A Political Economy Approach to the Grammar of Institutions: Theory and Methods

Matia Vannoni

This article proposes a political economy approach to the grammar of institutions, by building on the study of contracts in economics. First, I argue that more attention needs to be paid to the costs and benefits of institutions, thus allowing researchers to derive empirically testable hypotheses on the evolution of institutions. In this vein, the role of conditions in studying the grammar of institutions becomes central. Second, I propose, validate, and test an innovative method to study the grammar, based on computational linguistics. This method allows extracting the elements of the grammar in an automated and unsupervised manner, paving the way to large-N analyses. In so doing, this article seeks to bridge the gap between public policy and political economy.

KEY WORDS: public policy, grammar of institutions, NLP

Este artículo propone un enfoque de economía política a la gramática de las instituciones, basándose en el estudio de los contratos en economía. Primero, sostengo que se debe prestar más atención a los costos y beneficios de las instituciones, permitiendo así a los investigadores derivar hipótesis empíricamente comprobables sobre la evolución de las instituciones. En este sentido, el papel de las condiciones en el estudio de la gramática de las instituciones se vuelve central. En segundo lugar, propongo, valido y pruebo un método innovador para estudiar la gramática, basado en la lingüística computacional. Este método permite extraer los elementos de la gramática de forma automatizada y sin supervisión, allanando el camino a análisis de N grande. Al hacerlo, este artículo busca cerrar la brecha entre las políticas públicas y la economía política.

PALABRAS CLAVE: políticas públicas, gramática de instituciones, PNL
Introduction

One of the most successful institutionalist frameworks recently used to study public policy is the Institutional Analysis and Development (IAD) framework and, within this framework, the literature has revamped the grammar of institutions (Siddiki et al., 2019), originally proposed in Crawford and Ostrom (1995). In its original formulation, the grammar of institutions provided a simple way to code different elements of institutional statements, thus being able to categorize statements in order of complexity. The contemporary public policy literature has theoretically refined this code and empirically applied this approach to study policy change. In the last decades, a substantial body of work in public policy has developed in this realm.

Theoretically, I bring together the economic literature on optimal contracting and the public policy work on the grammar of institutions. In so doing, I argue that more attention should be paid to the (long-term) costs and benefits of institutions. This new theoretical focus bears some important analytical advantages: a shift from statics to dynamics and more attention to conditions. Empirically, I propose syntactic dependency parsing as a method to extract the main elements of the grammar in an automated and unsupervised manner. This method is validated against a set of manually coded provisions and then applied to a dataset of the U.S. state legislation in the second half of the twentieth century, to test the main proposition set out in the theoretical part. In so doing, I bring together theory and data-driven approaches to the institutional grammar, by providing a computational approach to test explicit hypotheses on the evolution of institutions.

A Public Policy Approach to the Grammar

The grammar of institutions, formally proposed in Crawford and Ostrom (1995), provides an approach to study institutions that in turn should be seen within the broader Institutional Analysis and Development (IAD) framework (McGinnis, 2011). More specifically, it breaks down the elements that make up institutional statements, providing a categorization of these statements. Five elements are identified: Attribute (A), Deontic (D), aIm (I), Conditions (C), and Or else (O)—the ADICO framework (Crawford & Ostrom, 1995; Ostrom, 2009). The Attribute refers to the agent of the statement, the aIm is the action, the Deontic refers to whether the statement is prescriptive (namely whether it contains a deontic modal, such as “shall” or “must”), the Conditions define the scope of application of the statement, and the Or else element refers to the sanction for non-compliance (Crawford & Ostrom, 1995). Different combinations of these elements give different categories:

- Attribute + aIm + Condition (AIC): shared strategy. An example of a shared strategy is “All citizens follow the rules all the time”.
- Attribute + Deontic + aIm + Condition (ADIC): norm. An example of a norm is “All citizens must follow the rules all the time”. The difference from the shared strategy above is the presence of the Deontic “must”.
- Attribute + Deontic + aIm + Condition + Or else (ADICO): rule. An example of a rule is “All
citizens must follow the rules all the time or else face arrest”. The difference with the norm above is the presence of the Or else element “or else face arrest”.

In the last decade, the public policy literature has revamped the “grammar of institutions” (Dunlop, Kamkhaji, & Radaelli, 2019). The attention has mostly been on revising the list of elements and giving more attention to the relationship between sentences. A strand of the literature has focused on removing redundant elements, such as the Or else, hence making the distinction between rules and norms redundant, adding more elements, such as the Object, and reviving some old elements that did not receive much attention in the original formulation, such as the Condition (Basurto, Kingsley, McQueen, Smith, & Weible, 2010; Siddiki, Basurto, & Weible, 2012; Siddiki et al., 2019; Siddiki, Weible, Basurto, & Calanni, 2011). Another strand has focused more on the link between sentences, focusing on the concept of nestedness (Frantz, Purvis, Nowostawski, & Savarimuthu, 2013; Basurto et al., 2010). The main argument is that not all sentences in a piece of legislation are on the same level and some are more specific than others. For instance, the sentence “Citizens must report their taxes on time, or else tax officials must fine citizens under any circumstance” consists of two statements: “Citizens must report their taxes on time” and “tax officials must fine citizens under any circumstance”, connected with “or else” (Frantz & Siddiki, 2020). The two statements are hierarchically nested and the latter statement is consequential to the former one (Frantz & Siddiki, 2020). These two strands have recently come together in the so-called Grammar 2.0 (Frantz & Siddiki, 2020), which brings together the refined coding of the grammar elements of the first strand with the nested approach of the second strand.

