

Text as Policy:

Measuring Policy Similarity through Bill Text Reuse

Fridolin Linder, Bruce Desmarais, Matthew Burgess, Eugenia Giraudy*

02/07/2017

Abstract

The identification of substantively similar policy proposals in both proposed and adopted legislation is important to scholars of public policy diffusion and legislative politics. Conventional, manual, approaches are prohibitively costly in constructing datasets that accurately represent policymaking across policy domains, jurisdictions, or time. We propose the use of text-sequencing algorithms, applied to legislative text, to identify bills that introduce similar policy proposals. We present three ground truth tests, applied to a corpus of 500,000 bills from US-state legislatures. First, we show that bills introduced by ideologically similar sponsors are more likely to exhibit a high degree of text reuse. Second, we show that bills classified by the National Council of State Legislatures as covering the same policies exhibit a high degree of text re-use. Third, we show that rates of text reuse across state borders correlate with the diffusion networks recently introduced by Desmarais, Harden and Boehmke (2015).

*This work was supported by National Science Foundation grants SES-1558661, SES-1619644, CISE-1320219 and DGE-1144860. Any opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect those of the sponsor.

1 Introduction

The diffusion of public policy has been a central focus in political science research since at least [Walker \(1969\)](#). Research on public policy adoption and diffusion has conventionally relied on modestly-sized, hand-coded datasets that record when a set of jurisdictions adopt one or a handful of similar policies ([Boehmke and Skinner, 2012](#)). In this paper we draw upon a voluminous and comprehensive source of data with which to measure the consideration and adoption of similar policies – the text of legislation considered in US state legislatures. We take an automated, text-as-data, approach to the analysis of bill text. This vastly increases the volume of data that can be included in policy adoption research.

Through the development of an automated text analytic approach to measuring policy similarity, we consider several methodological challenges and broader conceptual hurdles. The central methodological puzzles are two-fold. First, we need to extract segments of text in pieces of legislation that communicate similar policy enactments. Second, we need to develop a method for quantitatively scoring the reused text between bills. The main conceptual puzzle with which we engage is the potential ambiguity regarding whether two policies should be considered the same for the purposes of public policy policy research. Considering precedents in the literature, we argue for conceptualizing policy adoption in terms of a continuum of similarity rather than a dichotomy of equivalence.

We take a multi-pronged approach to evaluating the performance of text reuse in measuring the consideration and adoption of similar laws. First, we investigate the relationship between the ideological distance of the legislators that proposed two bills and the text reuse scores between these bills. Second, we assess the how well policy diffusion networks that have been established by previous research can be predicted using the amount of text reuse between states. And third, we use data on equivalent bills collected by the

National Council of State Legislatures to build an evaluation data set containing true policy overlap to assess the accuracy of text reuse in predicting policy overlap. We find that a high degree of text reuse between bills provides a precise signal of the presence of similar policies within the bills.

2 Background

The measurement task on which we are focused is the identification of comparable policy actions (e.g., adoption, consideration) across jurisdictions—US states in particular. Scholars of public policy, legislative politics, and related areas have used measurements of comparable policy actions in a variety of research tasks. These include the study of policy diffusion (Karch, 2007), the comparison of policies on a select legislative issue (e.g., Huber, Shipan and Pfahler, 2001; Mycoff, Wagner and Wilson, 2009), and influence/adoption of model legislation introduced by advocacy organizations (e.g., Garrett and Jansa, 2015; Burgess et al., 2016). In each of these areas of research, scholars make use of measurements of consistent policy actions across jurisdictions. Below we review a variety of methods, and conceptual definitions that researchers have used in identifying similar policy actions.

We first consider the levels at which measurement has been conducted in past work. Public policy research on cross-jurisdiction comparable policy actions has conventionally involved the manual identification of related policies across states, countries or local governments in one or a handful of policy areas (e.g., Walker, 1969; Berry and Berry, 1990; Simmons and Elkins, 2004; Gilardi, Füglistner and Luyet, 2009; Krause, 2011). This is a manual approach to measurement at the monadic (i.e., governing jurisdiction) level. Recognizing the limitations of monadic levels of measurement, researchers have developed dyadic approaches to studying the diffusion relationships between states (Volden, 2006; Boehmke, 2009). In the most recent iteration of dyadic approaches, Desmarais, Harden and Boehmke (2015) apply

network inference algorithms to US state adoption sequences in over 100 policy domains to empirically infer the underlying network through which policies diffuse. In another recent innovation in measurement for policy analysis, [Garrett and Jansa \(2015\)](#) analyze the text of US state legislation through pairwise comparison with model legislation introduced by interest groups to identify the influence of model legislation.

We now consider the coding rules used in past research to define sets of equivalent, or at least similar, policies. Perhaps the broadest approach—studying policies adopted within a domain, but moving the status quo in opposite directions—is represented by [Glick \(1992\)](#). Glick studies judicial enactments related to the “right to die”—the right of patients or their representatives to end the use of life-preserving medical technology. He considers an adoption to include any judicial enactment on this topic, whether they restrict medical providers’ deference to the patients or enact broad patients’ rights. [Berry and Berry \(1990\)](#) look at policy change in a uniform direction—states’ adoption of lotteries. Lotteries, of course, vary in terms of their rules and financial models, but all of the policies studied by [Berry and Berry \(1990\)](#) moved the status quo from no state lottery to the existence of a state lottery. [Volden \(2006\)](#) considers a broad array of state laws—those implemented for the Federal Health and Human Services Children’s Health Insurance Program (CHIP). CHIP laws were coded for six policy characteristics and then analyzed for diffusion dynamics. Lastly, [Mooney and Lee \(1995\)](#) provides an example of quantitative coding of policies adopted in a given domain. They code the permissiveness of state abortion laws (pre *Roe v. Wade*) on a five-point scale. These examples convey the variability with which scholars have defined the set of policies that are considered to be comparable across state borders.

3 Detecting Policy Similarity through Text Reuse

The above review of the ways in which researchers have defined comparable policies offers clarity regarding the implicit concepts underlying policy comparability. We see that policies that are considered comparable exhibit two qualities. First, they often, but do not always, move the status quo in a similar direction. This could be thought of as ideological similarity. Second, they enact policy in similar domains—either narrowly or broadly conceived. Before we describe the algorithm we use for detecting text re-use, and validate that algorithm, we draw upon these two concepts to define the latent variable we seek to measure through the assessment of text overlap. Specifically, we are motivated by the fact that researchers do not typically focus on strictly equivalent policies. Rather, they deem policies comparable if they are highly similar along one or two dimensions—policy domain and (optionally) ideological direction.

We see past work that has identified sets of comparable policies as focusing on sets of policies that meet some threshold of similarity. As such, the latent variable that we argue below is measured effectively through text re-use is *policy similarity*. The text in legislation is an aggregate representation of the dimensions underlying the policies proposed therein. These dimensions include the domain of the policy, the ideological position enacted by the policy, the level of specificity in the policy enactment, and several other salient features of policy that are communicated through the text in legislation. We do not claim that the re-use of text can be used to measure these dimensions in a specific, dis-aggregated way. Rather, text re-use serves as a summary measure of the greatest overlap observed across all relevant policy dimensions represented in legislative text. Furthermore, we note that by focusing on legislative text, we are measuring the similarity of the intended policies in terms of how those policies reflect in formal enacting legislation. Once enacted, the policies that are implemented in different states, under different bureaucratic and judicial regimes, will likely differ even if they are enacted under equivalent legislative language. This

divergence in policy implementation is attributable to the inevitable discretion exercised by bureaucratic officials in implementing policy (Weingast and Moran, 1983; Scott, 1997; Gerber, Maestas and Dometrius, 2005).

