysis to Title 26 (Internal Revenue Code), which is known to be a very complex title of the U.S. Code.¹⁹² Figure 17 shows the network structure induced by cross-citations in Title 26, with nodes in grey again showing nodes that are in the core of the title. As we can see, Title 26 is even denser and has a larger core than Title 12.

192. The 2012 National Taxpayer Advocate’s Report to Congress declared: “The most serious problem facing taxpayers—and the IRS—is the complexity of the Internal Revenue Code (tax code).” NAT’L TAXPAYER ADVOCATE, 2012 ANNUAL REPORT TO CONGRESS 3 (2012), <http://www.taxpayeradvocate.irs.gov/2012-Annual-Report/downloads/Most-Serious-Problems-Tax-Code-Complexity.pdf>. The Report estimated that Americans spend 6.1 billion hours per year on tax compliance. *Id.*

LAW IS CODE

Figure 17: Core-Periphery Network of Title 26 (Internal Revenue Code)



We can use cyclomatic complexity and PageRank to find significantly complex sections of Title 26. Table 10 shows the Title 26 sections with the highest cyclomatic complexity. Table 11 gives the sections in Title 26 with the highest PageRank. The one with the highest PageRank (and therefore the most influential under this metric) is 26 U.S.C. § 501, which defines exemptions from taxation.¹⁹³ As in our analysis of Title 12, this fact implies that changing these exemptions would have a wide reaching effect on the rest of Title 26.

Table 10: Sections of Title 26 with Highest Cyclomatic Complexity

Section Number	Name	Number of Conditional Terms
§ 168	Accelerated cost recovery system	392
§ 401	Qualified pension, profit-sharing, and stock bonus plan	344
§ 141	Private activity bond; qualified bond	213
§ 3121	Definitions [Subchapter C – General Provisions]	201
§ 42	Low-income housing credit	195

¹⁹³ 26 U.S.C. § 501 (2012).

Table 11: Title 26 (Internal Revenue Code) Sections with Highest PageRank

Section Number	Name	First Clause												
501	Exemption from tax on corporations, certain trusts, etc.	(a) Exemption from taxation An organization described in subsection (c) or (d) or section 401 (a) shall be exempt from taxation under this subtitle unless such exemption is denied under section 502 or 503.												
1564	Repealed	Section, added Pub. L. 91-172, title IV, § 401(b)(1), Dec. 30, 1969, 83 Stat. 600; amended Pub. L. 94-455, title XIX, §§ 1901(b)(1)(J)(vi), (21)(A)(ii), 1906(b)(13)(A), Oct. 4, 1976, 90 Stat. 1791, 1797, 1834, related to transitional rules in the case of certain controlled corporations.												
1	Tax Imposed	(a) Married individuals filing joint returns and surviving spouses There is hereby imposed on the taxable income of— (1) every married individual (as defined in section 7703) who makes a single return jointly with his spouse under section 6013, and (2) every surviving spouse (as defined in section 2 (a)), a tax determined in accordance with the following table: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">If taxable income is:</th> <th style="text-align: left;">The tax is:</th> </tr> </thead> <tbody> <tr> <td>Not over \$36,900</td> <td>15% of taxable income.</td> </tr> <tr> <td>Over \$36,900 but not over \$89,150</td> <td>\$5,535, plus 28% of the excess over \$36,900.</td> </tr> <tr> <td>Over \$89,150 but not over \$140,000</td> <td>\$20,165, plus 31% of the excess over \$89,150.</td> </tr> <tr> <td>Over \$140,000 but not over \$250,000</td> <td>\$35,928.50, plus 36% of the excess over \$140,000.</td> </tr> <tr> <td>Over \$250,000</td> <td>\$75,528.50, plus 39.6% of the excess over \$250,000.</td> </tr> </tbody> </table>	If taxable income is:	The tax is:	Not over \$36,900	15% of taxable income.	Over \$36,900 but not over \$89,150	\$5,535, plus 28% of the excess over \$36,900.	Over \$89,150 but not over \$140,000	\$20,165, plus 31% of the excess over \$89,150.	Over \$140,000 but not over \$250,000	\$35,928.50, plus 36% of the excess over \$140,000.	Over \$250,000	\$75,528.50, plus 39.6% of the excess over \$250,000.
If taxable income is:	The tax is:													
Not over \$36,900	15% of taxable income.													
Over \$36,900 but not over \$89,150	\$5,535, plus 28% of the excess over \$36,900.													
Over \$89,150 but not over \$140,000	\$20,165, plus 31% of the excess over \$89,150.													
Over \$140,000 but not over \$250,000	\$35,928.50, plus 36% of the excess over \$140,000.													
Over \$250,000	\$75,528.50, plus 39.6% of the excess over \$250,000.													
170	Charitable, etc., contributions and gifts	(a) Allowance of deduction (1) General rule There shall be allowed as a deduction any charitable contribution (as defined in subsection (c)) payment of which is made within the taxable year. A charitable contribution shall be allowable as a deduction only if verified under regulations prescribed by the Secretary.												
401	Qualified pension, profit-sharing and stock bonus plans	(a) Requirements for qualification A trust created or organized in the United States and forming part of a stock bonus, pension, or profit-sharing plan of an employer for the exclusive benefit of his employees or their beneficiaries shall constitute a qualified												

