Raw Ontological Model Driven by Institutional Grammar – Attempt To Introduce A New Subclass of Ontologies for Policy Design Studies ⋆

Anna Wróblewska[1][0000−0002−3407−7570], Bartosz Pieliński[2][0000−0003−2664−2135], Karolina Seweryn[1][0000−0003−0617−7301], Sylwia Sysko-Romańczuk[1][0000−0001−5960−8131], Stanisław Giziński[1][0000−0002−3695−9809], Karol Saputa[1][0000−0002−8809−5248], and Aleksandra Wichrowska[1]

[1] Warsaw University of Technology, Warsaw, Poland
{anna.wroblewska1}@pw.edu.pl
[2] University of Warsaw,
Faculty of Political Science and International Studies, Warsaw, Poland
{b.pielinski}@uw.edu.pl

Abstract. Currently, there are high expectations regarding ontologies developed on legal documents. They mainly refer to their coherence and completeness to support human-professional experts in legal reasoning. We challenge this status quo making a fresh attempt to develop legal ontologies for a different purpose and target group – assisting political scientists in their research studies of policy design. Scientists are interested in comparative studies of different types of public policies. To achieve this goal, they expect to use ontologies as analytical tools. We call them "raw ontologies" because ontologies for policy design studies do not have to meet the criteria of ontologies for legal reasoning.

Therefore, our study introduces a concept of a new class of ontologies – raw ontologies – and automated methods for developing them. Raw ontologies assist researchers in studying specific public policies as their machine-readable representations. They pick up features of public policies from legal documents separately reflecting all their relations and inconsistencies. Political scientists using proposed semi-automated methods can relatively quickly develop these ontologies. The methods rely heavily on Institutional Grammar (IG) – a universal schema for annotating any social rules set, including legal regulations. We used IG as an intermediate layer between a legal text and a raw ontology. It helped to identify raw ontologies' classes, their attributes, relations among them, and their axioms and other rules.

We present our results in the form of Raw Ontological Model Driven by Institutional Grammar together with semi-automatic procedures for (1) automatically pre-annotating (tagging) documents with IG, (2) providing an exemplar raw ontological model for the information contained

⋆ Research was funded by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme (grant no 1820/27/Z01/POB2/2021).

in legal acts. We adopted our approach to create the first raw ontology based on the Emergency Paid Sick Leave Act enacted in the USA in reaction to the COVID-19 pandemic.
The process of manual annotation with IG (described in Section 2.2) was enriched with automatic tools: (1) pre-processing (identification of regulative and constitutive statements), (2) IG Tagger (as a pre-annotation stage with IG tags), (3) Raw Ontology Builder (a converter from IG annotations to policy ontology). We also elaborated quality check and super-annotations with an expert in policy design. Finally, we consider policy design research questions which can be answered using raw ontology.

Keywords: Ontology Modeling · Natural Language Processing (NLP) · Digital Humanities · Policy Design · Institutional Grammar

1 Introduction

Not all branches of science fully adapt themselves to possibilities that have been created by the Big Data. One of such branches is political science, especially its section devoted to public policy studies. There are now available multiplicity of data sources, including legal acts, which can be analyzed by researchers. However, they lack proper tools allowing them to organized large quantity of data and to analyze it. Our research experiment aims to meet demands coming from public policy studies for these kind of tools by creating 'raw ontologies' – an analytical tools that allows to organized data coming from large number of regulations. The idea and practice behind raw ontologies are inspired on the one hand by ontologies created for legal reasoning on the other hand by Institutional Grammar an analytical tool developed in political science [9, 5]. Three aspects distinguish this study from other similar ones: (1) a new target users' group – political scientists, (2) a new look at the role of an ontology – as a tool for policy design conceptualization, and (3) a new intermediate layer between a legal text and an ontology build from that text – IG usage for extracting and debugging knowledge gained from legal documents [9].

The point of departure for creating raw ontologies was an observation that although IG has been meeting with a growing interest in political science [24] there are problems with its application in wider range of research. The methodology behind raw ontology has been created to take advantage of previous research on the IG development, as well as, make this method easier to apply for studding large number of legal acts. As regards ontologies, they seems to be very useful for arrange data coming from legal acts via IG provided that more 'liberal' than usual approach will be able for their development. We believe that this nuance related to our new take on the use of ontologies is not even picked up by Palmirani [20].

Hence, having prior experience in IG applications, we assumed that it could be exploited as a layer between the policy regulations written in a natural language text and a semantic model of these regulations. We introduced IG as an additional annotation layer to recreate policy designs and regulations in a computer-readable form of ontologies. IG's role in our experiment was to state as an intermediary between the raw regulation texts and ontology model.

One of the urgent research topics for IG's analysts and adopters is governments' responses to the global crisis caused by the COVID-19 pandemic. Studying public and social policies, political scientists' primary purpose of analyzing how the COVID-19-influenced regulations impact social rights globally. We adopted our approach to create the first raw ontology based on the Emergency Paid Sick Leave Act enacted in the USA in reaction to the COVID-19 pandemic.

Finally, the introduction of computer science techniques has promising potential to improve legal document processing significantly. It would allow building extensive databases of IG annotated legal documents, more easily compare the design of different public policies and action situations associated with them.

Our goal was to facilitate the automatic process of annotating and analyzing legal documents by researchers in policy design. Our efforts comprise semi-automatic procedures for:

1. a raw ontological model driven by Institutional Grammar,
2. an algorithm for pre-annotating (tagging) documents with Institutional Grammar,
3. an algorithm for building our raw ontological model from regulation texts,
4. providing an exemplar raw ontological model for our use case – the Emergency Paid Sick Leave Act enacted in the USA in reaction to the COVID-19 pandemic.

Moreover, for each automatic algorithm, we provide a quality check of its results done by professional political science expert.

We organized this paper as follows. Section 2 briefly describes existing solutions regarding ontology modeling , especially in legal field, Institutional Grammar used in in policy design, and ontology learning methods. Our approach is described in Section 3, where we started from reasons why we used IG in our research framework. Section 4 shows our use case and experiments with our approach. Finally, we discuss and summarize our methodology in Section 5.

2 Related Works

An ontology is defined as a conceptual representation of the entities, their properties, and relationships in a domain [27]. Ontologies have the potential for playing a mattering role in making artificial intelligence (AI) systems explainable. They provide a user's domain conceptualization, in turn data model is used for explaining or debugging the process (finding ambiguities) of analyzing legal acts [27].

In this section, we analyzed available research frameworks for the semantic description of institutional policy design. So far, many legal ontologies have been designed, as well as Institutional Grammar (IG) framework. Moreover, we analyzed and described sources and semi-automatic methods based on parsing natural language to gather and extract knowledge from texts to a model in a semi-structured way, i.e., IG or other ontologies.