There are several empirical applications of the grammar. For instance, Basurto et al., (2010) were among the first to apply the grammar to an empirical case. They compare the U.S. Transportation Bill and the Georgia Abortion Bill, finding that the former is more complex in content, as it deals with the federal and state level, and in structure, with more nested statements. Siddiki (2014) uses the grammar to code state aquaculture policies in Virginia and Florida and compare them in terms of coerciveness, by looking at the number of deontic modals. Weible and Carter (2015) compare Colorado’s 1977 and 2006 smoking bans by relying on the grammar of institutions. More specifically, they compare the objectives of the two bans and how these objectives are achieved with different combinations of statements restricting or allowing certain behaviors targeted to different populations under certain conditions, and what sanctions are applied.

A Political Economy Approach to the Grammar

In this paper, I draw from theories and concepts from economics, more specifically from contract theory, and argue that more attention should be paid to the costs and benefits of institutions. Indeed, the contract theory in economics starts with conceiving institutions as a series of contracts between parties interacting repeatedly with each other and focuses on the costs and benefits of entering those contracts (Battigalli & Maggi, 2002, 2008). The benefits consist of the surplus created by the
interaction that is under contract. Assume a contract regulating a relationship between an employer and a worker, the surplus is the worker’s output. The costs are determined by the type of provisions contained in the contract that in turn give rise to different writing, monitoring, and enforcement costs (Battigalli & Maggi, 2002, 2008). For instance, leaving complete discretion to the parties does not bear any (present) costs, whereas contingent provisions that consider all the likely scenarios are very costly to write, monitor, and enforce. Rigid provisions (called spot clauses, in this approach) that dictate a specific course of action are more costly than total discretion but less costly than contingent provisions. It is easy to see that some relationships bear more potential surplus than others, thus outweighing the contracting costs. For instance, it is not worth writing a detailed contract (or writing a contract at all) for a one-off delivery of a simple service, such as a gardener mowing your lawn once. Yet, it is worth writing a detailed contract for a permanent employee in your company whose work contributes to the value of the company, for instance.

More formally, a contract consists of clauses or provisions that in turn consist of a set of sentences describing events and actions, as well as logical connectives (Battigalli & Maggi, 2002, 2008). In the clause “A and B happen then Y does X”, “A” and “B” are events and “X” is the action, whereas “and” and “then” are connectives. By combining these different sentences, there are three main ways to regulate an action: discretion, spot clause, and contingent clause (Battigalli & Maggi, 2002, 2008). Discretion is the lack of any provisions setting out a course of action (no clause mentioning the action “X”). In other words, there is no formal regulation of the action with discretion. Discretion in legislation is closed to regulating something with what in the grammar literature is called a shared strategy (Crawford & Ostrom, 1995). A spot clause, instead, reads like: “Y does X”. This means that, regardless of the events, the action “X” always needs to be performed (or better, there is no qualifier on the action). In the grammar vocabulary, this can be associated with a rule or a norm (Crawford & Ostrom, 1995). Finally, a contingent clause specifies what actions need to be taken in which events: “A and B happen then Y does X”. In the grammar language, a contingent clause can be arguably associated with a rule or a norm containing a condition. Indeed, the sentence describing the events “A and B happen” is a condition to the sentence describing the action, namely the Attribute Y performing the aim X. The costs increase from discretion to contingency, as the costs are assumed to be proportional to the number of elements in clauses (i.e., sentences describing actions and events) (Battigalli & Maggi, 2002, 2008). In other words, the costs of writing, implementing, and enforcing provisions are proportional to the clauses contained in those provisions. As for the benefits, the surplus is maximized when the parties in the contract perform all the tasks, matching actions with events. The main goal of the parties of the contract is to minimize costs and maximize benefits, thus writing a contract with the least number of sentences, but which allows for all the tasks to be performed.

Not only present but also future costs are considered. Indeed, this literature sees contracting as a reiterated relationship between parties. As a consequence, a spot clause might be less costly today, because it does not require the specification of events (i.e., fewer sentences are required), but in the long run it might be more
costly than a contingent one. Indeed, when the context changes and the events under consideration change too, a spot clause requires constant re-writing, whereas a contingent clause requires only the addition of a further sentence. This attention to future costs implies a shift from statics to dynamics. The political economy literature identifies three dynamics: writing a contingent clause once for all, modifying a spot clause, and replacing a spot rule with a contingent clause (what is called “enrichment”) (Battigalli & Maggi, 2002, 2008). The literature derives some straightforward predictions on these dynamics (Ash, Morelli, & Vannoni, 2020; Battigalli & Maggi, 2002, 2008). An increase in potential surplus in a relationship increases the benefits, thus justifying higher costs and more complexity in the contract, namely more contingent clauses.

The political economy literature has already studied institutions under this light (Ash, Morelli, & Vannoni, 2020; Demsetz, 1967; Mulligan & Shleifer, 2005). The main assumption is that setting up a new institution takes high fixed costs, like writing a new clause from the scratch, but then revising it, like adding a sentence on events or actions, becomes less expensive (i.e., lower incremental costs). For instance, Mulligan and Shleifer (2005) find that the introduction of conscription in France was more successful than in many other countries, thanks to the already developed public administration, which reduced the incremental costs of introducing these reforms. This approach has applied the insights on the costs and benefits of clauses from contract theory to study the (long-term) evolution of institutions.

In this paper, I argue that studies of the grammar should take a similar approach: the grammar should be used to explain institutions. It is easy to derive some testable hypotheses at the macro-level on the drivers of legislative complexity. For instance, technological changes that increase the potential surplus of the economic relations in a system should lead to enrichment, because more complex legislation allows economic actors to perform all the tasks needed to meet that surplus. This approach bears an important analytical advantage: more attention to conditions. This has recently emerged in the public policy literature (Siddiki et al., 2019), but further attention should be paid to it.