LAW IS CODE

		trust under this section— (1) if contributions are made to the trust by such employer, or employees, or both, or by another employer who is entitled to deduct his contributions under section 404 (a)(3)(B) (relating to deduction for contributions to profit-sharing and stock bonus plans), or by a charitable remainder trust pursuant to a qualified gratuitous transfer (as defined in section 664 (g)(1)), for the purpose of distributing to such employees or their beneficiaries the corpus and income of the fund accumulated by the trust in accordance with such plan. . .
--	--	---

C. Comparing Titles 12 and 26 to Other Titles

As seen in our visualizations above, both Title 26 and Title 12 have very large cores, implying great complexity. A natural question is whether this characteristic is common to all titles of the U.S. Code. As elaborated more fully in Appendix D, this is not the case. In fact, Titles 12 and 26 have two of the largest cores in the U.S. Code. The top 5 titles with the largest cores are given in Table 12. The average core size of a U.S. Code title is 89.81, much lower than the size of the cores of Titles 12 and 26. In Appendix D, we show visualizations of the cores of all U.S. Code titles, illustrating how rare it is to have a very large and dense core. Thus, our techniques seem to be useful indicators of complexity for a given title.

Table 12: U.S. Code Titles with Largest Cores

Title	Core Size (Number of Sections)
26 (Internal Revenue Code)	1037
42 (Public Health and Welfare)	873
12 (Banks and Banking)	279
20 (Education)	234
49 (Transportation)	200

VII. CONCLUSION

The similarities between software and law is striking—in many respects, law *is* code. When viewed from a software engineering perspective, the U.S. Code resembles a large software system, and the application of software design principles allows us to quantify the extent to which the law is concise, changing, coupled, and complex. Our methods reveal the rise, spread, and fall of legal terms used in the U.S. Code,

the structure of the cross-references network, and the types of laws that Congress enacts. When applied to specific titles, these methods have identified particularly complex and highly interconnected sections, which should be prime candidates for regulatory reform and simplification. The sheer size and number of cross references within the core sections imply that software-engineering methods can play an important role in leveraging human ability. Therefore, a software engineering approach to measuring and managing the U.S. Code allows lawmakers to enact better legislation with fewer vulnerabilities.

Creating less complex laws and simplifying the existing legal code also reduce the number of unintended consequences and ensure more fair and equitable outcomes for all stakeholders. By developing a more coherent and systematic view of the entire body of laws governing our society, we create more informed participants in the legal system, empowering lawyers, judges, and individual citizens in their respective roles of proposing, enforcing, interpreting, and changing the law. One cannot manage what one does not measure, and as the U.S. Code becomes larger and more unwieldy, software-engineering methods can greatly enhance our ability to participate in the legislative process.

APPENDIX A: MATHEMATICAL DEFINITIONS

The purpose of this appendix is to give formal definitions of the terms used in our network analysis of the U.S. Code. Because of this, it is heavy in mathematics and is intended for the interested reader. Each definition includes a description of how it is relevant to the U.S. Code and this to Article.

Networks. A network $N = (V, E)$ is given by a set of vertices V (also called nodes) and a set of edges $E \subset V \times V$. We say that there is an edge from u to v (and write $u \rightarrow v$) if the pair (u, v) belongs to the set E . In our application, vertices are sections of the U.S. Code, and edges correspond to citations between sections. There is an edge $u \rightarrow v$ if and only if u cites v .

Reachability. Given two vertices u, v in a network, we say that v is reachable from u if there exists a set of vertices $x_1, \dots, x_n \in V$ such that $u \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n \rightarrow v$. In our application, this would imply that there is a chain of citations going from u to v . Thus, any change to section v of the U.S. Code would indirectly affect section u . Throughout this section, if v is reachable by u , we say that u indirectly cites v .

Strong Connectedness. Given two vertices u, v in a network, we say that u is strongly connected to v if:

- (1) v is reachable from u and
- (2) u is reachable from v .

In this work, two sections of the U.S. Code are strongly connected if there is a path of citations via which u affects v and there is another path of citations from v to u . The simplest way in which u, v can be strongly connected is if they both cite each other. It is immediate that, not only will changes to v affect u (because u cites v), but they can also affect v itself by following a loop of citations.