2.1 Legal-based Ontologies

There are a varied best practices and understanding of building ontologies in general [16] as well as legal ontologies [20, 7, 13]. Theoretical foundations for the transfer of law concepts and rules to ontology were discussed in [2]. There are very sophisticated methodologies for engineering legal ontologies – to mention only MeLON [20] and other promising attempts to automatize the process of generating legal ontologies [7]. Ontologies are usually seen as formal, explicit specifications of shared conceptualizations, that should express a shared view of a particular system [25, 12]. Therefore, their creation process is usually very collaborative in its nature and involves different stakeholders whose actions are affected by a system expressed through ontology [16]. In the case of legal ontologies, they have to consider not only an initial set of legal texts but also their judicial interpretations and their understanding by users of regulations. They are expected to support legal reasoning, to assist legal practitioners in making real-life decisions based on legal regulations.

Ontology review-driven findings for our motivation to a new approach design: (1) the common characteristic of the research studies is to treat ontologies as tools for building knowledge representations; (2) the potential for scaling automatization of ontologies engineering, including legal ones, is very limited, and (3) current process of ontologies creation is collaborative and labor-intensive.

Our study aims to simplify the process of creating ontologies from legal texts. Thus, it intends to increase the potential for scaling their applications up by changing its goal – no longer are they expected to express a wildly share view of a particular legal system. Such new subclass of ontologies we named "raw ontologies". They are so-called non-collaborative ontologies [11, 26] and, on the theoretical level, they do not need to express a shared view of the particular system. These characteristic raises a question about the sufficient level of shared understanding of a system: is high enough to build an ontology around it? Usually, it is expected to create a formal ontology only under condition a significant number of text sources and stakeholders agree on system features. However, an ontology could be built around a limited number of sources, but significant in their importance for the particular system.

2.2 Institutional Grammar

S. Crawford and E. Ostrom created Institutional Grammar (IG) in 1995 to solve discussion regarding one of the crucial issues in social science – the nature of institution [4], more specifically, how institutions regulate human behavior. The aim was to make the analytical level possible to distinct strategies, norms, and rules. These three forms of institutional statements regulate behavior through different means: advice, expectations, and threats of penalty. Also, each institutional statement has, and this is the central assumption behind IG, a different syntactic structure: ontological structure in semantic terminology. IG contributes to developing the Institutional Analysis and Development (IAD) framework [19]. Nowadays, IG is less linked to the discussions regarding the nature of institutions, and more to the development of research on public policy design [24]. Moreover, IG concept has gained some interest between computer scientists interested in agent-based modeling [10]. So, it has been extended to include new syntax elements as well as new types of statements. The new version of IG is referred to as IG 2.0 [9] and used in our approach.

Policy design is "concerned with the construction of public policies" [23, p. 1] and with identifying major policy actors (states, private companies, individuals) and relations between them. Institutional Grammar (IG) has been used to analyze public policies mainly from legal regulations [9]. Legal regulations are considered here to make a reference to the ontology engineering, shared enough views of public policies to treat them as representations of those policies. IG is very good at extracting crucial information about public policies based on legal texts. However, as a relatively new tool, it has not yet developed the analytical infrastructure associated with it. Here the ontology engineering lends a hand – it provides analytical support coming from computer science. Raw ontologies can be seen from this perspective as a valuable concept for both computer and political sciences. By their reference to IG, they show that ontologies created only based on legal texts expressed shared enough view of a particular policy system and, at the same time, offer interesting analytical tools for studying policy design.An example of the use of IG as an analytical tool is presented below.

IG aims to extract information on policy design from a written regulation. The design, referred in IG to as action, consists of actors defined by constitutive statements – *Local employment office is responsible for provision of the COVID-19 benefit* – and rules regulating their behavior (regulative statements) – *The unemployed have to visit the local employment office at least once a week.* The standardized procedure for working with a legal text using IG consists of a few rudimentary elements:

1. Selecting a legal text;
2. Selecting parts of legal text relevant for research and suited for IG annotation;
3. Identifying individual sentences;
4. Extracting from each sentences atomic statements – statements that have structure of constitutive or regulative statement and do not have IG components containing multiple values;
5. Identifying statements – observations – that do not constitute or regulate any aspect of policy design but describe circumstances when other statements should be applied;
6. Annotating IG components in each atomic statement.

It is possible to reveal the detailed structure of an action by studying the statements, their relations, and statements' components and their relations (see Table 2.2).

Table 1. IG main components depending on statement type (regulative or constitutive) based on [9, pp. 10-11].

Regulative statements	Description	Constitutive statements	Description
Attribute	The addressee of the statement.	Constituted Entity	The entity being defined.
Aim	The action of addressee regulated by the statement.	Constitutive Function	A verb used to define Constituted Entity.
Deontic	An operator determining level of discretion or constraint associated with Aim.	Modal	An operator determining level of necessity and possibility of defining Constituted Entity.
Object	The receiver of the action described by Aim	Constituting Properties.	The entity against which Constituted Entity is defined.
Activation Condition	The setting to which the statements applies.	Activation Condition	The setting to which the statements applies.
Execution Constraint	Quality of action described by Aim	Execution Constraint	Quality of Constitutive Function.

There were two challenges with IG, which we decided to tackle in our research experiment: the lack of well-developed analytical support and necessity to use long and complicated manual annotation of legal documents. The IG implementation is exceptionally labor-intensive and negatively affects work costs and time. The automatization of raw ontologies engineering addresses both issues linked to IG development in political science,

2.3 Semi-Automatic Text-Based Methods for Ontology Engineering

In [27], the author presented the current state-of-the-art of ontology engineering. She described three main challenges in this domain: (1) conceptualization – patterns and templates, (2) better tooling in automatic support for building ontologies and communicating among them, and (3) complexity of the languages and tooling, which makes it hard to align among ontologies. Our experiment concentrates on conceptualization and automatic support for building ontologies in institutional policy design.

Ontology-Text Layers The process of direct conceptualization and disambiguation of natural language texts is arduous. So far, there have been a few research building an intermediate layer between texts and ontologies. They are OntoLex-lemon – a lexical model expressing the linguistic structure of texts and its link to ontology itself [14, 3, 8], or a more superficial layer described in [21]. The authors of these studies underlined the need for intermediate layers between natural language texts and ontologies to ease the use of ontologies built based on text resources. These approaches also help in maintaining different terminologies and synonyms in texts. To the best of our knowledge, there are no studies of domain-specific intermediate layers between natural language text, e.g., legal regulations in our case, and an ontological domain model.

Ontology Learning and Population There is extensive research on building ontologies directly from natural language texts in different domains [27]. Usually, the methods are domain-specific or very general, making them hard or sometimes even impossible to adapt to specific domains and texts. In [1], the author identified four categories of methods for ontology learning and their specific tasks. These are: (1) linguistic-based approaches for term and concept extraction, concept hierarchy and relation discovery, and machine learning-based approaches, i.e., (2) statistic-based approaches for the same tasks and synonym discovery, (3) logic-based approaches for concept hierarchy, relations and axioms discovery, (4) deep learning tasks and methods.

Many of the methods and tools are based on syntactic dependency parsing of a linguistic sentence structure, regular expressions, and other rule-based algorithms. Thus they are usually language- and domain-specific. Some techniques are also based on named-entity recognition machine learning models [28]. However, usually, they are very specific task-oriented and require an extensive dataset with annotations of the entities and their relations.

Another approach is to use also structured information like databases, not only unstructured texts. One of these approaches to automatic legal ontology modeling is described in [13]. The authors extract essential terms from a database of the legal act in the Chinese language.