This political economy approach provides also some guidance on what matters less from an analytical perspective. The public policy literature focuses on the distinction between constitutive and regulatory clauses (Siddiki et al., 2019; Weible & Carter, 2015). The latter express the obligations, prohibitions, and permissions, and related conditions, namely what discussed so far, whereas constitutive clauses are broader clauses that “set the stage” (Siddiki et al., 2019). In other words, they define concepts, such as actors, actions, and so on. Constitutive clauses are very important for policy outcomes. For instance, Weible and Carter (2015) focus on the importance of the definition of “cigar bars” in smoking bans and find that different definitions led to different policy outcomes. One could see constitutive clauses as complementing condition sentences in describing the states of the world to which regulatory clauses apply. Yet, the attention to constitutive clauses is more suitable from a public policy approach that uses the grammar to compare policy outcomes than from a political economy approach that focuses on the variation in costs and benefits of provisions and what drives this variation. Moreover, although the focus
on the difference between constitutive and regulatory clauses is less important from the analytical perspective proposed here, distinguishing between the constitutive (setting out the “rules of the game”) and the operational level (related to day-to-day activities) of the analysis is very useful to answer some important questions, such as whether the evolution of institutions reaches a long-term equilibrium.

In conclusion, important lessons for the public policy literature can be drawn. First, closer attention to costs and benefits, which in turn can help shift the analytical focus to the study of the (dynamic) causes of institutions, is needed. Moreover, the public policy literature should carry on with its latest developments and keep focusing on the concept of conditions. In the next sections, I will provide an unsupervised method to measure conditions, and then, I will test the claim derived from the political economy literature on the link between technological innovation and contingent regulation, namely that technological innovation is associated with more contingent clauses regulating the economy.

**Syntactic Dependency Parsing**

In this section, I introduce the computational linguistics tool used in this work to study the grammar. More specifically, the first part discusses how syntactic parsing works and what its output is. Then, I provide some practical examples of parse trees for simple sentences. Finally, I conclude by introducing the extraction rules that will allow using the output of the syntactic parser to extract the elements of the grammar from statements, with a straightforward off-the-shelf approach. The next section will provide some preliminary validation to this approach, before discussing some of its limitations. In the final section, I will use this approach to provide some evidence for the claims made above on the link between contingent clauses and technological change.

In this work, I use the Python package spaCy. This is one of the best performing and most reliable dependency parsers on the market (Choi, Tetreault, & Stent, 2015). This is how it works. First, the parser segments the text into sentences and tokenizes them, namely it divides the text into words/phrases (related to a concept). Second, the algorithm tags the parts of speech (a procedure called POS tagging). In order words, it tells us whether the word is a noun, a verb, an adjective, and so on. The algorithm underlying the spaCy POS tagger is trained on a manually annotated corpus of these tags and when applied to the text under analysis, it makes a prediction on the likelihood that a tag applies to a specific word. For instance, the algorithm can easily predict that the word “the” is a determiner (DET). In other instances, the algorithm uses the context of the word. For instance, it is most likely that any word followed by “the” is a noun.

Once we know what every single word/phrase represents, namely whether it is a determiner, a noun, or a verb, the parser extracts the grammatical relationship between words, namely the syntactic dependences (Jurasky & Martin, 2000). For instance, it tells us whether a noun is (most likely to be) the subject of a verb or the object. As with the POS tagging, the algorithm is trained on a corpus of manually annotated syntactic dependencies (so-called dependency treebanks) and makes
predictions on the dependencies in the text under study (Goldberg & Nivre, 2012; Honnibal, Goldberg, & Johnson, 2013). In other words, the parser goes through the list of words in a sentence one by one and decides whether and how a word is related to the one before and the one after, by consulting a set of rules based on manually coded dependencies. These rules, in turn, rely on other linguistic information, such as the form and the POS tag of the word, and instruct the parser that, for instance, an active verb followed by a noun means that these two are likely to be related by a direct object relationship. Although some recent work in political science and political economy uses dependency parsing (O’Connor, Stewart, & Smith, 2013; Van Atteveldt, Kleinnijenhuis, & Ruigrok, 2008; Vannoni, Ash, & Morelli, 2020), applications in public policy are still in their infancy.

More formally, the result of the syntactic parsing is the structure of the sentence. Each structure consists of words and links (the arrows in the figures below). The links represent the grammatical relationships between the words and have a specific direction (Jurasky & Martin, 2000). These structures are usually represented as dependency trees, where there is a single root, each word is linked to each other (more specifically, each word, apart from the root, has a single incoming link) and there is a unique path from the root to each of the other words (Jurasky & Martin, 2000). Hence, each sentence is represented in a way that allows having a single head, all words are connected with each other and there is a single path from each word to the head. In the following section, I provide some examples of dependency trees.

**Examples of Dependency Trees**

This section provides some examples from the literature. I take some of the provisions manually coded in Siddiki et al., (2012) and run them through displaCy (spacy’s built-in dependency visualizer), which shows the part-of-speech tags below the words and the syntactic dependencies above with arrows. These sentences were selected for illustrative purposes because of their structure.

The first sentence is “Fish Health Board must constitute a public entity” (Figure 1). First, displaCy provides the part-of-speech tags: “Fish”, “Health”, and “Board” are tagged as proper nouns; “must” and “constitute” are verbs; “a” is a determiner; “public” is an adjective; “entity” is a noun. Then, the parser provides the syntactic dependencies, namely the grammatical relationships between words (Jurasky & Martin, 2000). It finds the head of the sentence, namely the main verb (“constitute”), and associates it with its auxiliary (“must”), its nominal subject (“Board”), and its direct object (“entity”). Then, it associates “Board” with its compound names (“Fish” and “Health”) and “entity” with its determiner and its adjectival modifier (“a” and “public”, respectively).

Figure 2 shows the result for the second sentence: “Fish Health Board must meet at least once a year”. As above, the parser identifies the part-of-speech tags. In this case, the new tags are adposition for “at” and adverb, associated with “once”. The interesting part for the syntactic parsing is the adverbial modifier relationship between the main verb “meet” and “once”. Moreover, the parser identifies “year” as
the noun adverbial modifier of “once”. Finally, it shows that “at” and “least” are in turn adverbial modifiers of “once”.