Strong Connectedness as an Equivalence Relation. Strong connectedness induces an equivalence relation on the set of vertices. That is, it satisfies:

- **Reflexivity:** For any vertex v , v is strongly connected to itself
- **Symmetry:** For any u, v we have that u is strongly connected to v if and only if v is strongly connected to u
- **Transitivity:** For any u, v, w , we have that if u is strongly connected to v , and v is strongly connected to w , then u is strongly connected to w

For any vertex $v \in V$ define the strongly connected component containing v as $C(v) = \{u : u \text{ is strongly connected to } v\}$. Note that, because strong connectedness is an equivalence relation, for every u strongly connected to v we have $C(u) = C(v)$. Thus, we can partition the set of vertices V into disjoint equivalence classes

$$V = V_1 \cup \dots \cup V_n,$$

where all the elements in a given equivalence class V_i are strongly connected to each other, but for any $i \neq j$ the elements in V_i, V_j are not strongly connected. Each V_i is called a strongly connected component of the network. In our legal application, the set V_i are sets of sections of the U.S. Code, which all indirectly cite each other. Thus, these sets can be interpreted as a *modular decomposition* of the U.S. Code, with each V_i representing a module.

Core of a Network. Given a network $N = (V, E)$, and a corresponding decomposition into strongly connected components $V = V_1 \cup \dots \cup V_n$, the *core* is the largest strongly connected component: $\text{Core}(N) = \text{argmax}_{V_i} |V_i|$.

In our legal application, the core of the U.S. Code is a subset of sections of the U.S. Code that satisfies the following two properties:

- (1) All sections in the core indirectly cite each other, and
- (2) The core is the largest set satisfying property (1)

Thus, the core can be seen as the largest “module” of the U.S. Code. Changing any section in the U.S. Code will, by definition, affect a large number of other sections that indirectly cite it, and is a possible way of introducing contradictions in the law, since each section in the core belongs to a large “citation loop.”

Measuring the Coupling of a Law. When a law passed by Congress gets codified, different sections of the law become incorporated into different sections of the U.S. Code. Thus, we can interpret a given law as a subset $S \subset V$ of the sections of the U.S. Code that it is modifying. At first approximation, a law that has multiple sections in the core of the U.S. Code will indirectly affect the operation of many other laws, while a law that does not modify the core of the U.S. Code will not have such a high impact. Thus, we can approximately model how “central” a given law is by:

- (1) Finding the set S of sections of the U.S. Code modified by the law, and
- (2) Computing the size of the intersection of S with the core of the U.S. Code.

That is, given a law that modifies a set S of sections in the code, we have that its coupling metric is given by $\text{coupling}(S) = |S \cap \text{Core}|$. Note that we identify the law with the set of sections of the U.S. Code it modifies. We can do this by using the Table III Tool provided by the Office of the Law Revision Counsel.¹⁹⁴

Subgraphs and the Core of a Given Law. Given a network $N = (V, E)$ and a subset $S \subset V$ of vertices, we can define the subgraph induced by S as $N' = (S, E'(S))$ where $E'(S) = \{(u, v) \in E : u, v \in S\}$. That is, N' only contains elements of S as vertices, and the edges are the edges of the original network N that connect nodes in S .

¹⁹⁴. Table III Tool, *supra* note 119.

This definition is useful for our work because the entirety of the U.S. Code is a very large network with tens of thousands of nodes. In our work, we also find it useful to focus on individual laws. As mentioned in the above paragraph, we identify a law with the subset $S \subset V$ of sections of the U.S. Code that it modifies. This induces a subgraph $N'(S)$ which contains only the sections of the U.S. Code modified by the given law. This subgraph is frequently much smaller, with only hundreds of nodes. We define the **core of a given law** as the core of the induced subgraph $N'(S)$. A law with a large core can be interpreted as more complex and more non-linear than a law with a small core, since changing one section of the law is likely to have indirect effects on a large number of other sections.

Analogously, we can define the core of a **title** of the U.S. Code as the core of the subgraph induced by all sections in that title.

PageRank. While analyzing the core of a law or title of the U.S. Code can help us understand the degree of coupling in said law or title, this type of analysis cannot be used to rank the complexity of individual sections. Because the core of a network is an equivalence class, all sections in the core are equally complex. In order to provide a ranking by complexity of sections in a piece of legislation, we use an algorithm called PageRank, which is one of the main backbones behind search engine algorithms.

We first give an informal definition of the PageRank procedure to give a general intuition. Afterwards, we give a formal definition for readers with a background in linear algebra. Informally, the PageRank procedure seeks to answer the following question: if a reader followed citation links in the U.S. Code randomly, following a random citation every time they reached a new section, what is the probability that they would end up in any given section? Intuitively, sections that have a high probability of being visited by such a “random walk” are sections that are highly central, and which have a high indirect impact on many sections of the U.S. Code.

The PageRank algorithm uses the network’s *modified transition matrix*, defined below, in order to quickly compute the probability that a given vertex will be visited by a random walk.