Kurcheeva et al. [17] presented in 2019 challenges related to building legal ontologies from texts and provides applications of such ontologies in public administration. For a use case of housing legislation, an analysis of concepts, e.g., "housing," "living premises," was prepared by domain experts based on natural language and used to build an ontology. Several applications of this methodology are listed, such as the identification of gaps and collisions in legislation. However, this complex analysis is developed as manual work.

The population of ontologies with entities from the text was described by Faria et al. [6]. The general framework of populating ontologies based on linguistic tagging, named entity recognition, and co-reference identification is presented, and a description of other NLP methods used for this task.

3 Our Approach

The work of analyzing legal acts has been challenging and error-prone so far. Political scientists manually assembled documents, annotated them with IG manually, and then analyzed them. Our goal is to provide a tool that will facilitate and at least partially automate their work.

In our study, we use IG as a text-semantic bridge. On one side, IG is a tool for describing and analyzing relations between institutional actors in a more systematic and structured way than natural language texts. We used definitions of IG for tagging. However, there are still an open-issues and need to define more precisely some IG statements, e.g., context data like Activation Conditions enitites. IG is not structuralized as a formal ontology, not precise, with classes and logical rules between them. Such structural ordered and precisely defined data (an ontology) further can be used, for example, for defining rules of institutions and network analysis and reasoning [18].

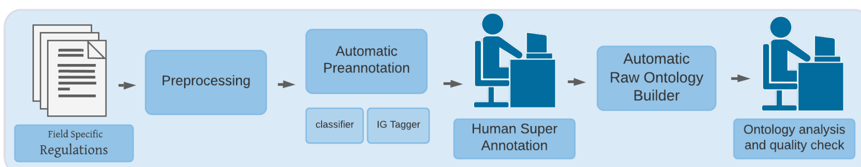

Fig. 1. The diagram of the system's pipeline.

The process of manual annotation with IG (described in Section 2.2) was enriched with automatic tools: (1) pre-processing (identification of regulative and constitutive statements), (2) IG Tagger (as a pre-annotation stage with IG tags), (3) Raw Ontology Builder (a converter from IG annotations to policy ontology). We also elaborated quality check and super-annotations with an expert in policy design. Finally, we consider policy design research questions which can be answered using raw ontology.

Tools, datasets, and algorithms described in the following sections are accessible on the paper's GitHub page[3] and more details on our algorithms in supplementary materials (also attached there).

3.1 Statements Pre-processing

Constitutive statements in IG are used for defining purposes, and regulative describe how to regulate behaviours or actions that can be done by actors. We

[3] https://github.com/institutional-grammar-pl/Raw-Ontological-Model-Driven-by-Institutional-Grammar

trained a classifier based on TFI-DF with 50 the most common words and Random Forest to distinguish between these types. The model's AUC is equal to 0.94, F1-score is 0.93[4].

Then, we built the IG tagger consisting of rules specific to mentioned types of sentences. It tags sentences based on their characteristics. Similarly, our Ontology Builder also follows these distinction, adding also observations for constitutive and regulative statements. Because they express additional conditions for statements (see Section 2.2).

3.2 IG Tagger

In the first stage of tagging, each word in the statement gets annotation containing lemma, part of speech tag, morphological features, and relation to other words (StanfordNLP package [22]). Because of different IG tags in regulative and constitutive statements, automatic tagger has two dedicated algorithms, respectively. These algorithms are based on sets of rules dedicated for each type of statement. In our supplement, we provided all particular algorithms for IG tagger.

Here, we show an example of its usage. For this purpose, below we present only the selected rules that are used in the analyzed sentence.

1. If a sentence contains one word with *root* tag and this word is a verb or an adjective:
   (a) If the word founded in 1 is a verb, then annotate it as *constitutive function* , otherwise as *constituing properties.*
   (b) If the word annotated in 1a has a child with *aux:pass* or *cop* relation, then annotate this child as *constitutive function.*
   (c) If the word annotated in 1a has a child with one of *nsubj*, *nsubj:pass* or *expl* relation, then annotate this child as *constituted entity.*
   (d) If the word annotated in 1a has a child with one of *nsubj*, *nsubj:pass* or *expl* relation, then annotate this child as *constituted entity.*
   (e) If the word annotated in 1d has a child with one of *det*, *compound*, *mark*, then annotate this child and all child's descendant as *constituted entity.*
   (f) If the word annotated in 1a has a child with one of *obl*, *advmod*, *xcomp* relation, then annotate this child and all child's descendants as *context.*

Considering the sentence given in Figure 2, our tagging algorithm takes into account the following rules:

- According to the rule 1a we annotate the word *unable* as *constituting properties.*
- According to the rule 1b we annotate the word *is* as *constitutive function.*
- According to the rules 1c and 1e we annotate words *the employee* as *constituted entity.*
- According to the rule 1f we annotate words *to work* as *context.*

[4] The code with the classification experiment is available in our GitHub repository

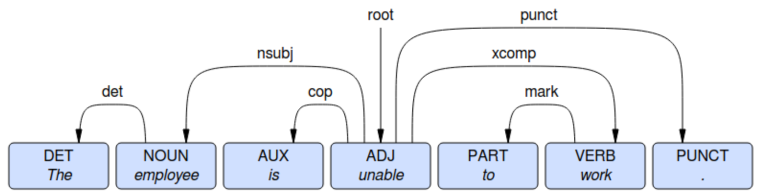

Fig. 2. Visualization of annotated constitutive sentence by Stanford NLP tagger in CONLL-U Viewer. This statment example in described in the text in paragraph **??**.

3.3 Raw Ontology Builder

Our automatic method for building ontologies from regulation texts annotated with IG tags comprises the following steps: (1) indicating classes and their hierarchy, (2) extracting relations, (3) defining class axioms and SWRL rules. In this process, we treat differently statements which are observations – we call them *observation regulative/constitutive* – and those which are not – we call them *proper constitutive/regulative* statements. From observation constitutive statements, we extract class hierarchy. From each observation regulative statement, we extract 3 classes, their hierarchical relations and possible relationships between those classes – we call such relationships *observational relations*. From each proper constitutive and proper regulative statements, we extracted 3 classes hierarchies, relations which are regulated or constituted by those statements, and also from activation condition of those statements, we are able to define axioms and SWRL rules by which such relations are enforced. In our supplement, we define in more details how we prepare class names, their relations, axioms and rules.

Here, we give examples of transforming statements into ontology.

```
[Constituted Entities] = employee,
[Function] = is,
[Constituted Properties] = unable,
[Constituted Properties Property] = to work
Class: employee,
Subclass: employeeThatIsUnableToWork
```

Example rule created automatically is as follows:

```
EmployeeThatIsUnableToWork(?x), EmployeeEmployedByEmployer(?x),
    Employer(?y), PaidSickTime(?q) -> shall_provide(?y,?q),
    shall_be_provided_to(?q,?x)
```

Limitations and Further Development We were not able to model activation conditions that do not refer to a specific statement, which limited our possibility to automatically produce SWRL rules. To overcome this, IG annotation should be made much more fine-grained – dividing each non-referential activation condition into smaller statements. Secondly, we have not modelled any other referential tags than activation condition.