We should emphasize at this point that we are not the first contributors to the legislative politics, or even state legislative politics, literature to use text similarity between bills to measure policy similarity. Wilkerson, Smith and Stramp (2015) use text similarity—indeed, the same algorithm we use—to trace the development of very similar policy proposals in legislation produced in the US Congress. Garrett and Jansa (2015) and Burgess et al. (2016) both use measures of text similarity in US state legislation to identify bills that incorporate similar forms of model legislation, as provided by legislative/policy advocacy groups. Our contributions to this developing literature are two-fold. First, we broaden the scope of measurement beyond tracing single ideas through bill versions or detecting model legislation—to assessing policy similarity between bills in general. Second, we present multiple validation tests to assess whether our text reuse based measure of policy similarity correlates with benchmarks that we deduce from theories of state policymaking.

3.1 Detecting Text Reuse

To assess the maximal similarity of policies proposed in two pieces of legislation, we seek to identify large segments of equivalent or highly similar text (i.e., text reuse between bills). Alignment algorithms for discovering similar long sequences of text stand in contrast to bag-of-words methods, which are based on the comparison of word occurrence frequencies in documents. Bag-of-words methods may be effective for capturing broad topical areas in legislation, but will provide results that are too coarse for precisely assessing policy overlap. For example, in modeling bills in the US Congress using statistical topic models, arguably the canonical bag-of-words method for text analysis, Gerrish and Blei (2011) find that topic models for legislative text perform most effectively when parameterized

with 64 topics to model a corpus of 4,447 bills. While 64 topics provides a fairly detailed categorization of the domains covered in legislation, it is unrealistic to think that this method can isolate individual policy proposals. Considering [Gerrish and Blei’s \(2011\)](#) results again, that would mean that the US Congress considers nearly seventy bills, on average, for each specific policy proposed.

There are a variety of algorithms designed to automatically detect text reuse between documents mostly originating in the plagiarism detection literature (see e.g. [Potthast et al., 2013](#), for an overview). In political science, [Wilkerson, Smith and Stramp \(2015\)](#) introduced the Smith-Waterman local alignment algorithm (SW algorithm) to detect overlapping language in congressional bills in order to trace policy ideas through the legislative process of the US Congress. We use the same algorithm to detect text reuse in US state bills, however, our implementation of the algorithm differs slightly. The SW algorithm was developed by [Smith and Waterman \(1981\)](#) in molecular biology, in order to match genetic sequences. Given two sequences (of genes or words) it calculates the optimal local alignment (match) between these sequences.

Sequences in text that convey similar—even equivalent—policy content often do not match perfectly. Formatting, white space, typographical errors, etc. would decrease the recall dramatically if only exact matches were used. The SW algorithm, therefore, returns the optimal alignment while allowing for mismatches and gaps. The extent to which such imperfections are tolerated is governed by parameters that are set by the researcher. There are three such parameters: the match score (reward for exactly matching words), the mismatch score (penalty for including words that do not match), and the gap score (penalty for including placeholder words in matching a shorter sequence to a longer sequence). The goal of the algorithm is to return the alignment with the highest score given the input sequences and the parameters. Consider the following two excerpts from Alaska State Bill 203 (28th session) and North Carolina House Bill 366 (2015) respectively:

"section 1.AS44.99 is amended by adding new sections to read: article 6. compact for a balanced budget (...)"

"chapter 143 is amended by adding a new article to read: article 80. compact for a balanced budget (...)"

These sequences could be aligned in a large number of possible ways. There is a trade-off between short sequences that match perfectly, and longer sequences with imperfections. For example, we could match just "is amended by adding new" or "compact for a balanced budget". However, by adding a gap between "adding" and "new" in the first sequence and accepting the mismatches "article" - "section" and "6." - "80.", a longer alignment can be obtained: "is amended by adding (*a*) new (*article/section*) to read: article (*6./80.*)".

Because there are so many different possibilities, it is computationally demanding to find the optimal alignment. [Smith and Waterman \(1981\)](#) therefore proposed a dynamic programming approach that is illustrated in [Figure 1](#). Dynamic programming refers to algorithms that divide a problem into sub-problems and store results of previous calculations for later ones. In the case of the SW algorithm, the property that makes it a dynamic problem is the fact that all the alignments of shorter sub-sequences of an optimal alignment of a sequence are themselves optimal. Intuitively, if the benefits of including a sub-sequence in an alignment outweigh the penalties from mismatches and gaps in that sub-sequence, that sub-sequence will contribute positively to longer alignments to which it is appended. To make this clearer consider the beginning of the sequences in the example above:

"section 1.AS44.99 is amended"

"chapter 143 is amended"

It is obvious that the best local alignment of the first three elements would be just "is" - "is" since the first two elements don't match and therefore would just reduce the overall quality

of the alignment. If we now consider the alignment of the first four words we know that we can discard all possible alignments that involve the first two elements because the quality of every alignment that involves these elements could be improved by dropping them. Re-using the result from the first three elements, the problem of aligning the first four elements is now reduced to the single decision about the elements “amended” and “amended”. This decision is easy to make because we only have to consider four possibilities: We match the elements and they are actually the same, we match the elements and they are not the same, we introduce a gap in one of the sequences or we decide to stop the alignment.

If the next two elements match, the decision is easy - we add them and increase the total alignment score. However, if they do not match the decision is more difficult. We only want to introduce a gap or mismatch if it pays off later (that is if we thereby can ‘reach’ another part of the sequence that matches again). But we only know that once we went through the whole sequence. The central idea is therefore to find all optimal alignments ending in all possible spots, keep these scores and after everything is calculated finding the alignment that maximizes the overall quality. Therefore, after the decision is made the same procedure is applied to the next pair of sequence elements, and repeated until the end of both sequences is reached.

This intuition can be formalized in an algorithm that is based on a dynamic programming matrix that contains all optimal sub-alignments. A matrix is created where each combination of elements of the two sequences is assigned a cell. And an additional row and column of 0 is appended to the beginning of each sequence (to allow for a gap in the beginning of one sequence). The quality or alignment score of every possible alignment is then calculated cumulatively with the following algorithm. Denote the two sequences as $\mathcal{A} = (a_1, a_2, \dots, a_n)$ and $\mathcal{B} = (b_1, b_2, \dots, b_k)$. Additionally let δ , ϵ and γ be the match,

mismatch and gap scores. Define the scoring function:

$$S(a_i, b_j) = \delta \mathbb{I}(a_i=b_j) + \epsilon \mathbb{I}(a_i \neq b_j) \quad (1)$$

Where $\mathbb{I}(\cdot)$ is the indicator function. Then the entry for each cell $M_{i,j}$, $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$ of the matrix is filled by the following rule:

$$M_{i,j} = \max(M_{i-1,j-1} + S(a_i, b_j), M_{i-1,j} + \gamma, M_{i,j-1} + \gamma, 0) \quad (2)$$

The algorithm is visualized for the example above in Figure 1¹. Each row corresponds to a word in the sentence from the Alaska bill and each column to the sentence from the North Carolina bill. On the right side of the matrix the optimal alignment is displayed.