We now more formally give this algorithm, for readers with a background in linear algebra and algorithms.¹⁹⁵ Let $N = (V, E)$ be a network with n vertices. Label the vertices of this network with the numbers 1 through n . Given two indices i, j representing vertices in V , define

$$a_{ij} = \begin{cases} 1 & \text{if } (j, i) \in E \text{ or if } j \text{ has no outgoing edges} \\ 0 & \text{otherwise} \end{cases}$$

195. Brian White, *Math 51 Lecture Notes: How Google Ranks Web Pages*, DEP’T OF MATHEMATICS STAN. UNIV. (Nov. 2004), http://math.stanford.edu/~brumfiel/math_51-06/PageRank.pdf.

$$n_j = \sum_i a_{ij}$$

$$p_{ij} = \frac{a_{ij}}{n_j}$$

Define the following matrices

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

$$T = \begin{pmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix}$$

Let $Q = \alpha P + (1-\alpha)T$, where α is a parameter of our choice when running the algorithm. In our work, we use $\alpha = 0.85$. One can prove that there exists a unique vector $x \in \mathbb{R}^n$ such that

- (1) $x = Qx$
- (2) $x_i \geq 0$
- (3) $\sum_i x_i = 1$

Intuitively, x is the steady state of a random walk on our given network, where the random walk resets itself with probability $1 - \alpha$. This steady state is given by the eigenvector of Q with eigenvalue 1.

LAW IS CODE

APPENDIX B: CORES OF APPROPRIATIONS BILLS

In this appendix, we show the cores of all appropriations bills passed since 1994. This visualization confirms our intuition that appropriation bills are very simple.

The definition of the core of a given law is given in Appendix A.

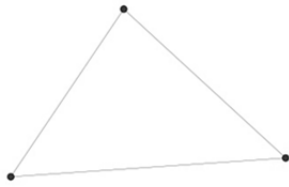


Figure 18: Agriculture, Rural Development, Food and Drug Administration, and Related Agencies Appropriations Act, 1997¹⁹⁶



Figure 19: Continuing Appropriations Act, 2011¹⁹⁷



Figure 20: Continuing Appropriations Resolution, 2013¹⁹⁸



Figure 21: Department of Defense Appropriations Act, 1999¹⁹⁹



Figure 22: Department of Defense Appropriations Act, 2001²⁰⁰

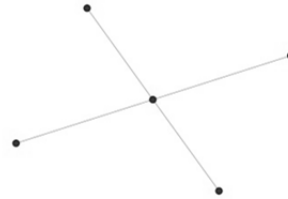


Figure 23: Department of Defense Appropriations Act, 2003²⁰¹

196. Agriculture, Rural Development, Food and Drug Administration, and Related Agencies Appropriations Act, 1997, Pub. L. No. 104-180, 110 Stat. 1569 (1996).

197. Continuing Appropriations Act, 2011, Pub. L. No. 111-242, 124 Stat. 2607 (2010).

198. Continuing Appropriations Resolution, 2013, Pub. L. No. 112-175, 126 Stat. 1313 (2012).

199. Department of Defense Appropriations Act, 1999, Pub. L. No. 105-262, 112 Stat. 2279 (1998).

200. Department of Defense Appropriations Act, 2001, Pub. L. No. 106-259, 114 Stat. 656 (2000).

201. Department of Defense Appropriations Act, 2003, Pub. L. No. 107-248, 116 Stat. 1519 (2002).

LAW IS CODE




Figure 24: Department of Defense Appropriations Act, 2005²⁰²




Figure 25: Department of Defense Appropriations Act, 2007²⁰³




Figure 26: Department of Defense Appropriations Act, 2008²⁰⁴

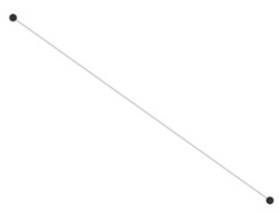


Figure 27: Department of Defense Appropriations Act, 2010²⁰⁵




Figure 28: Department of Homeland Security Appropriations Act, 2005²⁰⁶

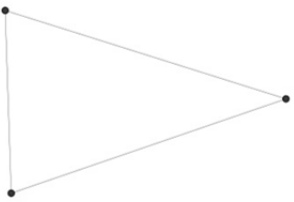
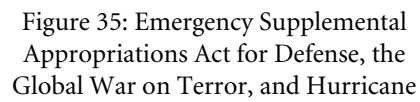
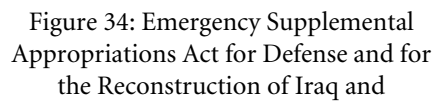
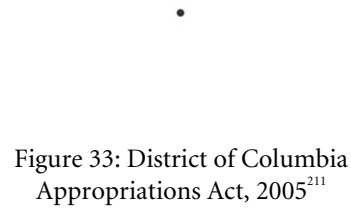
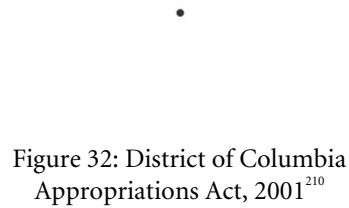
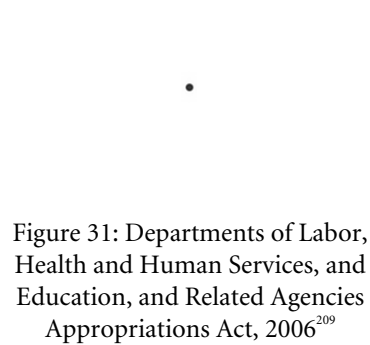
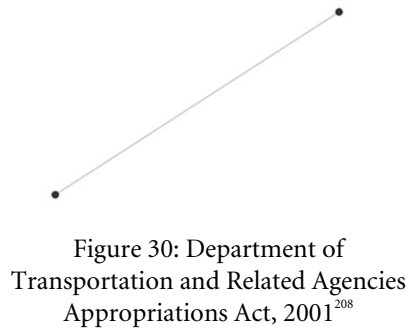