Figure 3 shows the third sentence: “Fish Health Board must elect a Chairman and Vice-Chairman annually”. Here, it is interesting to notice that the parser detects “Chairman” as a direct object and “annually” as an adverbial modifier. Then, it links the two “Chairman” through a conjunction relationship.

Figure 4 has been made up for illustrative purposes, by adding the phrase “unless instructed by the Governor” to the sentence in the previous figure. As can be seen, the results for the first part of this new sentence are identical to those for the old sentence. Then, the parser links “instructed” to the main verb “elect” as an adverbial clause modifier. Finally, the phrase “unless instructed by the Governor” is parsed separately, finding an agent and an object linked to the main verb “instructed”.

Finally, Figure 5 has also been made up for illustrative purposes. It is the same sentence as in Figure 3, but here the sentence is in its passive form. As can be seen, everything stays the same, but “Chairman” becomes the passive nominal subject and “Board” the object. Moreover, “be” is coded as a passive auxiliary.

Extraction Rules for the Grammar

By building on the discussion on the elements of the grammar and the illustrations of the syntactic parsing above, this section sets out the extraction rules that allow detecting the different elements of the grammar, by relying on the information obtained from the syntactic parser (i.e., the parse tree). In other words, I define the syntactic relationships between the different elements of the grammar ex-ante and then I compare them with the results from the parser (Table 1). It should be noticed that these extraction rules are decided in a rather arbitrary manner by the researcher, based on the knowledge of the grammar. More advanced applications of this information extraction approach rely on lexical databases, such as FrameNet, to identify
the linguistic features of abstract concepts (Vannoni, Ash, & Morelli, 2020). In this paper, I use the extraction rules for illustrative purposes. Also, as seen above, the application of the syntactic parser in this paper is mainly “off-the-shelf”, namely it does not require much input from the researcher.7

First of all, it should be noticed that there is no attempt to operationalize the Or else element. As seen above, there is a wide consensus that this category has become redundant, especially in coding written legislation (Siddiki et al., 2012, 2019). Indeed, it is extremely rare for legislators to explicitly state penalties or sanctions in an Or else form or even to state them at all. For instance, in the manual coding of U.S. national organic program regulations, only four of 746 statements contain an Or else element (Carter et al., 2016). In Basurto et al., (2010), only four of 245 statements are coded as rules, hence containing the Or else element.

The Deontic element is operationalized with the auxiliary. The Deontic tells us what is permitted (in the case of “may” and “can”), what is obliged (“must” and “shall”), and what is forbidden (“shall not”) (Ostrom, 2009). The transposition of the
element of the grammar into a syntactic dependence is rather straightforward. In fact, the Deontic is almost always linked to the main verb through an auxiliary syntactic dependence relationship. In the examples above, “must” is always coded as auxiliary. It should be noticed that recent political economy work suggests that the presence of a deontic modal, like “shall” or “must”, is not a necessary or sufficient condition to express an obligation (Vannoni et al., 2020). Yet, for simplicity’s sake, in this work, we assume that it is.

Also, the aIm, the Object, and the Attribute are rather easy to transpose into syntactic dependencies. First, the aIm is the action described. In the sentences in Figures 1 and 2, the aIm is, respectively, “constitute” and “meet”, which are the heads of the sentences in syntactic terms. As seen above, the head is the main component of the sentence and is at the center of the parse tree. Second, the Object is a new category and refers to the receiver of the aIm (Siddiki et al., 2011). I coded it with the (indirect) object of the sentence. In Figures 1 and 3, entity and Chairman are both the direct objects of the sentences. It should be noticed that in some cases, the Object can also be an indirect object. In passive sentences like that in Figure 5, the receiver of the aim becomes the passive subject, such as Chairman. Finally, the Attribute, namely the actor(s) supposed to carry out the action, is the nominal or clausal subject. In all the examples above, apart from Figure 5, “Board” is the nominal subject. In Figure 5, “Board” becomes the object, as this is a passive sentence. Hence, the Attribute in passive sentences is coded as a (direct or indirect) object.

The most difficult part of coding the grammar into syntactic dependencies is coding the Condition. The latter constrains the scope of the action from a space and temporal perspective (Siddiki et al., 2019). Figures 3–5 contain a Condition: “annually” and “unless instructed by the Governor”. Both are coded as adverbs, respectively, as an adverbial modifier and adverbial clause modifier. The three most common conditions are usually related to space, time, and state. These conditions in turn are usually expressed either with adverbs, such as “annually” or “locally”, or with phrases starting with a preposition, such as “every two years” or “every citizen”.

It should be noticed that the syntactic parser provides some useful information for a more advanced study of the grammar. Indeed, the parser tells us which tokens are syntactically linked with each element. For instance, in Figure 1, we know that the Object “entity” is associated with the adjectival modifier “public”. The same can be applied to the aIm. This will prove to be very useful in those applications of the grammar that use different levels of abstraction, where at one level, the researchers need to know only the Object and at another level, they need to know also its properties (Frantz & Siddiki, 2020).

Validation

We have seen so far how researchers can use simple categories created ex-ante, based on the grammar, and an off-the-shelf application of a syntactic parser in order to extract the elements of the grammar. In this section, I validate this simple approach
against a series of provisions manually coded by experts in the field (Carter et al., 2016).8

First, I start by showing the output of the code used for the analysis of the following sentence: “When preventive practices and veterinary biologics are inadequate to prevent sickness, a producer may administer synthetic medications”. As shown in Table 2, the “producer” is coded as the Attribute (in this case, there are no properties), the Deontic is “may”, the sentence is not negative, the aIm is “administer”, the Object is “medication”, and the Object with its properties is “synthetic, medication”. Finally, the Condition is “when, preventive, practice, and, veterinary, biologic, be, inadequate, to, prevent, sickness”. As can be seen, in order to make the output more externally valid, the code extracts the lemma of each token, namely the base form of the word (in this case, “are” becomes “be”).