The four values of which the maximum is chosen in Equation 2 describe four steps in this dynamic programming array. The first value corresponds to a diagonal step which means in the resulting alignment sequence elements a_i and b_j are matched and the score of the alignment is either increased by δ if they match or decreased by ϵ if they don't match. The second and third value describe a gap step either in horizontal or vertical direction. That is, if we move from cell (i, j) to cell $(i + 1, j)$, in the resulting alignment sequence element a_{i+1} is matched with a gap. On the other hand, if we move to cell $(i, j + 1)$ sequence element b_{j+1} is matched with a gap in the resulting alignment. If the first three values are all ≤ 0 , the score in the cell is 0 meaning that no step is taken and every potential alignment stops at this point. This can be observed in Figure 1. The highlighted (blue) path describes the optimal alignment. The first two elements in both sequences don't match, therefore the score remains zero. Therefore the resulting alignment starts at the fourth element of each sequence with four diagonal steps, each adding points ($\delta = 5$) to

¹An interactive version can be found at <http://fridolin-linder.com/2016/03/30/local-alignment.html>.

	chapter	143	is	amended	by	adding	a	new	article	to	read:	article	80.	compact	for	a	balanced	budget	Alignment Result	
-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	is	is
section	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	amended	amended
1.AS44.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	by	by
is	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	adding	adding
amended	0	0	0	2	6	5	4	3	2	1	0	0	0	0	0	0	0	0	-	a
by	0	0	0	1	5	9	8	7	6	5	4	3	2	1	0	0	0	0	new	new
adding	0	0	0	0	4	8	12	11	10	9	8	7	6	5	4	3	2	1	sections	article
new	0	0	0	0	3	7	11	10	14	13	12	11	10	9	8	7	6	5	4	to
sections	0	0	0	0	2	6	10	9	13	12	11	10	9	8	7	6	5	4	3	read:
to	0	0	0	0	1	5	9	8	12	11	15	14	13	12	11	10	9	8	7	article
read:	0	0	0	0	0	4	8	7	11	10	14	18	17	16	15	14	13	12	11	6.
article	0	0	0	0	0	3	7	6	10	14	13	17	21	20	19	18	17	16	15	compact
6.	0	0	0	0	0	2	6	5	9	13	12	16	20	19	18	17	16	15	14	for
compact	0	0	0	0	0	1	5	4	8	12	11	15	19	18	22	21	20	19	18	a
for	0	0	0	0	0	0	4	3	7	11	10	14	18	17	21	25	24	23	22	balanced
a	0	0	0	0	0	0	3	7	6	10	9	13	17	16	20	24	28	27	26	budget
balanced	0	0	0	0	0	0	2	6	5	9	8	12	16	15	19	23	27	31	30	
budget	0	0	0	0	0	0	1	5	4	8	7	11	15	14	18	22	26	30	34	

Figure 1: Dynamic programming array of local alignment algorithm. The parameters in this instance are: Match Score: 5, Mismatch Score: -2, Gap Score: -1. The optimal alignment is displayed on the right side of the matrix. The left column corresponds to the text sequence in the rows of the matrix, the right column to the sequence in the columns. Mismatches are colored in red, gaps are indicated by "-" and orange color. The highlighted (blue) cells in the matrix display the path of the optimal alignment.

the final score which reaches 20 at the alignment ending in "adding"- "adding". After that, a step to the right is taken, meaning that the additional "a" in the column sequence is matched with a gap in the row sequence (indicated by "-" and orange coloring in the result on the right hand side). The cumulative score of the alignment is reduced by $\gamma = -1$ to 19. After that only diagonal steps are taken. Note that there are two mismatches each of which reduces the score by $\epsilon = -2$.

The optimal alignment is identified by finding the highest score in the matrix, at which the alignment ends, and back-tracing the path of the alignment. The steps in the path are chosen using Equation 2, and are indicated in Figure 1 by arrows pointing to the left, the upper left, up or no arrow. The back-tracing continues until a zero is encountered.

This algorithm is computationally intensive ($\mathcal{O}(nk)$ in computation time and memory demand), but returns the globally optimal local alignment given the scoring parameters. We provide a proof of this in the appendix. The property follows from the recursive relation used to fill the dynamic programming matrix. An alignment that is constructed by adding one element to an already optimal alignment is optimal itself. Starting by aligning the first two elements of the sequence optimally (which is trivial), a globally optimal can be found by iteratively adding to this alignment.

In our analyses we use a slightly modified version of the local alignment algorithm, in which the first gap in a series of multiple gaps receives a higher penalty than the following ones (Wilkerson, Smith and Stramp, 2015, use the same modification). This penalizes many small gaps and makes the algorithm produce longer gaps. The idea behind this modification is that someone who changes text in a bill might insert several words into an existing piece of text. The fact that something was inserted should weigh heavier than the length of the insertion which is achieved by down-weighting gap extensions.

4 Policy Similarity in US State Legislation

Through the application of the SW algorithm to US state legislation, we will provide a quantitative measure of policy similarity between bills. The similarity measures between bills that we identify can be aggregated up to the legislator or state levels. The measures we provide can be used to test hypotheses regarding, among other topics, public policy diffusion, legislative politics, political parties, and interest groups.

4.1 Data

We apply the SW algorithm to a database of US state bills, collected by [Burgess et al. \(2016\)](#) and the Sunlight Foundation.² This data base contains approximately 500,000 bills from 2008 to 2015³. Not all bills from all states are available for the whole time period. Figure 2 displays the year ranges and number of bills in the bill data base. This collection of bills is based on all bills that are available through the Sunlight Foundation’s [openstates.org](#) API. [openstates.org](#) is a website maintained by the Sunlight Foundation, in order to increase transparency in state politics. The Sunlight Foundation uses web scrapers to access all bills that are available on the websites of legislatures in all US states. This includes enacted legislation as well as bills that are still in the legislative process, or where not enacted.

Our approach of applying the SW algorithm is similar to [Wilkerson, Smith and Stramp \(2015\)](#) but differs in several key respects. The analysis consists of three major steps: 1) Pre-processing and selection of documents to compare, 2) alignment computation, 3) post hoc refinement (boiler plate removal).

4.2 Pre Processing and Selection

The SW algorithm is computationally demanding and an exhaustive comparison of all possible bill-pairs is not feasible (with 500,000 bills this would amount to about $1.24 * 10^{11}$ bill pairs). [Wilkerson, Smith and Stramp \(2015\)](#) encountered the same problem and excluded bills that don’t have at least five matching 10-grams. Since this implies considerable exact character matches, we see this as a too restrictive for our application. Because the text we are using originates in 50 different legislatures, exact character matches

²<http://sunlightfoundation.com/>

³Since computation is very demanding for this problem, the results reported in this version of the paper are based on a random sample of about half of the bills. We are in the process of completing the analyses for the whole dataset.