In future, it would be worth considering adding *class trimming* step, as the last phase of ontology building algorithm. Such step could use NLP tools such as word similarities to merge some classes with similar meaning.

3.4 Experimental Journey – Research in Action

The first need in our research journey was to automate the process of tagging legal acts with Institutional Grammar. The scientists had tried to automate their work using Excel or other annotation tools, e.g., INCEpTION [15]. Unfortunately, both tools have drawbacks that make the work challenging and lengthy. Especially, comparing annotations made by different annotators or even self-checking was hard. Thus, we design a hierarchical structure of sentences and visualizations to compare the annotations for the same statements[5].

During the work on manual annotating and also the automatic tagger, we had realized that IG is not precise. The IG is a new approach, and it is still evolving. Some entities in IG are too general and ambiguous. Thus, we need a more structuralized ontology to analyze legal regulations further, i.e., reasoning based on regulations, checking the consistency of the model, and further aligning different regulations among them. This study shows experiments based on a concrete use case – one text legal act, how we extract ontology from this act and check its quality.

4 Experimental Challenges

4.1 Testing Set – Our Use Case

Our approach was tested against a regulation linked to the COVID-19 pandemic - the Emergency Paid Sick Leave Act (H.R.6201, Division E) introduced in March 2020 in the USA [6]. This regulation was chosen because it is part of a more extensive data set of regulations on social policy reactions to the COVID-19 pandemic. Therefore it will be relatively easy to scale up the whole system of raw ontology engineering for more advanced research. Also, the regulation is relatively simple. It constitutes a limited number of policy actors and objects and creates a limited number of rules regulating their actions and relations. These qualities allow controlling the experimental development of automatizing raw ontologies carefully. The regulation was chosen also because it allows to introduce simple metrics for analysing characteristics of the ontology derived from it. Metrics that describe the access of employees in the USA to the paid sick leave – the major feature of this policy design.

[5] https://github.com/pw-mini-ig

[6] https://www.congress.gov/bill/116th-congress/house-bill/6201/text

4.2 *IG Tagger* Performance

IG Tagger has been evaluated on the legal act annotated by an expert. The prediction was performed on *atomic sentences* manually extracted from complex sentences. The regulation included 333 statements (98 regulative and 235 constitutive statements). Applied measures were determined based on the accuracy of the classification of the individual words in a sentence.

Table 2 shows metrics with a breakdown per component type and statement type. For the purposes of this analysis predicted and correct tags were mapped before the evaluation: (A, prop) to (A), (B, prop) to (B) for regulative ones, and (E, prop) to (E), (P, prop) to (P) for constitutive statements. Best results are achieved in recognizing Aim, Deontic and Modal components – over 80% of F1-score.

Table 2. Results of IG Tagger detailed on components.

Regulative Layer				**Constitutive Layer**			
Component	**F1 score**	**Precision**	**Recall**	**Component**	**F1 score**	**Precision**	**Recall**
Attribute	0.74	0.83	0.66	Entity	0.41	0.38	0.44
Object	0.54	0.69	0.45	Property	0.54	0.57	0.52
Deontic	0.93	0.94	0.92	Function	0.57	0.46	0.71
Aim	0.84	0.93	0.76	Modal	0.89	0.97	0.82
Context	0.57	0.44	0.82	Context	0.02	0.01	0.02
Overall	0.62	0.67	0.61	**Overall**	0.45	0.45	0.46

4.3 Ontology Usage

Equality of access to a resource As it was mentioned previously 4.1, the COVID-19 regulations was chosen to show how our raw ontology could be used to characterize a simple element of policy design – the access of employees to a resource – in this case to a paid sick leave during the COVID-19 pandemic. Using raw ontology we can measure the difference in access to the leave between different types of employees. To achieve aim, we define proxy metric of equality of access subclasses of specific class (employee) to specific resource. We define it as follow: $e \in E$ - specific subclass of employee, E - set of all subclasses, $r(e)$ - number of constitutive relations that can connect e to specific resource (e.g. to paid sick leave). If we define X as random variable with $P(X = r(e)) = \frac{r(e)}{|E|}$ then we can define equality of access of E to resource as $\eta(X)$, where η is Normalized Shannon Entropy defined as $\eta(X) = -\sum_{i=1}^{|X|} \frac{P(x_i)\log(P(x_i))}{\log(|X|)}$. Such metric, by computing it on different ontologies constructed from correspondent regulations (regarding a paid sick leave in other countries), can allow us to compare equality of access to simmilar resources accross different legislations.

Control over access to a resource It is possible to specify a role of a policy actor in access to a resource by studding SWRL rules. They inform us how often relations between policy actors (in our case employee) and a specific resource (in our case the paid sick leave) are controlled by other actor (in our case employee). It is possible to calculate the control index of an actor over access to a resource by dividing the number of rules regulating the access to the resource involving the controlling actor by the total number of rules regarding this particular resource.

4.4 Conclusions and Limitations of Our Approach

The driving premise of raw ontologies and automatic methods for building them is to help political scientists to take advantage of new opportunities: easy access via Internet to thousands of legal text. Our motivation was to create the tool that would allow them to study one specific aspect of the legal Big Data – policy design. Raw ontologies were created to allow extract from legal regulations relatively easily the major characteristic of policies they regulate. They bring possibility of expressing complex, multidimensional information coming from legal regulation in a one framework. Raw ontologies could be treated as computer readable representations of legal acts hence as representations of policy design build by these regulations. Our method of the automatization of raw ontology engineering creates potential for developing this new subclass of ontologies into significant number of regulations allowing their very effective comparison.

Our approach, being an experimental one, has few limitations. There is a need for more explicit articulation between previous takes on legal onotolgies and our approach. The most crucial unanswered question are: (1) the extend of 'rawness' acceptable for creating the subclass of ontologies; (2) the level of incoherence and vagueness acceptable in our approach; (3) the trade off between the roughness of our ontologies and their effectiveness in expressing major futures of legal regulations. Behind these questions hides another one, probably event more important – to what extend onotlogies, as we presently understand them in computer science, can be used as analytical tools. We are building case for treating ontologies not as tools which help different databases communicate with each other nor as tools for expressing coherent knowledge representations, but as means of creating computer readable representation of legal regulations with all their imperfections. Raw ontologies are treated as manifestation of legal regulation, and therefore: (1) their characteristics are treated as characteristic of regulations, and (2) metrics of their comparison are treated as metrics of legal texts/policy designs comparison. We aspire to at least encourage researchers to think about ontologies as potential source of interesting analytical tools. These ambitious goals, behind the research, have to be confronted with the limited scope of our use case. Our study was focus mainly on two aspects of raw ontologies: (1) developing the idea behind our new subclass of ontologies and (2) developing the automated process of their development. The use case lacks in raw ontologies measures, as well as, in comparative studies. Also a relative straightforward regulation was chosen for developing the first raw ontology. The policy design of paid sick leave is relatively simple – it consists of only a few actors and limited number of rules which could be expressed in the form of SWRL rules. Another limitation of our study is IG. It is still in the process of active development therefor not all aspect of its application are written in stone. Also, IG has not been developed for engineering ontologies. The compatibility between these two analitical tools is limited.