The first validation exercise is carried out on the whole sample of statements. I apply my code using the Python package spaCy to those statements and the result is a dataframe where each row corresponds to a statement and the columns represent the elements of the grammar resulting from the original hand-coding and my machine-coding (see Table 2). To find the degree of agreement between these two sets of results, I use a fuzzy match function in Python that compares strings with different lengths and order of words, providing a similarity score. For instance, if the hand-coded Deontic and the machine-coded Deontic are, respectively, “must” and “must”, this is a match with a 100 score. If the machine-coded Attribute is “procedure” and the hand-coded one is “the procedure”, this is a match (even though the score is below 100). Instead, if the machine-coded Attribute is “procedure” but the hand-coded Attribute is “administer”, this is not a match (and the score will be close to zero).9

By using this function, I compare my machine-coded results with the hand-coded ones. Findings are rather promising and suggest that syntactic parsing, even in its off-the-shelf form, may be useful in studying the grammar. Indeed, for the Deontic, I find a 47 percent agreement, for the Condition, the agreement is 53 percent, for the Attribute it is 88 percent and 61 percent for the aIm.

Yet, applying the code to the whole set of statements is problematic, because of the different units of analysis. Indeed, the statements used for this part of the analysis have been manually extracted from the legal text based on subjective criteria, resulting in high variation in terms of length and content. Syntactic parsing, instead,

| Category                  | Result                                      |
|---------------------------|---------------------------------------------|
| “Attribute”               | “Producer”                                  |
| “Object”                  | “Medication”                                |
| “Deontic”                 | “May”                                       |
| “neg”                     | –                                           |
| “aIm”                     | “Administer”                                |
| “Object_properties”       | “Synthetic”, “medication”                   |
| “Attribute_properties”    | “A”, “producer”                             |
| “Condition”               | “When”, “preventive”, “practice”, “and”, “veterinary”, “bio-
|                           | “logic”, “be”, “inadequate”, “to”, “prevent”, “sickness” |
segments the text into sentences based on objective syntactic criteria, before providing parts-of-speech and syntactic dependence tagging. This makes comparing the results on the manually segmented statements and on the machine segmented ones problematic. Hence, I perform a second validation exercise, where I restrict the sample to short sentences that are closer to the unit of analysis of the syntactic parser. More specifically, I restrict the sample to those sentences with a single head, which represent roughly half of the whole dataset. This is in line with how the parser would segment a corpus. I use the same approach explained above to compare results. When using a similar unit of analysis, syntactic parsing performs much better for all the categories (apart from Attribute, for which the performance is very similar). The agreement for Deontic is 59 percent, for Condition is 66 percent and for Attribute and alm is, respectively, 85 percent and 71 percent.

Finally, I focus on the performance of the code to extract the Condition element and use a more standard approach to validation, by calculating the weighted precision, recall, and F-score measures. In this case, I test the performance on whether the code extracts a condition against whether a condition is manually coded (regardless of the actual content of the condition). Results are rather good: precision is 0.87, recall is 0.78, and the F-score is 0.85.

In conclusion, although performance varies across elements, with some elements showing rather low performance, off-the-shelf applications of syntactic parsing perform relatively well in extracting the elements of the grammar. This finding is subject to some caveats. First of all, performance depends on the unit of analysis. Syntactic parsers, like SpaCy, perform better on shorter sentences, closer to how they would segment the text according to the syntax of the provision. This would be less of an issue if the starting point of the analysis was a whole corpus and not already segmented sentences, as in the validation exercise above. Second, parsers perform best on the category alm and worst on the Deontic one. In fact, the alm, coded with the head of the sentence, is a syntactically central feature and very easy to extract. Moreover, the alm of a sentence cannot be assumed by the reader, whereas the other categories can and sometimes legislation is written in such a way that the reader needs to go back to the previous sentence/statement to understand what the Attribute is, for instance. This might explain the relatively poor performance of the Deontic. The reason is that in the hand-coded set of provisions, sometimes the Deontic is assumed from some previous information. In a list of obligations, for instance, it is common to specify the Deontic at the beginning and then simply list the main verbs (alm). Text such as the following is very common: “Under the current rules, the agent shall: perform action A, perform action B and perform action C”. In this case, the Deontic (“shall”) is present only in the first sentence and it is implied in the following sentences. Finally, it should be noticed how the code performs relatively well on the Condition element. Usually, conditions are among the most difficult elements to code (Basurto et al., 2010). Finally, caution should be used when using hand-coding as a benchmark for validation. Although hand-coding performs rather well on the components of the grammar, the agreement between coders across all components is on average 80 percent, far from the perfect agreement (Siddiki et al., 2011).
Analysis

In this section, I test the main hypothesis formulated above, namely that techno-
logical change leads to more contingent regulation. I do so in the context of the U.S.
states, in order to leverage time and space variation. I apply the approach outlined
above to a corpus consisting of the whole U.S. state legislation from 1964 to 2000.10

I preprocess the corpus and then extract the contingent provisions with the
approach above, using the Python package spaCy. I extract those provisions that con-
tain one of the following syntactic elements (see the table with the extraction rules
above): adverbial (clause) modifier or prepositional (clausal) modifier. I also extract
those provisions that do not contain any of those syntactic elements (what above I
call spot clauses). Finally, I apply a topic modeling algorithm (the LDA algorithm
with the Python gensim package) to the corpus. This is an unsupervised technique,
commonly used in contemporary social research that extracts clusters of words (top-
ics) from documents based on the frequency with which they appear in the same
sentence and document. In doing so, I associate (contingent and non-contingent)
provisions to different topics. The algorithm suggests that the optimal number of
topics is 42. I then manually code those topics into macro-categories to make results
more understandable. For the analysis below, I focus only on two macro-categories:
social and economic. The result is a dataset with the number of economic and social
(non-)conditional provisions per state per year. I measure technological innovation
with the (logged) number of trademarks, which is a standard measure in the innova-
tion literature (Gotsch & Hipp, 2014). The number of trademarks in an economy is a
valid proxy of the output of R&D in that economy, namely the number of inventions
that are available on the market. The empirical expectation I test in this section is
that technological innovation increases the number of economic conditional clauses.