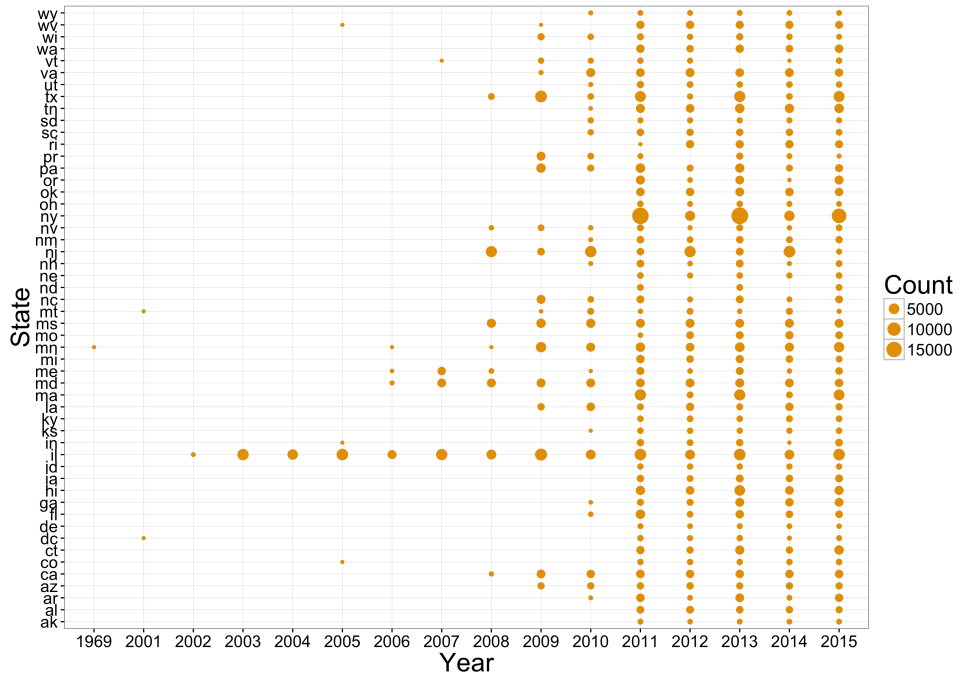


Figure 2: Description of the bill database. The horizontal bars display the time range for which bills are available for each state. The color of the bar indicates how many bills are available for this state.

might obfuscate valuable text reuse just because of small formatting changes, changes of state names, etc. Instead we choose to rely on a procedure that is based on finding bills with similar language. We store all available bills in an Elastic Search database. Elastic search is an open source web search engine which is designed to find documents that match a search query (Gormley and Tong, 2015). We utilize the ‘more like this’ query, which is designed to find documents that are similar to a given document. It works in the following way: From the focus document, the k n -grams with the highest tf-idf scores are selected and transformed into a search vector that is representative of the query document. Then, for each document in the collection, a variation of the cosine similarity is calculated and the m bills with the highest scores are selected for further analysis. For this analysis we set k to 25, n to 5 and m to 1000.⁴

⁴For details on the exact algorithm we refer the reader to the elastic search documentation (<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-mlt-query.html>) as well as the lucene engine documentation (https://lucene.apache.org/core/4_9_0/core/org/apache/)

4.3 Bill Alignment Computation

Once the candidate bills are selected, the alignments are calculated for all possible bill-dyads. Bill pairs from the same state are available but excluded for the analyses below. We exclude same-state comparisons in the current analysis since our validation exercises are focused on cross-state comparisons. We refer to the two bills involved in each alignment process as ‘left’ and ‘right’ bills. The algorithm returns parts of the left and right bills that are determined to be aligned as well as a score. We refer to these three pieces of information as an alignment.

[lucene/search/similarities/TFIDFSimilarity.html](#)). In order to account for the possibility of selection bias being introduced to our analyses through the cap of 1000 similar bills, we analyzed the results with different numbers of pre-selected bills. We find that this parameter, when set to a sufficiently high value, does not lead to such bias because the set of potentially relevant bills is always much smaller than 1000.

Table 1: Alignment Examples. Bold (red) text are mismatches. Dashes (orange, light text) are gaps introduced by the alignment algorithm. Non highlighted text are matches.

Left Text	Right Text	Score
<p>nj_214_A1167: “the entire credit may not be taken for the taxable year in which the renewable energy property is placed in service but must be taken in five equal - installments beginning with the taxable year in which the renewable energy property is placed in service. if, in one of the years in which the installment of a credit accrues, the renewable energy property with respect to which the credit was claimed is disposed of, taken out of service, or moved out of state, the credit expires and the taxpayer may not take any remaining installment of the credit. the taxpayer may, however, take the portion of an installment that accrued in a previous year and was carried forward to the extent permitted under”</p>	<p>nc_2011_SB747: ”the entire credit may not be taken for the taxable year in which the facility - - is placed in service but must be taken in five equal annual installments beginning with the taxable year in which the facility - - is placed in service. if, in one of the years in which the installment of a credit accrues, the facility - - with respect to which the credit was claimed is disposed - - - of - or taken out of service, the credit expires and the taxpayer may not take any remaining installment of the credit. the taxpayer may, however, take the portion of an installment that accrued in a previous year and was carried forward to the extent permitted under”</p>	388
<p>nj_214_A1167: “the entire credit may not be taken for the taxable year in which the renewable energy property is placed in service but must be taken in five equal installments - - - - beginning with the taxable year in which the”</p>	<p>ga_2011_12_HB146: “the entire credit may not be taken for the - year in which the - - property is placed in service but must be taken in four equal installments over four successive taxable years beginning with the taxable year in which the”</p>	110
<p>nj_214_A1167: “the entire credit may not be taken for the taxable year in which the renewable energy property is placed in service but must be taken in five equal installments beginning with the taxable year in which the”</p>	<p>nc_2009_SB305: “the entire credit may not be taken for the taxable year in which the costs are paid - - - but must be taken in five equal installments beginning with the taxable year in which the”</p>	108

Table 1 displays three examples of alignments. The ‘Left Text’ and ‘Right Text’ columns display the optimal alignment for the two sections the alignment score is displayed in the ‘Score’ column. This example shows the reuse of legislative text in three states (New Jersey, North Carolina and Georgia).

For our data we obtain from this procedure ca. $1.2 * 10^8$ individual alignments (that is, shared text between two bills). Figure 3 displays the cumulative frequency distribution of the alignments. The lowest score is 2.3, the highest 28811 with a mean alignment score of 18.8 (median 14.9). From this distribution it is clear that the majority of alignments is small with only relatively few bill pairs having large alignments. Of the 120 million bill-pairs about 800,000 have scores larger than 100 and ca. 44,000 have scores larger than 1000.

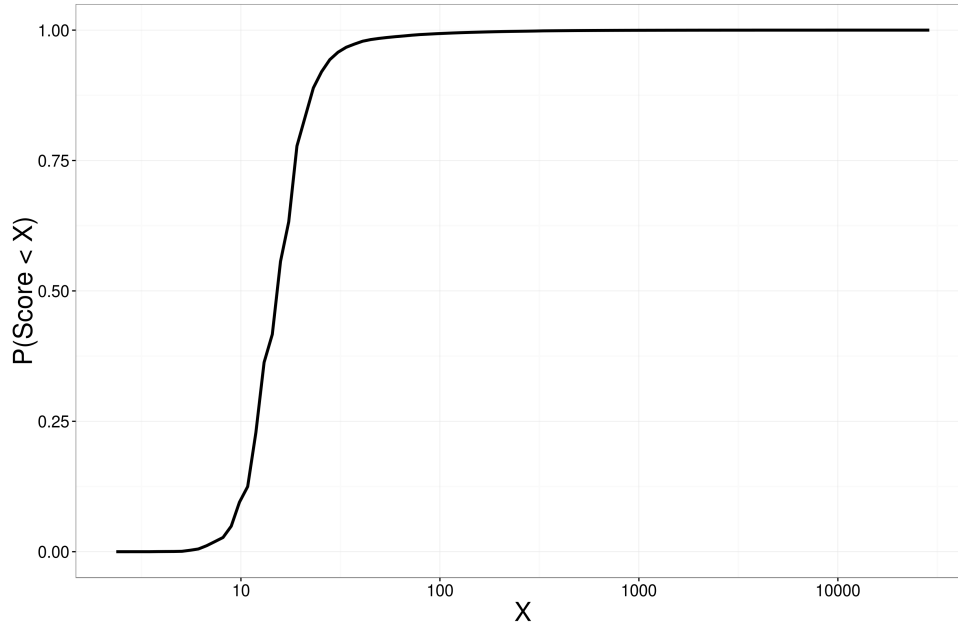


Figure 3: Cumulative frequency distribution of the dyad level alignment scores. The x-axis represents the dyad level alignment score on the \log_{10} scale.