Figure 29: Department of Transportation and Related Agencies Appropriations Act, 1997²⁰⁷

-
202. Department of Defense Appropriations Act, 2005, Pub. L. No. 108-287, 118 Stat. 951 (2004).
203. Department of Defense Appropriations Act, 2007, Pub. L. No. 109-289, 120 Stat. 1257 (2006).
204. Department of Defense Appropriations Act, 2008, Pub. L. No. 110-116, 121 Stat. 1295 (2007).
205. Department of Defense Appropriations Act, 2010, Pub. L. No. 111-118, 123 Stat. 3409 (2009).
206. Department of Homeland Security Appropriations Act, 2005, Pub. L. No. 108-334, 118 Stat. 1298 (2004).
207. Department of Transportation and Related Agencies Appropriations Act, 1997, Pub. L. No. 104-205, 110 Stat. 2951 (1996).



208. Department of Transportation and Related Agencies Appropriations Act, 2001, Pub. L. No. 106-346, 114 Stat. 1356 (2000).

209. Departments of Labor, Health and Human Services, and Education, and Related Agencies Appropriations Act, 2006, Pub. L. No. 109-149, 119 Stat. 2833 (2005).

210. District of Columbia Appropriations Act, 2001, Pub. L. No. 106-522, 114 Stat. 2440 (2000).

211. District of Columbia Appropriations Act, 2005, Pub. L. No. 108-335, 118 Stat. 1322 (2004).

LAW IS CODE

Afghanistan, 2004²¹²

Recovery, 2006²¹³

212. Emergency Supplemental Appropriations Act for Defense and for the Reconstruction of Iraq and Afghanistan, 2004, Pub. L. No. 108-106, 117 Stat. 1209 (2003).

213. Emergency Supplemental Appropriations Act for Defense, the Global War on Terror, and Hurricane Recovery, 2006, Pub. L. No. 109-234, 120 Stat. 418 (2006).

Figure 36: Energy and Water
Development Appropriations Act,
1997²¹⁴

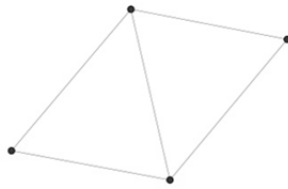


Figure 38: Energy and Water
Development Appropriations Act,
2001²¹⁶

Figure 37: Energy and Water
Development Appropriations Act,
1999²¹⁵

Figure 39: Energy and Water
Development Appropriations Act,
2004²¹⁷

Figure 40: Energy and Water
Development Appropriations Act,

Figure 41: Foreign Operations, Export
Financing, and Related Programs

214. Energy and Water Development Appropriations Act, 1997, Pub. L. No. 104-206, 110 Stat. 2984 (1996).
215. Energy and Water Development Appropriations Act, 1999, Pub. L. No. 105-245, 112 Stat. 1838 (1998).
216. Energy and Water Development Appropriations Act, 2001, Pub. L. No. 106-377, 114 Stat. 1441 (2000).
217. Energy and Water Development Appropriations Act, 2004, Pub. L. No. 108-137, 117 Stat. 1827 (2003).

LAW IS CODE

2006²¹⁸

Appropriations Act, 1998²¹⁹

218. Energy and Water Development Appropriations Act, 2006, Pub. L. No. 109-103, 119 Stat. 2247 (2005).
219. Foreign Operations, Export Financing, and Related Programs Appropriations Act, 1998, Pub. L. No. 105-118, 111 Stat. 2386 (1997).

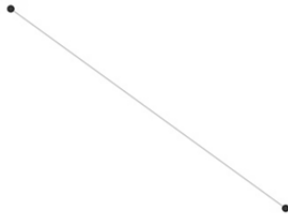


Figure 42: Foreign Operations, Export Financing, and Related Programs Appropriations Act, 2006²²⁰



Figure 43: Legislative Branch Appropriations Act, 1997²²¹



Figure 44: Legislative Branch Appropriations Act, 1999²²²



Figure 45: Military Construction and Veterans Affairs and Related Agencies Appropriations Act, 2009²²³



Figure 46: Military Construction Appropriations Act, 1997²²⁴

Figure 47: Military Construction Appropriations Act, 1999²²⁵

220. Foreign Operations, Export Financing, and Related Programs Appropriations Act, 2006, Pub. L. No. 109-102, 119 Stat. 2172 (2005).