I first show the trend of the share of conditional provisions on total economic
provisions over time (I use the number of economic conditional provisions divided
by the total number of economic provisions), averaged across states (Figure 6). The
figure shows that U.S. states have increasingly regulated the economy with condi-
tional provisions in the past decades. It should be noticed that the increase in the
share of conditional provisions is not constant and, rather, we observe that the trend
starts leveling off at the beginning of the 1990s. These two findings suggest that pol-
cymakers use contingent regulation in a strategic manner, considering its costs and
benefits. As seen above, regulating the economy with more complexity (in this case,
more contingent provisions) bears some costs, and policymakers do so only if the
potential benefits outweigh those costs.

I then test the effect of technological innovation on the share of conditional
provisions among economic and social provisions. Table 3 shows the results of
regressing the share of (non-)conditional provisions over the total number of pro-
visions in economic and social topics in a state in a year on the (logged) number of
trademarks in that state in that year. I use state fixed effects to control for any time-
invariant state-level confounding factors, as some states might already have higher
levels of technological innovation and contingent regulation for some other reasons
(than the hypothesis above). I also use year fixed effects to control for nationwide
time-varying factors and state-level time trends to allow for preexisting confounding trends, as it might be that both technological change and contingent regulation increase over time (without being correlated), at the same rate across states or with different rates. This controls for the fact that regulation in states with high initial levels of contingency might evolve differently than in those states with low initial levels of (contingent) regulation. Finally, I cluster standard errors at the state level. This research design allows testing the main empirical implication discussed above, namely that technological innovation is one of the main drivers of the potential surplus of economic relations and hence is positively related to contingent regulation.

I find a significant and positive effect for conditional economic provisions (Column 1 in the table), but not for non-conditional economic provisions (Column

**Figure 6.** Economic Conditional Provisions over Total Economic Provisions, 1964-2000.

**Table 3.** Effect of the (Logged) Number of Trademarks on Share of (Non-)conditional Provisions over Economic and Social Provisions

| Variables         | (1) Cond Econ | (2) Non-Cond Econ | (3) Cond Soc | (4) Non-Cond Soc |
|-------------------|---------------|-------------------|--------------|-----------------|
| Log trademarks    | 0.00857***    | 0.0112            | 0.000476     | 0.00562         |
|                   | (0.00297)     | (0.0102)          | (0.00468)    | (0.00688)       |
| Constant          | −0.647***     | −1.422***         | −0.486***    | −1.290***       |
|                   | (0.0604)      | (0.208)           | (0.0951)     | (0.140)         |
| Observations      | 997           | 996               | 996          | 995             |
| R-squared         | 0.688         | 0.706             | 0.664        | 0.748           |
| State FE          | X             | X                 | X            | X               |
| Time FE           | X             | X                 | X            | X               |
| Trends            | X             | X                 | X            | X               |

*Note:* SE clustered by state.

***p < .01; **p < .05; *p < .1.
2 in the table). This means that more technological innovation leads to a higher share of conditions in economic regulation. Moreover, I do not find any difference between conditions and non-conditions for social provisions (Columns 3 and 4 in the table, respectively): either type is not statistically related to technological change. This finding further supports the expectations set out above, excluding potential alternative explanations that might link technological innovation and contingent regulation. One might argue that, for instance, technology makes it easier for regulators to draft, monitor, and implement more complex regulation and this might explain the increase in the number of contingent clauses. Theoretically speaking that might be true, but I can exclude this as the main explanation for what I find in the analysis above. Indeed, if this alternative explanation was true, I would see an effect on contingent clauses of all the topics, not only economic topics.

Table 4 shows the effect of regressing the share of contingent economic provisions on technological change and its lead and lag effects (Columns 1 and 2, respectively). Only the contemporaneous effect is statistically significant, suggesting that future or past values of technological change do not have any effect on regulation. In conclusion, I find support for the empirical expectation set out above: technological innovation leads to more contingent clauses regulating the economy.

### Limitations and Opportunities for Future Research

We have seen that the approach presented in this work, despite its simplicity, provides promising results. Yet, there are some broad limitations to this approach. First of all, the part above shows an off-the-shelf application of the parser, where little coding is needed. Yet, in order to get results in a user-friendly format, such as those shown above, some coding is needed. For instance, some coding is needed to extract together all the tokens belonging to the different branches of the parse tree.

| Variables           | (1)                      | (2)                      |
|---------------------|--------------------------|--------------------------|
| Cond Econ           | Cond Econ                |
| Log trademarks      | 0.00757** (0.00298)       | 0.00858** (0.00337)      |
| Lead log trademarks | 0.00319 (0.00329)        |                          |
| Lag log trademarks  |                          | 0.00287 (0.00351)        |
| Constant            | -0.691*** (0.0748)       | -0.707*** (0.126)        |
| Observations        | 947                      | 947                      |
| R-squared           | 0.684                    | 0.697                    |
| State FE            | X                        | X                        |
| Time FE             | X                        | X                        |
| Trends              | X                        | X                        |

*Note: SE clustered by state.*

---

***p < .01; **p < .05; *p < .1."
and obtain, for instance, all the tokens in the main sentence, namely the sentence containing the head verb, and all the tokens in the Object branch.

Second, one of the main limitations of the approach presented here relates to co-referencing, which is the use of a pronoun to refer to something said before. For instance, if we were to find the sentences in Figures 1 and 2 one right after the other in a piece of legislation, they might look like: “Fish Health Board must constitute a public entity. It must meet at least once a year”. The pronoun “it” clearly refers to “Fish Health Board”, but the parser might struggle to detect this, especially if the two sentences are far apart in the text. This is something not easy to fix, but it should be noticed that the use of co-referencing is much less common in formal legislation than, for instance, in more informal media such as newspaper articles. Relatedly, certain elements of the grammar might be implicit. For instance, a provision might refer to a condition that is explicit in another part of the text or that is assumed by the legislators. Under this light, the approach presented in this paper extracts the explicit conditions in a legal text.