5 Further Research Directions

Our study show a potential for computer science and political science in taking a fresh look at ontologies. Computer scientists may start to see them not only as ways to allow communication between databases or expressing very coherent representations of a field specific knowledge, but as way of representing some aspect of reality with all its imperfections. Our research on raw ontologies is going to be developed in several direction. First of all, the linkage between IG and raw ontologies will be investigated. The very important question is to what extend the usage of IG for ontologes development enrages introduction of some changes into IG itself. Is it possible to make IG compatible with ontologies development. The second aspect of further research is to establish a minimal standards for raw ontologies. Their roughness does not means the complete lack of guideline in creating them. This issue relates to other – the human involvement in raw ontologies engineering. When and to what extend humans should be engaged in creating a new subclass of ontologies even though they should be 'raw' and engineered in the automated way?

The third issue related to raw ontologies advancement is the development of indicators which could measure characteristics of these ontologies relevant for policy design studies. Political scientists are interested in wide range of public policy characteristics: their inclusiveness, their transparency, their emphasis on equality or freedom, to mention the most prominent ones. The indicators should be developed that allow to follow this characteristic based on analysing raw ontologies threaded as representation of policy designs.

References

1. Al-Aswadi, F., Chan, H., Gan, K.: Automatic ontology construction from text: a review from shallow to deep learning trend. Artificial Intelligence Review **53**, 3901–3928 (2020). https://doi.org/10.1007/s10462-019-09782-9, https://doi.org/10.1007/s10462-019-09782-9
2. Breuker, J., Valente, A., Winkels, R.: Use and Reuse of Legal Ontologies in Knowledge Engineering and Information Management, pp. 36–64. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32253-5_4, https://doi.org/10.1007/978-3-540-32253-5_4
3. Chiarcos, C., Ionov, M., de Does, J., Depuydt, K., Khan, A.F., Stolk, S., Declerck, T., McCrae, J.P.: Modelling frequency and attestations for OntoLex-lemon. In: Proceedings of the 2020 Globalex Workshop on Linked Lexicography. pp. 1–9. European Language Resources Association, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.globalex-1.1

4. Crawford, S., Ostrom, E.: A Grammar of Institutions. The American Political Science Review **89**(3), 582–600 (1995)
5. Crawford, S.E.S., Ostrom, E.: A grammar of institutions. American Political Science Review **89**(3), 582–600 (1995). https://doi.org/10.2307/2082975
6. Faria, C., Serra, I., Girardi, R.: A domain-independent process for automatic ontology population from text. Science of Computer Programming **95**, 26–43 (2014). https://doi.org/https://doi.org/10.1016/j.scico.2013.12.005, https://www.sciencedirect.com/science/article/pii/S0167642313003419, special Issue on Systems Development by Means of Semantic Technologies
7. Fawei, B., Pan, J.Z., Kollingbaum, M., Wyner, A.Z.: A Semi-automated Ontology Construction for Legal Question Answering. New Generation Computing **37**(4), 453–478 (2019). https://doi.org/10.1007/s00354-019-00070-2, https://doi.org/10.1007/s00354-019-00070-2
8. Fiorelli, M., Stellato, A., Lorenzetti, T., Turbati, A., Schmitz, P., Francesconi, E., Hajlaoui, N., Batouche, B.: Editing ontolex-lemon in vocbench 3. In: LREC. pp. 7194–7203 (2020), https://www.aclweb.org/anthology/2020.lrec-1.889/
9. Frantz, C.K., Siddiki, S.: Institutional grammar 2.0: A specification for encoding and analyzing institutional design. Public Administration **n/a**(n/a) (2020). https://doi.org/10.1111/padm.12719, http://dx.doi.org/10.1111/padm.12719
10. Frantz, C.K., Purvis, M.K., Savarimuthu, B.T.R., Nowostawski, M.: Modelling Dynamic Normative Understanding in Agent Societies. Scalable Computing: Practice and Experience **16**(4), 294–310 (jan 2016). https://doi.org/10.12694/scpe.v16i4.1128, http://link.springer.com/10.1007/978-3-319-13191-7_24 http://www.scpe.org/index.php/scpe/article/view/1128
11. Giunchiglia, F., Dutta, B., Maltese, V., Farazi, F.: A Facet-Based Methodology for the Construction of a Large-Scale Geospatial Ontology. Journal on Data Semantics **1**(1), 57–73 (2012). https://doi.org/10.1007/s13740-012-0005-x, https://doi.org/10.1007/s13740-012-0005-x
12. Guarino, N., Oberle, D., Staab, S.: What is an ontology? In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. pp. 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3_0, https://doi.org/10.1007/978-3-540-92673-3_0
13. Hwang, R.H., Hsueh, Y.L., Chang, Y.T.: Building a taiwan law ontology based on automatic legal definition extraction. Applied System Innovation **1**(3) (2018). https://doi.org/10.3390/asi1030022, https://www.mdpi.com/2571-5577/1/3/22
14. J. McCrae, G. Aguado-de-Cea, P.B.e.a.: Interchanging lexical resources on the semantic web. Lang Resources & Evaluation **46**, 701–719 (2021). https://doi.org/https://doi.org/10.1007/s10579-012-9182-3
15. Klie, J.C., Bugert, M., Boullosa, B., Eckart de Castilho, R., Gurevych, I.: The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. pp. 5–9. Association for Computational Linguistics, Santa Fe, New Mexico (Aug 2018), https://www.aclweb.org/anthology/C18-2002
16. Kotis, K.I., Vouros, G.A., Spiliotopoulos, D.: Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations. The Knowledge Engineering Review **35**, e4 (2020). https://doi.org/10.1017/S0269888920000065
17. Kurcheeva, G., Rakhvalova, M., Rakhvalova, D., Bakaev, M.: Mining and indexing of legal natural language texts with domain and task ontology. In: Chugunov, A.,