221. Legislative Branch Appropriations Act, 1997, Pub. L. No. 104-197, 110 Stat. 2394 (1996).

222. Legislative Branch Appropriations Act, 1999, Pub. L. No. 105-275, 112 Stat. 2430 (1998).

223. Military Construction and Veterans Affairs and Related Agencies Appropriations Act, 2009, Pub. L. No. 110-329, 122 Stat. 3574 (2008).

224. Military Construction Appropriations Act, 1997, Pub. L. No. 104-196, 110 Stat. 2385 (1996).

225. Military Construction Appropriations Act, 1999, Pub. L. No. 105-237, 112 Stat. 1554 (1998).

LAW IS CODE

Figure 48: Military Construction Appropriations Act, 2001²²⁶



Figure 50: Military Construction Appropriations and Emergency Hurricane Supplemental Appropriations Act, 2005²²⁸



Figure 52: Science, State, Justice, Commerce, and Related Agencies

Figure 49: Military Construction Appropriations Act, 2004²²⁷



Figure 51: Military Construction, Military Quality of Life and Veterans Affairs Appropriations Act, 2006²²⁹



Figure 53: Supplemental

226. Military Construction Appropriations Act, 2001, Pub. L. No. 106-246, 114 Stat. 512 (2000).
227. Military Construction Appropriations Act, 2004, Pub. L. No. 108-132, 117 Stat. 1374 (2003).
228. Military Construction Appropriations and Emergency Hurricane Supplemental Appropriations Act, 2005, Pub. L. No. 108-324, 118 Stat. 1220 (2004).
229. Military Quality of Life and Veterans Affairs Appropriations Act, 2006, Pub. L. No. 109-114, 119 Stat. 237 (2005).

WILLIAM LI ET AL.

Appropriations Act, 2006²³⁰

Appropriations Act, 2008²³¹

230. Science, State, Justice, Commerce, and Related Agencies Appropriations Act, 2006, Pub. L. No. 109-108, 119 Stat. 2290 (2005).

231. Supplemental Appropriations Act, 2008, Pub. L. No. 110-252, div. A, 122 Stat. 2323 (2008).

LAW IS CODE



Figure 54: Supplemental Appropriations Act, 2010²³²

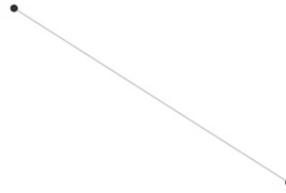


Figure 55: Transportation, Housing and Urban Development, and Related Agencies Appropriations Act, 2010²³³

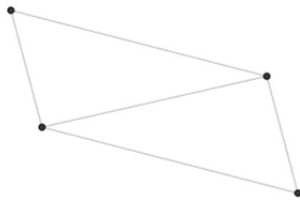


Figure 56: Transportation, Treasury, and Independent Agencies Appropriations Act, 2004²³⁴



Figure 57: Transportation, Treasury, Housing and Urban Development, the Judiciary, the District of Columbia, and Independent Agencies Appropriations Act, 2006²³⁵

-
232. Supplemental Appropriations Act, 2010, Pub. L. No. 111-212, div. F, 124 Stat. 2302 (2010).
233. Consolidated Appropriations Act, 2010, Pub. L. No. 111-117, 123 Stat. 3034, 3035-113 (2009).
234. Consolidated Appropriations Act, 2004, Pub. L. No. 108-199, div. D, 118 Stat. 3, 279-362 (2004).
235. Transportation, Treasury, Housing and Urban Development, the Judiciary, the District of Columbia, and Independent Agencies Appropriations Act, 2006, Pub. L. No. 109-115, 119 Stat. 2396 (2005).

WILLIAM LI ET AL.

APPENDIX C: BILLS WITH LARGE CORES

In this appendix, we show the cores of all laws passed since 1994 that have a core of size larger than 50. This includes many well-known complex laws, including, for example, the Wall Street Transparency and Accountability Act of 2010. The purpose of this appendix is to illustrate instances of bills that differ from the simple core structure of appropriations bills.

The definition of the core of a given law is given in Appendix A.

LAW IS CODE

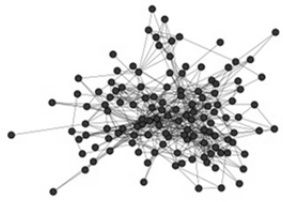


Figure 58: Fair and Equitable Tobacco Reform Act of 2004²³⁶

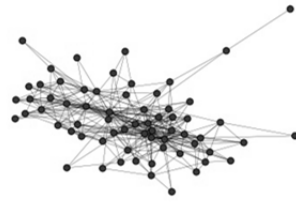


Figure 59: Pension Protection Act of 2006²³⁷



Figure 60: Personal Responsibility and Work Opportunity Reconciliation Act of 1996²³⁸

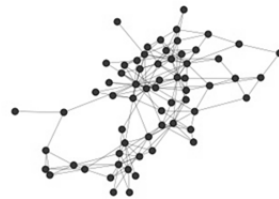


Figure 61: Private Student Loan Transparency and Improvement Act of 2008²³⁹



Figure 62: Small Business Job Protection Act of 1996²⁴⁰



Figure 63: Small Public Housing Authorities Paperwork Reduction Act²⁴¹

236. Fair and Equitable Tobacco Reform Act of 2004, Pub. L. No. 108-357, 118 Stat. 1521 (2004).

237. Pension Protection Act of 2006, Pub. L. No. 109-280, 120 Stat. 780 (2006).

238. Personal Responsibility and Work Opportunity Reconciliation Act of 1996, Pub. L. No. 104-193, 110 Stat. 2105 (1996).