In their application of the SW algorithm to the dataset that we use, Burgess et al. (2016) optimize the SW algorithm parameters through a predictive experiment. They collected 165 bills that were hand labeled by experts to reuse text from one or more bills within the

165 bill corpus. Looking at all 13,530 pairs of bills that can be formed from the corpus of 165, they used the alignment score to predict whether or not there was text reuse between the bills. They found the optimal parameters, in terms of area under the ROC curve, were a match score of 3, mismatch score of -2, and gap score of -3. We followed [Burgess et al. \(2016\)](#) in setting the parameters of the SW algorithm since they use the same data and have already optimized the parameter values. Future work should expand upon treatment of these parameter values in at least two directions. First, it would be valuable to develop a criterion function such that one does not need hand labeled data, which is costly and may exhibit its own biases, to select parameter values. Second, it would be valuable to develop an approach to representing uncertainty in the parameter values as well as in the optimal alignments and scores.

4.4 Approaches to Adjusting for Boiler Plate Text

Legislation is full of textual content that is not very useful in identifying the novel policy proposal(s) offered by the bill. Such text includes procedural language, headers, section titles, and common phrases such as definitions of legal terms, tax brackets, etc. Text that is irrelevant to specific policy proposals represents a major—perhaps the primary—limitation on the capacity for text reuse to identify the emulation of policies across bills. [Wilkerson, Smith and Stramp \(2015\)](#) use human coders to identify alignment text that is unrelated to policy content. They call this text “boiler plate.” Given a large set of hand-labeled alignments, they use a machine learning algorithm to classify text as either boiler plate or not based on the textual content of the alignment, and exclude alignments labeled as boiler plate. This represents one possible approach, but has two limitations. The first is that the use of human coders to label data to train the classifier is costly. The second is that the classification and exclusion approach treats legislative text as conforming to a procedural-or-not dichotomy, where language in a bill may be situated on a continuum in

terms of the degree to which it contributes to the policy proposal(s) offered in the bill. We discuss two alternative approaches to adjusting for boiler plate text in alignments.

The first approach also relies on an automatic classifier, but we propose a novel approach for generating training data without relying on human coders. We posit that boiler plate language is highly repetitive within states. Furthermore, language that is passed several times (i.e., in separate bills) within the same state is likely to be substantively irrelevant. Using these facts allows us to accumulate a large amount of labeled data—all alignments between bills that have both passed in the same state—which could be used to train a machine learning classifier to exclude boiler plate alignments. In addition to the text itself, the position in the text can be used as a feature in the classifier. This approach is very similar to that taken by [Wilkerson, Smith and Stramp \(2015\)](#), except that no human coders would be required to label boiler plate alignments.

In the second approach we describe, which we have implemented in the analyses we present below, we do not exclude boiler plate according to a classification dichotomy. Rather, we down-weight alignments that consist of language that is very common to all alignments. The underlying logic is that language that appears in a large number of other alignments is probably not related to specific policies. For example, alignments consisting of many stop words such as ‘the’ or ‘a’ or of common procedural language like ‘legislature’ or ‘section’ are most likely part of phrases that are either very frequent in common language or very frequent in the legislative context. In order to quantify the similarity to common alignments, we calculate the cosine similarity of the bag-of-words vector of each alignment with a random sample of 1000 alignments.⁵ The adjusted score for two bills X and Y is then:

⁵We rely on a random sample for computational reasons. Calculating the complete similarity matrix is computationally infeasible and unnecessary.

$$S_{X,Y}^* = S_{X,Y} \left(1 - \frac{1}{1000} \sum_{i=1}^{1000} \frac{\mathcal{A}_{\mathbf{X},\mathbf{Y}} \bullet \mathcal{A}_i}{\|\mathcal{A}_{\mathbf{X},\mathbf{Y}}\| \|\mathcal{A}_i\|} \right) \quad (3)$$

Where $S_{X,Y}$ is the alignment score, calculated as described in Section 3.1, \mathcal{A}_i is the bag-of-word vector representation of the matching alignment text where $i = 1, \dots, 1000$ are the sample alignments.

In this approach we follow the common intuition in natural language processing that the informativeness of text is inversely related to the frequency with which that text occurs (Robertson, 2004) (see, e.g., the regular application of term frequency–inverse document frequency (tf-idf) to pre- or post-process text-based datasets and results). Though we cannot speak to whether or not our approach is ultimately better than that taken by Wilkerson, Smith and Stramp (2015), in which they train a classifier to recognize and exclude boilerplate text based on human-labeled data, we note three relative strengths of our approach. First, we do not rely on the development and implementation of a human coding scheme, which is costly and always exhibits some degree of coding error. Second, our approach results in a continuous measure of the information content of text rather than a dichotomous classification of boilerplate or not. Third, and related to the first strength, our approach assigns an informativeness score to all word types, whereas reliance on human classifiers necessarily limits the boilerplate contributions of word types to the word types that were included in the ground truth labeled set. All this said, dealing with boilerplate/procedural text is an exciting open area of methodological inquiry for quantitative text-as-data research that is focused on legislation.

5 Empirical Evaluation of Validity

We use three strategies in order to assess the validity of text reuse as a measure of policy similarity. First, we evaluate whether there is a disproportionately high number of alignments between bills on the same policy area. Second, we analyze how well the bill similarity aggregated to the state level corresponds with policy diffusion networks identified in previous research (Desmarais, Harden and Boehmke, 2015). And third, we study the relationship between the ideological distance between the sponsors of bills and the alignment scores between those bills. If substantive policy content, and importantly, the same ideological direction of the provisions is detected by the alignment algorithm, we will expect an inverse relationship between these two measures. In the validation experiments that follow, we make use of the quantitative, interval level, alignment scores to construct continuous measures of policy similarity. We do this to impose as little information reduction as possible. In the conclusion we discuss the possibility of thresholding these scores in order to classify bill pairs as either containing or not containing matching policy content.

5.1 NCSL Tables

In our first validation exercise, we consider whether high alignment scores provide a reliable signal of similar policy proposals. To build a ground truth data set in terms of similar policy proposals, we rely on thematic tables published by the National Council of State Legislatures (NCSL).⁶ We collected a sample of these tables with the following procedure:

1. Collect all urls returned by the web search query: "site:ncsl.org "legislation.aspx""⁷
2. Sample 50 urls
3. Select bills that fulfill the following three conditions:

⁶www.ncsl.org

⁷We used the bing websearch API to collect the urls.

- (a) The website is a NCSL table
 - (b) The table refers (at least in part) to legislation introduced in 2011 or later.
 - (c) The table contains less than 100 individual bills.
4. Scrape all bills from these tables

Condition a) is necessary, since some of the urls returned by the web query refer to blog entries or to protected websites that require NCSL membership. The rationale behind condition b) is to maximize the number of bills we can match to our database (see Figure 2). We additionally constrain the number of bills per table to be below 100, in order to avoid tables that cover very broad topics. We chose 100 since this number would correspond roughly to two bills per state (often there is a house and a senate bill in each state for a particular policy). The web query from step 1 returned 266 urls. From the 50 sampled urls 22 fulfilled the three criteria. From these 22 tables we constructed a dataset of 950 bills, of which we could match 704 to our bill database. Since this is a relatively low number of bills, we then calculate the complete set of pairwise alignments for these bills. After removing same state bill pairs we obtained ca. 216,000 alignments.