Misnikov, Y., Roshchin, E., Trutnev, D. (eds.) Electronic Governance and Open Society: Challenges in Eurasia. pp. 123–137. Springer International Publishing, Cham (2019)
18. Olivier, T., Scott, T.A., Schlager, E.: Institutional design and complexity: Protocol network structure in response to different collective-action dilemmas. In: Fischer, M., Ingold, K. (eds.) Networks in Water Governance. pp. 267–293. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-46769-2_10, https://doi.org/10.1007/978-3-030-46769-2_10
19. Ostrom, E.: Understanding institutional diversity. Princeton paperbacks, Princeton University Press, Princeton (2005)
20. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal reasoning. In: Kő, A., Francesconi, E. (eds.) Electronic Government and the Information Systems Perspective. pp. 139–152. Springer International Publishing, Cham (2018)
21. Protaziuk, G., Wróblewska, A., Bembenik, R., Rybiński, H., Podsiadły-Marczykowska, T.: Lexical ontology layer – a bridge between text and concepts. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) Foundations of Intelligent Systems. pp. 162–171. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
22. Qi, P., Dozat, T., Zhang, Y., Manning, C.D.: Universal Dependency parsing from scratch. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 160–170. Association for Computational Linguistics, Brussels, Belgium (Oct 2018). https://doi.org/10.18653/v1/K18-2016, https://www.aclweb.org/anthology/K18-2016
23. Siddiki, S.: Understanding and Analyzing Public Policy Design. Elements in Public Policy, Cambridge University Press (2020). https://doi.org/10.1017/9781108666985
24. Siddiki, S., Heikkila, T., Weible, C.M., Pacheco-Vega, R., Carter, D., Curley, C., Deslatte, A., Bennett, A.: Institutional Analysis with the Institutional Grammar. Policy Studies Journal **18**(7), 643 (2019). https://doi.org/10.1111/psj.12361
25. Studer, R., Benjamins, R., Fensel, D.: Knowledge engineering: Principles and methods. Data Knowl. Eng. **25**, 161–197 (1998)
26. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology (otkm). In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. pp. 117–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_6, https://doi.org/10.1007/978-3-540-24750-0_6
27. Tudorache, T.: Ontology engineering: Current state, challenges, and future directions. Semantic Web **11**(1), 125–138 (2020). https://doi.org/10.3233/SW-190382
28. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., Sun, M.: How does NLP benefit legal system: A summary of legal artificial intelligence. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5218–5230. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.466, https://www.aclweb.org/anthology/2020.acl-main.466

Supplementary Materials for: Raw Ontological Model Driven by Institutional Grammar – Attempt To Introduce A New Subclass of Ontologies for Policy Design Studies

Anna Wróblewska[1[0000−0002−3407−7570]], Bartosz Pieliński[2[0000−0003−2664−2135]], Karolina Seweryn[1[0000−0003−0617−7301]], Sylwia Sysko-Romańczuk[1[0000−0001−5960−8131]], Stanisław Giziński[1[0000−0002−3695−9809]], Karol Saputa[1[0000−0002−8809−5248]], and Aleksandra Wichrowska[1]

[1] Warsaw University of Technology, Warsaw, Poland
{anna.wroblewska1}@pw.edu.pl
[2] University of Warsaw,
Faculty of Political Science and International Studies, Warsaw, Poland
{b.pielinski}@uw.edu.pl

This document is a supplement to our paper. Here we describe in more detail our algorithms. Section 1 lists our contributed resources. Then, we present an annotation file, also available in the resources (see Section 2). Section 4 and Section 5 describe in detail algorithms for automatic tagger of Institutional Grammar and Raw Ontology Builder, respectively.

1 Resources Description

The following software resources make up our workflow:

1. Legal Acts annotated with Institutional Grammar and examplary ontology of COVID-19 sick leaves
`https://github.com/institutional-grammar-pl/Raw-Ontological-Model-Driven-by-Institutional-Grammar`
2. Dataset – COVID-19 use case
`https://github.com/institutional-grammar-pl/ig-annotations`
3. Institutional Grammar tagger
`https://github.com/institutional-grammar-pl/policydemic-annotator`
4. IG-based ontology builder
`https://github.com/institutional-grammar-pl/ig-ontology-builder`

2 Annotation Format

Manual annotations of legal acts with Institutional Grammar are prepared in Excel spreadsheet format. The format is easy for manual use and collaboration. This columnar format is also easy to process by software. As an additional resource, we include the file with the IG annotations of the "COVID-19 seek leaves" act.

Section	Difficulty level (hard, medium, easy)	Statement function	IG syntax (regulative, constitutive)	Statement No.	Statement	Constituted Entity (Content)	Constituted Entity Property (Content)	Constituted Entity Property (Reference to statement)	Modal	Function	Constituted Properties
		observation	constitutive	22.3	Any person engaged in commerce that in the	person	AND[any|employs fewer than 500 employees]			engage	in commerce
		observation	constitutive	22.4	Any person engaged in commerce that in the	person	AND[any|employs 1 or more employees]			engage	in commerce
		observation	constitutive	22.5	Any person engaged in commerce that in the	person	AND[any|employs 1 or more employees]			engage	in commerce
		observation	constitutive	22.6	Any person engaged in any industry that in the	person	AND[any|employs fewer than 500 employees]			engage	in any industry
		observation	constitutive	22.7	Any person engaged in any industry that in the	person	AND[any|employs fewer than 500 employees]			engage	in any industry
		observation	constitutive	22.8	Any person engaged in any industry that in the	person	AND[any|employs 1 or more employees]			engage	in any industry
		observation	constitutive	22.9	Any person engaged in any industry that in the	person	AND[any|employs 1 or more employees]			engage	in any industry
		observation	constitutive	22.10	Any person engaged activity affecting	person	AND[any|employs fewer than 500 employees]			engage	activity
		observation	constitutive	22.11	Any person engaged activity affecting	person	AND[any|employs fewer than 500 employees]			engage	activity
		observation	constitutive	22.12	Any person engaged activity affecting	person	AND[any|employs 1 or more employees]			engage	activity
		observation	constitutive	22.13	Any person engaged activity affecting	person	AND[any|employs 1 or more employees]			engage	activity
		constitutive	constitutive	22.14	In subparagraph (A)(i)(I), the term ``covered	the term	``covered employer''			includes	

Fig. 1. A part of an annotated file with columns describing elements of sentence by Institutional Grammar.

3 Atomic Statement Classification

4 IG Tagger

The atomic institutional statements can be divided into two groups: regulative and constitutive. We built the tagger consisting of rules specific to mentioned types of sentences. It tags words based on their characteristics. Each word in the statement gets annotation containing lemma, part of speech tag, morphological features, and relation to other words (StanfordNLP package [2]).

Tags referring to each statement type are presented in our paper in Section on IG and Table 1.

Regulative statements	Description	Constitutive statements	Description
Attribute	The addressee of the statement.	Constituted Entity	The entity being defined.
Aim	The action of addressee regulated by the statement.	Constitutive Function	A verb used to define Constituted Entity.
Deontic	An operator determining level of discretion or constraint associated with Aim.	Modal	An operator determining level of necessity and possibility of defining Constituted Entity.
Object	The receiver of the action described by Aim	Constituting Properties.	The entity against which Constituted Entity is defined.
Activation Condition	The setting to which the statements applies.	Activation Condition	The setting to which the statements applies.
Execution Constraint	Quality of action described by Aim	Execution Constraint	Quality of Constitutive Function.

Table 1. IG main components depending on statement type (regulative or constitutive) based on [1, pp. 10-11].

4.1 Rules for Regulative Statements

Function *annotate_ all_ descendants(node, tag)* assigns *tag* to all descendants of *node*. Algorithms 1, 3, 2 shows rules of tagger to find aim, attribute, and deontic tag, respectively. Algorithm 4 (and points listed below) presents a way of annotating object and context.