239. Private Student Loan Transparency and Improvement Act of 2008, Pub. L. No. 110-315, 122 Stat. 3478 (2008).

240. Small Business Job Protection Act of 1996, Pub. L. No. 104-188, 110 Stat. 1755 (1996).

241. Small Public Housing Authorities Paperwork Reduction Act, Pub. L. No. 110-289, 122 Stat. 2863 (2008).

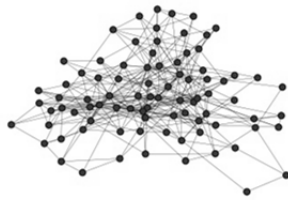


Figure 64: Tax Technical Corrections Act of 2005²⁴²



Figure 65: Tax Technical Corrections Act of 2007²⁴³



Figure 66: TEA 21 Restoration Act²⁴⁴



Figure 67: Transportation Research and Innovative Technology Act of 2012²⁴⁵



Figure 68: Veterans Benefits Act of 1998²⁴⁶



Figure 69: Wall Street Transparency and Accountability Act of 2010²⁴⁷

242. Tax Technical Corrections Act of 2005, Pub. L. No. 109-135, 119 Stat. 2610 (2005).

243. Tax Technical Corrections Act of 2007, Pub. L. No. 110-172, 121 Stat. 2473 (2007).

244. TEA 21 Restoration Act, Pub. L. No. 105-206, 112 Stat. 834 (1998).

245. Transportation Research and Innovative Technology Act of 2012, Pub. L. No. 112-141, 126 Stat. 405, 864-905 (2012).

246. Veterans Benefits Act of 1998, Pub. L. No. 105-178, 112 Stat. 492 (1998).

247. Wall Street Transparency and Accountability Act of 2010, Pub. L. No. 111-203, 124 Stat. 1641 (2010).

LAW IS CODE



Figure 70: Web-Based Education Commission Act²⁴⁸



Figure 71: Workforce Investment Act of 1998²⁴⁹

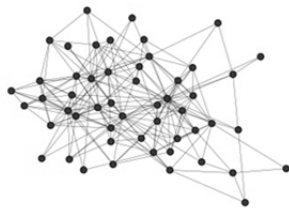


Figure 72: Working Families Tax Relief Act of 2004²⁵⁰

248. Web-Based Education Commission Act, Pub. L. No. 105-244, 112 Stat. 1822 (1998).

249. Workforce Investment Act of 1998, Pub. L. No. 105-220, 112 Stat. 936 (1998).

250. Working Families Tax Relief Act of 2004, Pub. L. No. 108-311, 118 Stat. 1166 (2004).

APPENDIX D: CORES OF TITLES OF THE U.S. CODE

In this appendix, we show the cores of all the titles in the U.S. Code. This visualization confirms our intuition that some titles, such as Titles 13 and 14, are relatively simple while other titles, such as Titles 12, 26 and 42, are highly complex.

The definition of the core of a title is given in Appendix A.