Figure 4 displays the performance of the alignment score (left panel) in predicting if two bills are in the same NCSL table compared to the performance of a simple cosine similarity measure (right panel). We present precision—the proportion of bill pairs with a given alignment score that exceed the threshold on the x -axis that are in the same NCSL table, and recall—the proportion of same-NCSL-table pairs that exceed a given alignment score threshold. We see very high precision when setting the threshold to relatively high scores (> 100). Recall is low at thresholds that produce reasonable precision (above 100), but that is expected for our case, as (1) the introduction of similar policies in bills in tow different states is a rare event, and relatedly, (2) the appearance of two bills in the same NCSL table is a relatively rare event (8% in the current analysis); classifier recall is notoriously low in rare event data (Weiss and Hirsh, 2000)—even commonly approaching

zero, as in our case (Weiss, 2004). Further more, using a high precision - low recall classifier in our scenario where the total number of bill pairs is extremely large, produces a large number of interesting bill pairings in absolute terms.

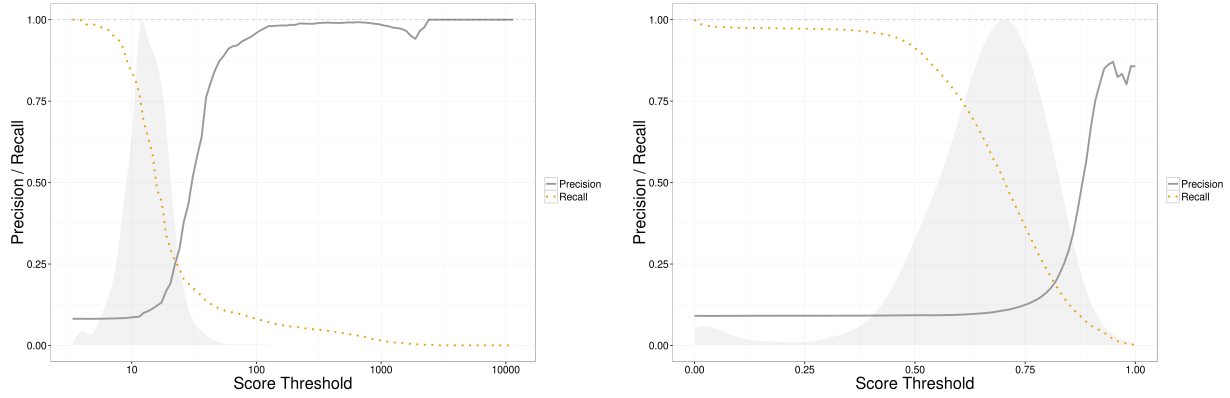
	AUC	P(X<Random)	P(X<Cosine)
Random	0.50		
Cosine	0.57	0.00	
Alignment	0.62	0.00	0.00

Table 2: Area under the curve for classifier based on thresholding alignment scores (Alignment), classifier based on thresholding cosine similarity score (Cosine) and random classifier (Random). The last three columns report one-tailed p-values for comparison between the AUCs. p-values are derived from 2000 non-parametric bootstrap iterations.

For a better understanding how the alignment score classification performs in comparison to other methods, in Table 2 we report the area under the ROC curve for the alignment classifier compared to a classifier based on cosine similarity of bill pairs and a random classifier. The first column contains the AUC values for each classifier. The remaining two columns display bootstrapped one-tailed p-values for the differences between the classifiers. Our results suggest that the presence of a large alignment score between bills is an effective indicator of policy similarity.

5.2 Diffusion networks and Text re-use

To evaluate whether text re-use corresponds to the transfer of policy, we test whether the presence of a diffusion network tie between two states is a predictor of text reuse. We use the policy diffusion networks inferred in Desmarais, Harden and Boehmke (2015). The diffusion networks were inferred using policy adoption sequences, and the network inference algorithm developed by Gomez Rodriguez, Leskovec and Krause (2010). A tie from state i to state j in the diffusion network indicates that state j has frequently emulated state i 's policies in the preceding thirty-five years. To calculate an aggregate alignment score for each state-pair we calculate the sum of the alignment scores associated with each pair



(a) Precision-Recall curve for classifier based on alignment scores. (b) Precision-Recall curve for classifier based on cosine similarity.

Figure 4: Precision-Recall curve for classifying two bills being in the same ncs1 table by thresholding the bill-dyad alignment score. The dotted (yellow) line is recall, the solid (black) line is precision. The x axis displays the threshold, the y axis represents precision and recall. The shaded area in the background displays the frequency distribution of the alignment scores.

of bills across the two states. The “Diffusion Ties” variable indicates the presence of a diffusion edge between states in the 2008 diffusion network, as measured by [Desmarais, Harden and Boehmke \(2015\)](#). The diffusion network in 2008 is inferred using policy adoptions in the 35 years preceding (and excluding) 2008. There is one observation in the analysis for each of the 1,225 unique state-pairs. Since this is dyadic data, we use a matrix permutation method, quadratic assignment procedure, to calculate p -values ([Krackhardt, 1988](#)). As a robustness check, we run the model with both the identity and log link.⁸ We also control for a dyadic variable (Coverage) equal to the product of the number of years for which we have data for each state in the dyad.

Results of the dyadic regression are presented in Table 3. In both specifications there is a positive relationship between the number of diffusion ties and the number of alignments. The relationship is statistically significant at the 0.05 level (one tailed) in each analysis. Furthermore, the magnitudes of the relationships are substantively significant. With the identity link, the presence of a diffusion tie leads to a 0.23 standard deviation increase

⁸The p -values were calculated using 5,000 random matrix permutations.

	Identity Link		Log Link	
	Coefficient	p-value	Coefficient	p-value
Intercept	-3.297	0.997	9.014	0.000
Diffusion Tie	0.238	0.042	0.239	0.050
Coverage	1.074	0.000	1.238	0.000

Table 3: Predicting aggregate cross-state alignment scores with diffusion ties. Coefficients calculated with OLS regression and normalized with standard deviation of cross state alignment scores. One-tailed p -values based on 5,000 QAP permutations.

in the expected aggregate alignment score. Based on the log link, the addition of a diffusion tie corresponds to a 30% increase in the expected aggregated alignment score.⁹ These results offer further evidence of the validity of quantifying text re-use as a measure of policy similarity, as the aggregate volume of text re-use between states is positively associated with previously-identified diffusion ties between states.

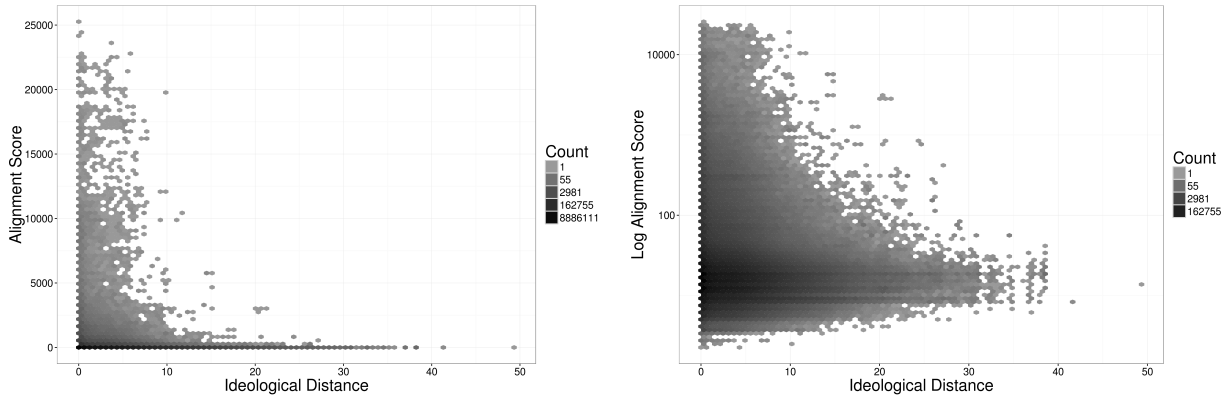
5.3 Ideological Distance of Aligned Bills