Algorithm 1: AIM tagger.

```
Data: tree: LexicalTreeNode, root: tree.root, tags: list of tags
tags.append(AIM(root))
for child in root.children do
    if child.relation in ("aux:pass", "cop") then
        tags.append(AIM(child))
    if child.relation = "aux" and child.lemm in ("be", "have", "do") then
        tags.append(AIM(chid))
    if child.relation = "advmod" and child.lemm = "not" then
        tags.append(AIM(chid))
```

Algorithm 2: DEONTIC tagger.

```
Data: tree: LexicalTreeNode, root: tree.root, tags: list of tags
for child in root.children do
    if child.relation = "aux" and child.lemm in ("must", "should", "may",
      "might", "can", "could", "need", "ought", "shall") then
        tags.append(DEONTIC(child))
```

Algorithm 3: ATTRIBUTE and ATTRIBUTE_PROPERTY tagger.

```
Data: tree: LexicalTreeNode, root: tree.root, tags: list of tags
for child in tree.children do
    if child.relation in("nsubj", "nsubj:pass") then
        for desc in child.descendants do
            if desc.relation="det" and desc.parent=child.parent then
                tags.append(ATTR(desc))
            else
                tags.append(ATTR_PROP(desc))
    else if child.relation="conj" then
        annotate_all_descendants(desc, ATTR_PROP)
```

Algorithm 4: OBJECT AND CONTEXT tagger.

```
Data: tree: LexicalTreeNode, root: tree.root, tags: list of tags
for child in tree.children do
    if child.relation == "obj" then
        tags.append(OBJECT(child))
        for desc in child.children do
            if desc.relation="advcl" then
                annotate_all_descendants(desc, CONTEXT)
            else if desc.relation in ("det", "amod", "case", "compound")
             then
                tags.append(OBJECT(desc))
                annotate_all_descendants(desc, OBJECT_PROP)
            else if desc.relation in (nmod", "nmod:poss") then
                tags.append(OBJECT(desc))
                for child_desc in desc.descendants do
                    if child_desc.relation in ("case", "amod") then
                        tags.append(OBJECT(child_desc))
                    else
                        tags.append(OBJECT_PROP(child_desc))
            else
                annotate_all_descendants(desc, OBJECT_PROP)
    else if child.relation in ("advcl", "obl") or (child.relation ==
     "advmod" and child.lemm != "not") then
        annotate_all_descendants(desc, CONTEXT)
    else if child.relation="ccomp" then
        if any(child.children = "that" and
         child.children.relation="mark") then
            annotate_all_descendants(child, OBJECT)
        else
            annotate_all_descendants(child_desc, CONTEXT)
    else if child.relation="xcomp" then
        for desc in child.children do
            if desc.relation="obj" then
                tags.append(OBJECT(desc))
                annotate_all_descendants(child_desc, OBJECT_PROP)
            else
                annotate_all_descendants(child_desc, CONTEXT)
```

1. AIM
   (a) If a sentence contains one word with *root* tag, then annotate this word as AIM.
   Otherwise, stop the annotation and return empty results. (AIM is required in the regulative institutional statement and no AIM was found.)

(b) If the word annotated in 1a has a child with *passive auxiliary* or *copula* relation, then annotate this child as AIM.

(c) If the word annotated in 1a has a child with *auxiliary* relation and that child's lemma is one of "be", "have", "do", then annotate this child as AIM.

(d) If the word annotated in 1a has a child with *adverb modifier* relation and that child's lemma is "not", then annotate this child as AIM.

2. DEONTIC

If the word annotated in 1a has a child with *auxiliary* relation and that child's lemma is one of "must", "should", "may", "might", "can", "could", "need", "ought", "shall", then annotate this word as DEONTIC.

3. ATTRIBUTE

(a) If the word annotated in 1a has a child with *nominal subject* or *passive nominal subject* relation, then annotate this child as ATTRIBUTE. Otherwise, stop the annotation and return empty results. (ATTRIBUTE is required in the regulative institutional statement and no ATTRIBUTE was found.)

(b) If the word annotated in 3a has a child with *determiner* or *adjectival modifier* relation, then annotate this word as ATTRIBUTE.

(c) Annotate remaining children of the word annotated in 3a as ATTRIBUTE PROPERTY

(d) If the word annotated in 1a has a child with *conjunct* relation, then annotate this child and all descendants as ATTRIBUTE PROPERTY.

If no attribute has been found after these steps, then stop the annotation and return empty results.

4. OBJECT and CONTEXT

(a) If the word annotated in 1a has a child with *obj* relation, then annotate this word as OBJECT.

(b) If the word annotated in 4a has a child with one of *det*, *amod*, *case*, *compound* relation, then annotate this child as OBJECT.

(c) If the word annotated in 4a has a child with *adverbial clause modifier*, then annotate this child and all descendants as CONTEXT.

(d) Annotate remaining children of the word annotated in 4a as OBJECT PROPERTY.

(e) If the word annotated in 1a has a child with *open clausal complement* relation:

   – If this word has a child with *object* relation, then annotate this child as OBJECT and all child's descendants as OBJECT PROPERTY.
   – Annotate remaining children of the word found in 4e and all descendants as CONTEXT.

(f) If the word annotated in 1a has a child with one of *adverbial clause modifier*, *obl*, *adverb modifier* or *clausal complement* relation, then annotate this word and all descendants as CONTEXT.

4.2 Rules for Constitutive Statements

MODAL tag is determined analogously to tag DEONTIC in regulative layer. The way the tagger works in constitutive statements is shown in Algorithm 5 and points listed below.

Algorithm 5: CONSTITUTIVE tagger.

```
Data: tree: LexicalTreeNode, root: tree.root, tags: list of tags,
      properties: root's children with relation "obl", "obj", or "advcl",
      entities: root's children with relation "nsubj", "nsubj:pass", or
      "expl", context: root's children with relation "advmod" or
      "xcomp"; csubj: root's children with relation "csubj", cop: root's
      children with relation "cop"; entities_noun: root's children with
      relation "det", "acl", or "nmod:npmod"
if root.tag="ADJ" or root.tag="VERB" then
    if root.tag="ADJ" then
        tags.append(CONST_PROP(root))
    else
        tags.append(CONST_FUNCTION(root))
    for entity in entities do
        tags.append(CONST_ENTITY(entity))
        for child in entity.children do
            if child.relation in ("det", "compound", "mark") then
                annotate_all_descendants(child, CONST_ENTITY)
            else
                annotate_all_descendants(child,
                  CONST_ENTITY_PROP)
    for property in properties do
        annotate_all_descendants(property, CONST_FUNCTION)
    for c in context do
        annotate_all_descendants(c, CONTEXT)
else if root.tag="NOUN" then
    if len(csubj)=1 then
        tags.append(CONST_FUNCTION(csubj))
        for child in csubj.children do
            if child.relation="obl then
                annotate_all_descendants(child, CONTEXT)
        for child in cop do
            annotate_all_descendants(child, CONST_ENTITY)
    else
        if len(cop)=1 then
            tags.append(CONST_FUNCTION(cop))
        tags.append(CONST_ENTITY(root))
        for entity in entities_root do
            for desc in entity.descendants do
                if desc.relation in ("det", "mark") then
                    annotate_all_descendants(desc, CONST_ENTITY)
                else
                    annotate_all_descendants(desc,
                      CONST_ENTITY_PROP)
        for child in properties do
            annotate_all_descendants(child, CONST_PROP)
```