Figure 73: Core of Title 1



Figure 74: Core of Title 2

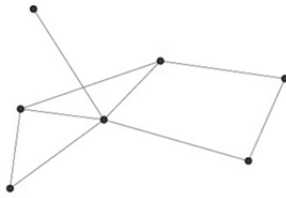


Figure 75: Core of Title 3

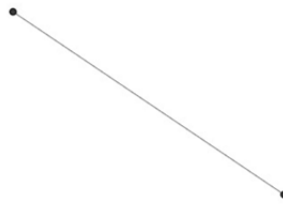


Figure 76: Core of Title 4



Figure 77: Core of Title 5

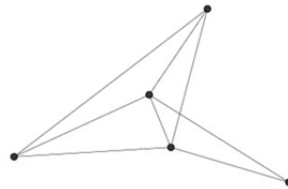


Figure 78: Core of Title 6



Figure 79: Core of Title 7



Figure 80: Core of Title 8



Figure 81: Core of Title 9



Figure 82: Core of Title 10



Figure 83: Core of Title 11



Figure 84: Core of Title 12

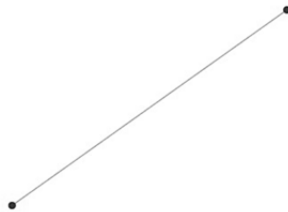


Figure 85: Core of Title 13

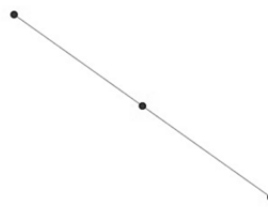


Figure 86: Core of Title 14



Figure 87: Core of Title 15

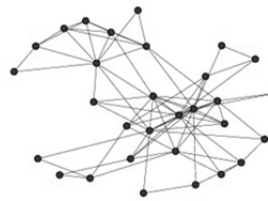


Figure 88: Core of Title 16

LAW IS CODE

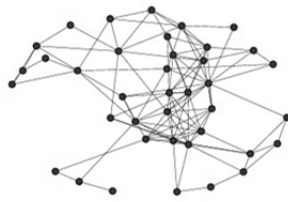


Figure 89: Core of Title 17



Figure 90: Core of Title 18



Figure 91: Core of Title 19



Figure 92: Core of Title 20



Figure 93: Core of Title 21



Figure 94: Core of Title 22

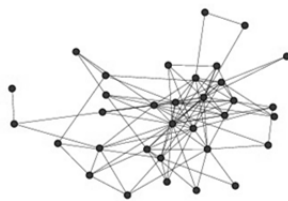


Figure 95: Core of Title 23

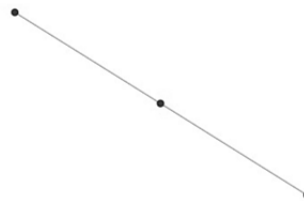


Figure 96: Core of Title 24

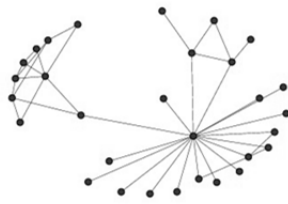


Figure 97: Core of Title 25



Figure 98: Core of Title 26



Figure 99: Core of Title 27



Figure 100: Core of Title 28



Figure 101: Core of Title 29

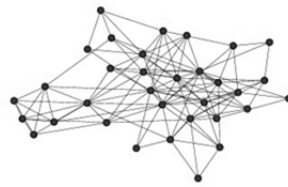


Figure 102: Core of Title 30

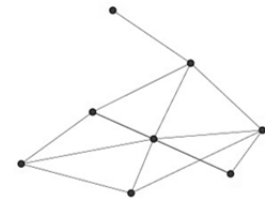


Figure 103: Core of Title 31



Figure 104: Core of Title 32



Figure 105: Core of Title 33



Figure 106: Core of Title 34



Figure 107: Core of Title 35

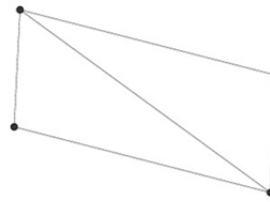


Figure 108: Core of Title 36

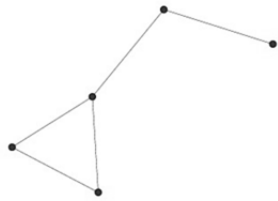


Figure 109: Core of Title 37



Figure 110: Core of Title 38

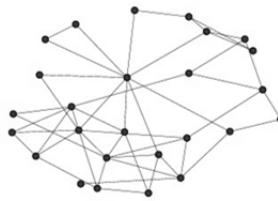


Figure 111: Core of Title 39

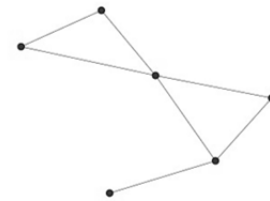


Figure 112: Core of Title 40



Figure 113: Core of Title 41

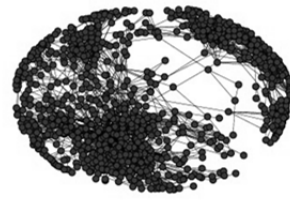


Figure 114: Core of Title 42



Figure 115: Core of Title 43

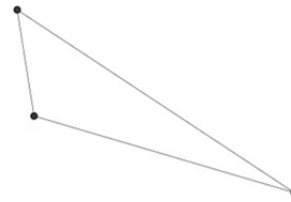


Figure 116: Core of Title 44

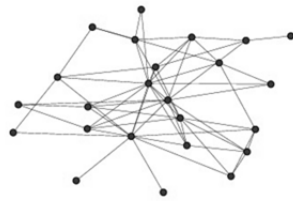


Figure 117: Core of Title 45



Figure 118: Core of Title 46



Figure 119: Core of Title 47

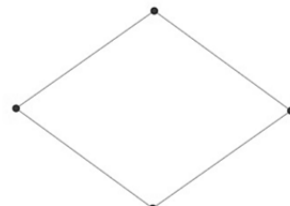


Figure 120: Core of Title 48

LAW IS CODE

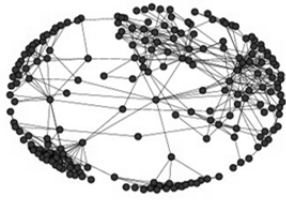


Figure 121: Core of Title 49



Figure 122: Core of Title 50

Figure 123: Core of Title 51