In the previous two validity tests, we focused mainly on the policy domain aspect of policy similarity, rather than ideological similarity. In the third and final validity test we ask whether we observe a greater volume of text reuse between bills introduced by legislators who are ideologically similar. For calculating ideological distance, we rely on latent ideology scores measured by (Shor and McCarty, 2011). The data set contains scores for 20,738 legislators from 50 state legislatures. The `openstates` API, which we can match to our bills data, contains data and identifiers on 12,000 legislators. Of these, we were able to uniquely match about 8,000 legislators to their records in the ideal point data. This allowed us to obtain ideal points for the sponsors of 58% of bills and $48 * 10^6$ (38%) pairs of bills. In order to assess the validity of text reuse as a measure of substantive policy overlap, we expect the quality of the alignments to be inversely correlated to the distance between the bills sponsors’ ideal points. The ideal points are

⁹Calculated as $100 \times [\exp(0.266) - 1] = 30.47$.

located on a common scale across all states, the distance between sponsors from different states is therefore meaningful.

In the following sections we present several analyses to assess this correlation. The ideal point of the bill is derived from the ideal point of the primary bill sponsor. For bills with several primary sponsors we use the average of those sponsors' ideal points to obtain a single measure. Figure 5 displays the distribution of ideological distance and log-alignment score for the roughly 50 million dyads in the dataset. The left panel displays the alignment scores on the natural scale. In order to get a better view of the distribution for small alignments, the right panel on the logarithmic scale. Each dot is a hexagonal tile, the shading of the tile represents the density in the area of the tile. Two major observations can be made in this plot. First, the vast majority of alignment scores are close to zero. Second, the triangular shape of the distribution indicates the high precision - low recall character of the alignment score discussed above: few bills exhibit meaningful policy overlap with other bills, but in the region of lower ideological distance we observe a greater occurrence of positive outliers, which arise from significant policy overlap.



(a) Hexbin plot with alignments on natural scale. (b) Hexbin plot with alignments on \log_{10} scale.

Figure 5: Hexbin plot of alignment score and ideological distance of all 50 million bill-dyads. The shading of the bins indicates the density of data in the bin.

In order to investigate this pattern more closely, we additionally calculate quantile regressions for the relationship between the alignment score and the ideological distance. In a

quantile regression, the dependent variable is the value of the predicted quantile of the distribution of the dependent variable conditional on the independent variables (e.g., the predicted median for median regression with the quantile set to 0.50 (Harden and Desmarais, 2011)). Figure 6 displays the coefficients from quantile regressions on a sequence of quantiles from the median to the 0.995th quantile. Confidence intervals are calculated by resampling bills, which represents an implementation of the clustered nonparametric bootstrap (Harden, 2011). Coefficients significant at the 95% level are displayed in black, insignificant coefficients are grey. We see that there is only a very weak relationship between the ideological similarity of bill sponsors and the median alignment scores between bills. However, as the quantile we model increases we see that the relationship grows strong and negative in magnitude, and is statistically significant. These results both validate the use of text reuse as a measure of policy similarity, and reinforce the high precision and low recall nature of text reuse as a measure of policy similarity.

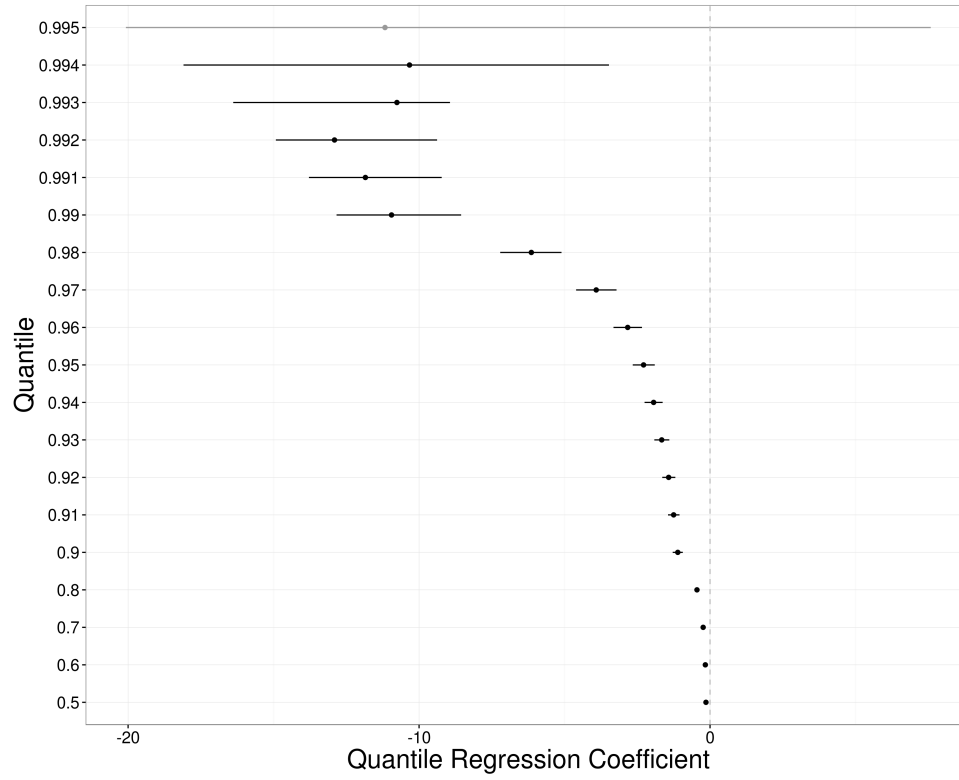


Figure 6: Coefficients of quantile regression for alignment score on ideological distance. Dots indicate the point estimate. Horizontal bar represent 95% confidence intervals obtained with clustered bootstrap. Note that the y-axis is not in a natural scale.

6 Conclusion

Scholarship on public policy and legislative politics relies heavily on measuring the content of legislation, especially with a relative or comparative approach. Manual comparison of bills is prohibitively time consuming when it comes to covering a large proportion of legislation introduced in one or more legislatures. We show that the automated detection of similar text strings in bills provides an effective approach to comparing bill contents. The log alignment scores serve as a highly valid summary measure of the similarity of two bills. We show high validity of this measure in three complimentary tests. First, it correlates with the presence of a diffusion tie inferred from patterns of policy diffusion. Second, it correlates with the ideological similarity between bill sponsors. Third, it exhibits extremely high precision in predicting whether two bills are listed in the NCSL policy area tables. We demonstrate the potential for using this data to test new or unexplored hypotheses on lawmaking in the states by testing whether bill text similarity correlates with campaign contributions to US state legislators.

The alignment scores we derive provide a resource for scholars to test hypotheses of the causes and consequences of the introduction and adoption of similar policies—a measure which crosses different policy areas, states, and time. In a validation experiment, we show that a wide span of quantiles of alignment scores between bills—from the median up through and above the 99th percentile—are negatively correlated with the ideological distance between the legislators who introduced the two bills. This result opens the door to many other hypotheses that could be tested using alignment scores for bill pairs. Where and when do similar policies cross state borders? Which legislators introduce similar legislation? Does lobbying by pressure groups result in the adoption of similar policies across states? Do similar patterns of campaign contributions to legislators predict similar policies in proposed legislation? Questions such as these can be directly investigated using the policy similarity measures we introduce.