1. If a sentence contains one word with *root* tag and this word is a verb or an adjective:
   (a) If the word founded in 1 is a noun, then annotate it as CONSTITUTIVE FUNCTION, otherwise as CONSTITUTING PROPERTIES.
   (b) If the word annotated in 1a has a child with *aux:pass* or *cop* relation, then annotate this child as CONSTITUTIVE FUNCTION.
   (c) If the word annotated in 1a has a child with *aux* relation and that child's lemma is one of "be", "have", "do", then annotate this child as CONSTITUTIVE FUNCTION.
   (d) If the word annotated in 1a has a child with one of *nsubj*, *nsubj:pass* or *expl* relation, then annotate this child as CONSTITUTED ENTITY.
   Otherwise, stop the annotation and return empty results. (CONSTITUTED ENTITY is required in the constitutive institutional statement, and no CONSTITUTED ENTITY was found.)
   (e) If the word annotated in **??** has a child with one of *det*, *compound*, *mark*, then annotate this child and all child's descendant as CONSTITUTED ENTITY. Annotate remaining children of the word annotated in **??** as CONSTITUTED ENTITY PROPERTY.
   (f) If the word annotated in 1a has a child with *obj* or *advcl* relation, then annotate this child and all child's descendants as CONSTITUTED PROPERTIES.
   (g) If the word annotated in 1a has a child with one of *obl*, *advmod*, *xcomp* relation, then annotate this child and all child's descendants as CONie wiem co wiecej o nim pisac, bo to model, ktory zbudowalam NTEXT.
2. If a sentence contains one word with *root* tag and this word is a noun, then:
   (a) Annotate the word founded in 2 as CONSTITUTIVE ENTITY.
   (b) If the word annotated in 2a has a child with one of *det*, *acl* or *nmod:npmod*, then annotate this child and all child's descendants as CONSTITUTIVE ENTITY.
   (c) If the word annotated in 2a has a child with *csubj* relation, then annotate this child as CONSTITUTIVE FUNCTION.
   (d) If no word was annotated in 2c and the word annotated in 2a has a child with *cop* relation, then annotate this child as CONSTITUTIVE FUNCTION.
   Otherwise, stop the annotation and return empty results. (FUNCTION is required in the constitutive institutional statement, and no FUNCTION was found.)
   (e) If any word was annotated in 2c and this word has a child with *obl* relation, then annotate this child and all child's descendants as CONTEXT.
   (f) If the word annotated in 2a has a child with *nsubj* or *nsubj:pass* relation, then annotate this child and all child's descendants as CONSTITUTING PROPERTIES.

References

1. Frantz, C.K., Siddiki, S.: Institutional grammar 2.0: A specification for encoding and analyzing institutional design. Public Administration **n/a**(n/a) (2020). https://doi.org/10.1111/padm.12719, `http://dx.doi.org/10.1111/padm.12719`
2. Qi, P., Dozat, T., Zhang, Y., Manning, C.D.: Universal Dependency parsing from scratch. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 160–170. Association for Computational Linguistics, Brussels, Belgium (Oct 2018). https://doi.org/10.18653/v1/K18-2016, `https://www.aclweb.org/anthology/K18-2016`

5 Raw Ontology Builder

5.1 Defining Classes and Their Hierarchy

Class and subclass names are defined by unique combinations of IG tags values from statements. Table 2 describes which specific IG tags are used in which statement type to define classes and subclasses. Example:

```
[Constituted Entities] = employee,

[Function] = is,
[Constituted Properties] = unable,
[Constituted Properties Property] = to work
Class: employee,
Subclass: employeeThatIsUnableToWork
```

Table 2. IG tags used to define class and subclass by each statement type. Note that '+' denotes concatenation of IG tag value.

statement type	class	subclasses
observation constitutive	*Constituted Entities*	*Constituted Entities* + 'that' + *Function* + *Constituted Properties*
proper constitutive	*Constituted Entity*	*Constituted Entity* + *Constituted Entity Property* + *Constituted Properties Property*
proper constitutive	*Constituted Properties*	*Constituted Properties* + *Constituted Properties Property*
proper/observation regulative	*Attributes*	*Attributes* + *Attributes property*
proper/observation regulative	*Direct Object*	*Direct Object* + *Direct Object Property*
proper/observation regulative	*Indirect Object*	*Indirect Object* + *Indirect Object Property*

5.2 Defining Relations

Relations are defined based on *observations regulative*, *proper regulative* and *proper constitutive*. As raw ontology is supposed to be used for the analysis of possible relationships that emerge from regulation, we treat differently relations that are observational and relations that are *regulative* or *constitutive*. From *regulative observations*, we define *possible observed relations*, which are later modelled as antecedents in SWRL rules. Relation is named by *Aim* tag. If there is an non empty *Indirect Object* tag, we define the second relation, which name is created by transforming *Aim* to passive form. Then we define *regulative relations*. Relation name is defined by [*Deontic* + *Aim*]. *Deontic* is also stored as a relation property, which is later used for analysis. Simmilary, if there is *Indirect Object*, we define the second relation, named after [*Deontic* + passive(*Aim*)]. Finally, we define *constitutive modal relations* from proper constitutive statements using [*Modal* + *Function*] tags as the relation name. Details of domains and ranges of each relation are provided in Table 3.

Both in *constitutive modal relations* and *regulative relations*, we treat each relation independently – that means that for every class domain we create different relations – reason for this, is again, that we want use this ontology primarily for quantitative analysis of such relations.

Table 3. IG tags used to define relations by each statement type.

statement type	domain	relation name	range
observation regulative	*Atttribute* + *Attribute Property*	Aim	*Direct Object* + *Direct Object Property*
observation regulative	*Direct Object* + *Direct Object Property*	passive(Aim)	*Indirect Object* + *Indirect Object Property*
proper regulative	*Atttribute* + *Attribute Property*	*Deontic* + *Aim*	*Direct Object* + *Direct Object Property*
proper regulative	*Direct Object* + *Direct Object Property*	*Deontic* + passive(*Aim*)	*Indirect Object* + *Indirect Object Property*
proper constitutive	*Entity* + *Entity Properties*	*Modal* + *Function*	*Constituted property* + *Constituted Property Properties*

5.3 Defining Axioms and SWRL Rules

Axioms and SWRL rules are build only on basis of proper regulative and proper constitutive statements – those statements which really constitute or regulate some part of reality. In those statements, if *Activation Condition* is empty, relation defined by such statement must always hold, then we add axiom that every subject of the class defined by this row must be in such relation with object (same for the relation between object and indirect object).

If there exists *Activation Condition* referring to some observational statements, we define SWRL rules where the antecedent is defined by referred statement and the consequent is statement that is referring. We iterate over all statements referred in activation condition tag. For every statement, we check whether it is *regulative observation* or *constitutive observation.* For *regulative observation* we add rule that has all relations defined in that statement in the antecedent. For *constitutive observation* antecedent, consist of the constraint on being a specific subclass defined by such statement. In consequent, there are all relations defined this proper regulative/constitutive statement that we are considering.

Example rule created automatically is as follows:

```
EmployeeThatIsUnableToWork(?x),
    EmployeeEmployedByEmployer(?x), Employer(?y),
    PaidSickTime(?q) -> shall_provide(?y,?q),
    shall_be_provided_to(?q,?x)
```