Third, although the results are promising, this approach generates some false positives and false negatives. This can be overcome in the future by blending different approaches. The approach used in this paper is a fully unsupervised approach based on computational linguistics. In the future, researchers should consider supervised approaches like machine learning. For instance, Anastasopoulos and Bertelli (2020) propose a machine learning approach to extract from legal texts those sentences that delegate powers. There are also possibilities to combine the approach proposed here and machine learning. The syntactic parser SpaCy allows researchers to train parsers for custom semantics. This approach is particularly promising, as it combines the flexibility of allowing the algorithm to learn from new information and the reliability of already constructed ontologies with clear extraction rules based on syntactic features.

Moreover, it should be noticed that the approach proposed in this paper and its potential developments discussed above are fully compatible with the new approaches to the grammar, such as the nested grammar (nADICO) (Frantz et al., 2013) and the Grammar 2.0 (Frantz & Siddiki, 2020). Indeed, syntactic parsing allows identifying the main part of a provision (the head) and its dependents (the children) and the different types of relationships between these two, as seen above. Visualizing nested elements of the grammar on a parse tree proves particularly useful. Moreover, unsupervised computational linguistics tools like those used here represent a good starting point for the application of new versions of the grammar that distinguish between different levels of abstraction. These tools produce very coarse results, such as what words/phrases represent (whether a noun, a verb, etc) and how they are related to one another syntactically (whether they are subject-verb, verb-object, etc).

Finally, other text analysis tools can be used to improve the study of the grammar, such as named-entity recognition (NER) and topic modeling. Named-entity recognition, already in use in political economy (Shaffer, 2020), can identify entities, such as time/date, events, locations, and organizations, without supervision. For instance, if we take the provision “Fish Health Board must elect a Chairman and
Vannoni: A Political Economy Approach to the Grammar

Vice-Chairman annually” from Siddiki et al., (2012), the SpaCy NER tool identifies two entities: “Fish Health Board” as an organization and “annually” as a date. If this information is combined with that obtained with the syntactic parser, we will find that the Attribute is an organization and the Condition is a time condition. This is very helpful for those approaches that seek to differentiate between types of attributes, objects, and conditions, such as the Grammar 2.0 (Frantz & Siddiki, 2020). Complementing the syntactic information of a phrase with its semantics/meaning can also be achieved by using topic modeling. The latter will allow categorizing elements of the grammar according to their content: for instance, categorizing conditions according to their type (time, space, and so on). Alternatively, topic modeling can be used as done above to categorize provisions according to their scope of application: whether they apply to the economy, the government, and so on.

Conclusion

The validation exercises performed above show promising results, with the off-the-shelf application of syntactic dependency parsing performing relatively well on all the elements of the grammar, including those that have been proved so far most difficult to code, such as conditions. Although there are several limitations to this application, this method is very flexible and hence can be complemented with machine learning, by creating custom semantics, for instance. Moreover, given the granularity of its output, this method is suitable for more advanced approaches to the grammar that look at different levels of abstraction. The approach proposed in this work entails that the study of institutions using the grammar can now be scaled up to large volumes of legislative texts, thus opening up new research opportunities.

The contribution of this paper is also theoretical. Indeed, this paper places itself in the new strand of the “political economy of public policy” (John, 2018), which calls for closer integration between economics and political science in public policy. In this new strand, the outcome variable is the usual one in public policy, namely policy change, but the theories and methods are different and draw mainly from economics (John, 2018). The current work on the grammar, indeed, already brings attention to institutions in the study of public policy. Yet, this paper calls for more attention to the costs and benefits of establishing and reforming regulation and what drives these costs and benefits, in order to derive testable hypotheses on the causes and consequences of institutions.

William D. Schreckhise, Lecturer in Public Policy, Department of Political Economy, King’s College London.

Notes

1. We have already seen that there is little analytical point in distinguishing between rules and norms when studying legislation.
2. Available at: spacy.io.
3. The full list of universal POS tags is here: https://universaldependencies.org/u/pos/.
4. For a user-friendly application see: https://explosion.ai/demos/displacy.
5. This paper contains provisions from aquaculture regulations in Colorado.
6. See: https://framenet.icsi.berkeley.edu/fndrupal/about.
7. The only part that requires some minor coding is making sure the algorithm outputs parse tree branches and not individual tokens, which will be discussed below.
8. This dataset contains provisions from the National Organic Program (NOP), with regulations for the production, trade, and labeling of US Department of Agriculture (USDA) organic products. These provisions are manually coded according to the grammar of institutions, namely for every provision the coders annotate the Attribute, the aim, the object, and so on.
9. This function gives a similarity score from 0 to 100. For each category, I set the value of this score that qualifies a ‘match’ by checking a subsample of results manually. This function is used to overcome some differences in the output of the coding. Take the full statement: “Any agreement reached during or as a result of the mediation process shall be in compliance with the Act and the regulations in this part”. The manual coding gives as Condition “[how] with the Act and the regulations in this part”, whereas the machine coding gives “in’, ‘compliance’, ‘with’, ‘the’, ‘act’, ‘and’, ‘the’, ‘regulation’, ‘in’, ‘this’, ‘part”’. The mismatch is mainly due to the fact that the code extracts “in compliance” as part of the condition, whereas the manual coding does not.
10. For more information on the dataset, see Vannoni et al., (2020) and Ash (2016).
11. For a full list of entities see: https://spacy.io/api/annotation#section-named-entities.

References

Anastasopoulos, L. Jason, and Anthony M. Bertelli. 2020. “Understanding Delegation Through Machine Learning: A Method and Application to the European Union.” American Political Science Review 114 (1): 291–301.