The nature of our measure of policy similarity based on text reuse is that it is high precision and low recall. As such, in future research we advise scholars to focus on upper quantiles of the distribution of alignments when the inquiry requires the identification of bills that introduce equivalent policies. For predictive modeling, this can be done through the use of quantile regression at high quantiles. For searching out bill pairs, this can be done through thresholding on high alignment scores (e.g., 50+) to indicate the presence of highly similar policy proposals in legislation. Such thresholding may discard informative variation on policy similarity in terms of bill pairs that exhibit borderline overlap when it comes to containing equivalent policy content. On the other hand, high-score thresholding would focus the analysis on bill pairs that reliably contain very similar policy enactments, and pose little risk of a false positive.

References

- Berry, Frances Stokes and William D. Berry. 1990. “State Lottery Adoptions as Policy Innovations: An Event History Analysis.” *American Political Science Review* 84(2):395–415.
- Boehmke, Frederick J. 2009. “Policy Emulation or Policy Convergence? Potential Ambiguities in the Dyadic Event History Approach to State Policy Emulation.” *Journal of Politics* 71(3):1125–1140.
- Boehmke, Frederick J. and Paul Skinner. 2012. “State Policy Innovativeness Revisited.” *State Politics & Policy Quarterly* 12(3):303–329.
- Burgess, Matthew, Eugenia Giraudy, Julian Katz-Samuels, Joe Walsh, Derek Willis, Lauren Haynes and Rayid Ghani. 2016. The Legislative Influence Detector: Finding Text Reuse in State Legislation. In *KDD 2016, San Francisco, CA*.

- Desmarais, Bruce a., Jeffrey J. Harden and Frederick J. Boehmke. 2015. “Persistent Policy Pathways: Inferring Diffusion Networks in the American States.” *American Political Science Review* 109:392–406.
- Garrett, Kristin N and Joshua M Jansa. 2015. “Interest Group Influence in Policy Diffusion Networks.” *State Politics & Policy Quarterly* .
- Gerber, Brian J, Cherie Maestas and Nelson C Dometrius. 2005. “State legislative influence over agency rulemaking: The utility of ex ante review.” *State Politics & Policy Quarterly* 5(1):24–46.
- Gerrish, Sean and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning*. pp. 489–496.
- Gilardi, Fabrizio, Katharina Füglistner and Stephane Luyet. 2009. “Learning from Others: The Diffusion of Hospital Financing Reforms in OECD Countries.” *Comparative Political Studies* 42(4):549–573.
- Glick, Henry R. 1992. “Judicial innovation and policy re-invention: State supreme courts and the right to die.” *The Western Political Quarterly* pp. 71–92.
- Gomez Rodriguez, Manuel, Jure Leskovec and Andreas Krause. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM pp. 1019–1028.
- Gormley, Clinton and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide*. O’Reilly Media, Inc.
- Harden, Jeffrey J. 2011. “A bootstrap method for conducting statistical inference with clustered data.” *State Politics & Policy Quarterly* 11(2):223–246.

- Harden, Jeffrey J and Bruce A Desmarais. 2011. "Linear Models with Outliers Choosing between Conditional-Mean and Conditional-Median Methods." *State Politics & Policy Quarterly* 11(4):371–389.
- Huber, John D, Charles R Shipan and Madelaine Pfahler. 2001. "Legislatures and statutory control of bureaucracy." *American Journal of Political Science* pp. 330–345.
- Karch, Andrew. 2007. "Emerging issues and future directions in state policy diffusion research." *State Politics & Policy Quarterly* 7(1):54–80.
- Krackhardt, David. 1988. "Predicting with networks: Nonparametric multiple regression analysis of dyadic data." *Social networks* 10(4):359–381.
- Krause, Rachel M. 2011. "Policy innovation, intergovernmental relations, and the adoption of climate protection initiatives by US cities." *Journal of Urban Affairs* 33(1):45–60.
- Mooney, Christopher Z and Mei-Hsien Lee. 1995. "Legislative morality in the American states: The case of pre-Roe abortion regulation reform." *American Journal of Political Science* pp. 599–627.
- Mycoff, Jason D, Michael W Wagner and David C Wilson. 2009. "The Empirical Effects of Voter-ID Laws: Present or Absent?" *PS: Political Science & Politics* 42(01):121–126.
- Potthast, Martin, Matthias Hagen, Tim Gollub, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efstathios Stamatatos and Benno Stein. 2013. Overview of the 5th international competition on plagiarism detection. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*. CELCT pp. 301–331.
- Robertson, Stephen. 2004. "Understanding inverse document frequency: on theoretical arguments for IDF." *Journal of documentation* 60(5):503–520.

- Scott, Patrick G. 1997. "Assessing determinants of bureaucratic discretion: An experiment in street-level decision making." *Journal of Public Administration Research and Theory* 7(1):35–58.
- Shor, Boris and Nolan McCarty. 2011. "The Ideological Mapping of American Legislatures." *American Political Science Review* 105:530–551.
- Simmons, Beth A and Zachary Elkins. 2004. "The globalization of liberalization: Policy diffusion in the international political economy." *American Political Science Review* 98(01):171–189.
- Smith, Temple F and Michael S Waterman. 1981. "Identification of common molecular subsequences." *Journal of molecular biology* 147(1):195–197.
- Volden, Craig. 2006. "States as Policy Laboratories: Emulating Success in the Children's Health Insurance Program." *American Journal of Political Science* 50(2):294–312.
- Walker, Jack L. 1969. "The Diffusion of Innovations Among the American States." *American Political Science Review* 63(3):880–899.
- Weingast, Barry R and Mark J Moran. 1983. "Bureaucratic discretion or congressional control? Regulatory policymaking by the Federal Trade Commission." *Journal of Political Economy* 91(5):765–800.
- Weiss, Gary M. 2004. "Mining with rarity: a unifying framework." *ACM SIGKDD Explorations Newsletter* 6(1):7–19.
- Weiss, Gary M and Haym Hirsh. 2000. Learning to predict extremely rare events. In *AAAI workshop on learning from imbalanced data sets*. pp. 64–68.
- Wilkerson, John, David Smith and Nicholas Stramp. 2015. "Tracing the Flow of Policy Ideas in Legislatures: A Text Reuse Approach." *American Journal of Political Science*

59(4):943–956.

URL: *<http://dx.doi.org/10.1111/ajps.12175>*

Appendix

Proof that SW alignment is globally optimal

F_i is the alignment of size i with $i = 1, 2, \dots, (n+k-1)$. $S(F_i)$ is the score of the alignment.

1. There are four options for F_1 (Gap in \mathcal{A} , Gap in \mathcal{B} , Mismatch, Match) . This alignment is optimal when $S(F_1)$ is maximized
2. Every alignment's score $S(F_i)$ with $i > 1$ when maximized, is composed of the score of the 'one-shorter' alignment $S(F_{i-1})$ and the score of the i -th element. If not the score could be increased by choosing a different F_{i-1} that has a higher score.
3. By induction therefore follows that every alignment represented in the dynamic programming matrix (Figure 1) and calculated according to Equations 1 and 2 is an optimal alignment ending in the locations specified by the row and column index in the matrix.
4. The dynamic programming matrix contains all possible alignments. Since every cell is optimal, the highest score in the matrix must be globally optimal.