Ash, Elliott. 2016. The Political Economy of Tax Laws in the US States. In Dondena Seminar. Milan, IT: Bocconi University.

Ash, Elliott, Massimo Morelli, and Matia Vannoni. 2020. “More Laws, More Growth? Evidence from U.S. States.” CEPR Discussion Paper DP15629. http://cepr.org/active/publications/discussion_papers/dp.php?dpno=15629.

Basurto, Xavier, Gordon Kingsley, Kelly McQueen, Mshadoni Smith, and Christopher M. Weible. 2010. “A Systematic Approach to Institutional Analysis: Applying Crawford and Ostrom’s Grammar.” Political Research Quarterly 63 (3): 523–37.

Battigalli, Pierpaolo, and Giovanni Maggi. 2002. “Rigidity, Discretion, and the Costs of Writing Contracts.” American Economic Review 92 (4): 798–817.

———. 2008. “Costly Contracting in a Long-Term Relationship.” The RAND Journal of Economics 39 (2): 352–77.

Carter, David, Sara Chonaiew, Christopher Weible, Saba Siddiki, Xavier Basurto, and John Brett. 2016. Institutionally Coded USDA Organic Regulations. Unpublished raw data.

Choi, Jinho D., Joel Tetreault, and Amanda Stent. 2015. “It Depends: Dependency Parser Comparison Using a Web-based Evaluation Tool.” Paper presented at the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

Crawford, Sue E., and Elinor Ostrom. 1995. “A Grammar of Institutions.” American Political Science Review 89 (3): 582–600.

Demsetz, Harold. 1967. “Toward a Theory of Property Rights.” American Economic Review 57 (2): 347–59.

Dunlop, Claire A., Jonathan C. Kamkhaji, and Claudio M. Radaelli. 2019. “A Sleeping Giant Awakes? The Rise of the Institutional Grammar Tool (IGT) in Policy Research.” Journal of Chinese Governance 4 (2): 163–80.
Frantz, Christopher, Martin K. Purvis, Mariusz Nowostawski, and Bastin Tony Roy Savarimuthu. 2013. “nADICO: A Nested Grammar of Institutions.” Paper presented at the International Conference on Principles and Practice of Multi-Agent Systems.

Frantz, Christopher K., and Saba Siddiki. 2020. “Institutional Grammar 2.0: A Refined Specification for Encoding and Analyzing Institutional Design.” Working paper.

Goldberg, Yoav, and Joakim Nivre. 2012. “A Dynamic Oracle for Arc-Eager Dependency Parsing.” Paper presented at the Proceedings of COLING 2012.

Gotsch, Matthias, and Christiane Hipp. 2014. “Using Trademarks to Measure Innovation in Knowledge-Intensive Business Services.” Technology Innovation Management Review 4 (5): 18–30.

Honnibal, Matthew, Yoav Goldberg, and Mark Johnson. 2013. “A Non-Monotonic Arc-Eager Transition System for Dependency Parsing.” Paper presented at the Proceedings of the Seventeenth Conference on Computational Natural Language Learning.

John, Peter. 2018. “Theories of Policy Change and Variation Reconsidered: A Prospectus for the Political Economy of Public Policy.” Policy Sciences 51 (1): 1–16. https://doi.org/10.1007/s11077-017-9297-x.

Jurasky, Daniel, and James H. Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Upper Saddle River, NJ: Prentice Hall.

McGinnis, Michael D. 2011. “An Introduction to IAD and the Language of the Ostrom Workshop: A Simple Guide to a Complex Framework.” Policy Studies Journal 39 (1): 169–83.

Mulligan, Casey B., and Andrei Shleifer. 2005. “The Extent of the Market and the Supply of Regulation.” The Quarterly Journal of Economics 120 (4): 1445–73.

O’Connor, Brendan, Brandon M. Stewart, and Noah A. Smith. 2013. “Learning to Extract International Relations from Political Context.” Paper presented at the Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Ostrom, Elinor. 2009. Understanding Institutional Diversity. Princeton, NJ: Princeton University Press.

Shaffer, Robert 2020. “Power in Text: Implementing Networks and Institutional Complexity in American Law.” Working Paper.

Siddiki, Saba. 2014. “Assessing Policy Design and Interpretation: An Institutions-Based Analysis in the Context of Aquaculture in Florida and Virginia, United States.” Review of Policy Research 31 (4): 281–303.

Siddiki, Saba, Xavier Basurto, and Christopher M. Weible. 2012. “Using the Institutional Grammar Tool to Understand Regulatory Compliance: The Case of Colorado Aquaculture.” Regulation & Governance 6 (2): 167–88.

Siddiki, Saba, Tanya Heikkila, Christopher M. Weible, Raul Pacheco-Vega, David Carter, Cali Curley, Aaron Deslatte, and Abby Bennett. 2019. “Institutional Analysis with the Institutional Grammar.” Policy Studies Journal. https://doi.org/10.1111/psj.12361.

Siddiki, Saba, Christopher M. Weible, Xavier Basurto, and John Calanni. 2011. “Dissecting Policy Designs: An Application of the Institutional Grammar Tool.” Policy Studies Journal 39 (1): 79–103.

Van Atteveldt, Wouter, Jan Kleinnijenhuis, and Nel Ruigrok. 2008. “Parsing, Semantic Networks, and Political Authority Using Syntactic Analysis to Extract Semantic Relations from Dutch Newspaper Articles.” Political Analysis 16 (4): 428–46.

Vannoni, Matia, Elliott Ash, and Massimo Morelli. 2020. “Measuring Discretion and Delegation in Legislative Texts: Methods and Application to US States.” Political Analysis 29 (1): 43–57.

Weible, Christopher M., and David P. Carter. 2015. “The Composition of Policy Change: Comparing Colorado’s 1977 and 2006 Smoking Bans.” Policy Sciences 48 (2): 207–31.