Web Semantics: Science, Services and Agents on the World Wide Web 27–28 (2014) 10–18

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents
on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Constitute: The world's constitutions to read, search, and compare

Zachary Elkins[a], Tom Ginsburg[b], James Melton[c], Robert Shaffer[a], Juan F. Sequeda[d,*], Daniel P. Miranker[d]

[a] *Department of Government, The University of Texas at Austin, United States*
[b] *Law School, University of Chicago, United States*
[c] *Department of Political Science, University College London, United Kingdom*
[d] *Department of Computer Science, The University of Texas at Austin, United States*

ARTICLE INFO

Article history:
Received 18 March 2014
Received in revised form
25 June 2014
Accepted 7 July 2014
Available online 8 September 2014

Keywords:
Constitution
Search
RDB2RDF
Direct mapping
R2RML
Ontology

ABSTRACT

Constitutional design and redesign is constant. Over the last 200 years, countries have replaced their constitutions an average of every 19 years and some have amended them almost yearly. A basic problem in the drafting of these documents is the search and analysis of model text deployed in other jurisdictions. Traditionally, this process has been ad hoc and the results suboptimal. As a result, drafters generally lack systematic information about the institutional options and choices available to them. In order to address this informational need, the investigators developed a web application, Constitute [online at http://www.constituteproject.org], with the use of semantic technologies. *Constitute* provides searchable access to the world's constitutions using the conceptualization, texts, and data developed by the *Comparative Constitutions Project*. An OWL ontology represents 330 "topics" – e.g. right to health – with which the investigators have tagged relevant provisions of nearly all constitutions in force as of September of 2013. The tagged texts were then converted to an RDF representation using R2RML mappings and Capsenta's Ultrawrap. The portal implements semantic search features to allow constitutional drafters to read, search, and compare the world's constitutions. The goal of the project is to improve the efficiency and systemization of constitutional design and, thus, to support the independence and self-reliance of constitutional drafters.

1. The problem: Drafting new constitutions

Like any other legal document, Constitutions build on prior models and templates. After all, drafters face many of the same challenges and goals as their predecessors, so it is unsurprising that constitutional phrasing and ideas are repeated across jurisdictions. Think of this as something akin to Constitutional plagiarism. Model text from other jurisdictions can help drafters identify multiple dimensions to a given constitutional problem and provide effective language to address the problem. Thus, one of the early tasks in the constitutional design process is the search for, and analysis of, alternative models. Often, early drafts built on these models become the basis of discussion during the deliberative stages of the design process. Also, because of strong inertial factors, these early drafts can be particularly consequential [1]. In short, a highly consequential step in constitutional design is the search for models and templates on which to build and adapt.

Such search is understandably challenging. Part of the problem is unavoidable. Drafters are under all sorts of constraints. They often work in periods of crisis, under significant time pressure, with a limited research staff, and with very little experience in drafting higher law. Another part of the problem – perhaps the most significant part – is easily remedied. Drafters commonly do not have unfettered access to an adequate sample of constitutional texts, indexed by topic. As a result, the search process can be highly inefficient, unsystematic, and its results unrepresentative of modern models of constitutional design.

The scale of the problem is hard to overestimate. Most (but certainly not all) constitutions are vested with enormous amounts of symbolic and legal power. Founders use these documents to establish national principles and aspirations, to define and unify the state, and to restrict future governments from crossing certain limits. When they work, constitutions form the backbone for

[*] Corresponding author. Tel.: +1 5124844977.
E-mail addresses: zelkins@austin.utexas.edu (Z. Elkins), tginsburg@uchicago.edu (T. Ginsburg), j.melton@ucl.ac.uk (J. Melton), rbshaffer@utexas.edu (R. Shaffer), jsequeda@cs.utexas.edu, juanfederico@gmail.com (J.F. Sequeda), miranker@cs.utexas.edu (D.P. Miranker).

http://dx.doi.org/10.1016/j.websem.2014.07.006
1570-8268/© 2014 Elsevier B.V. All rights reserved.

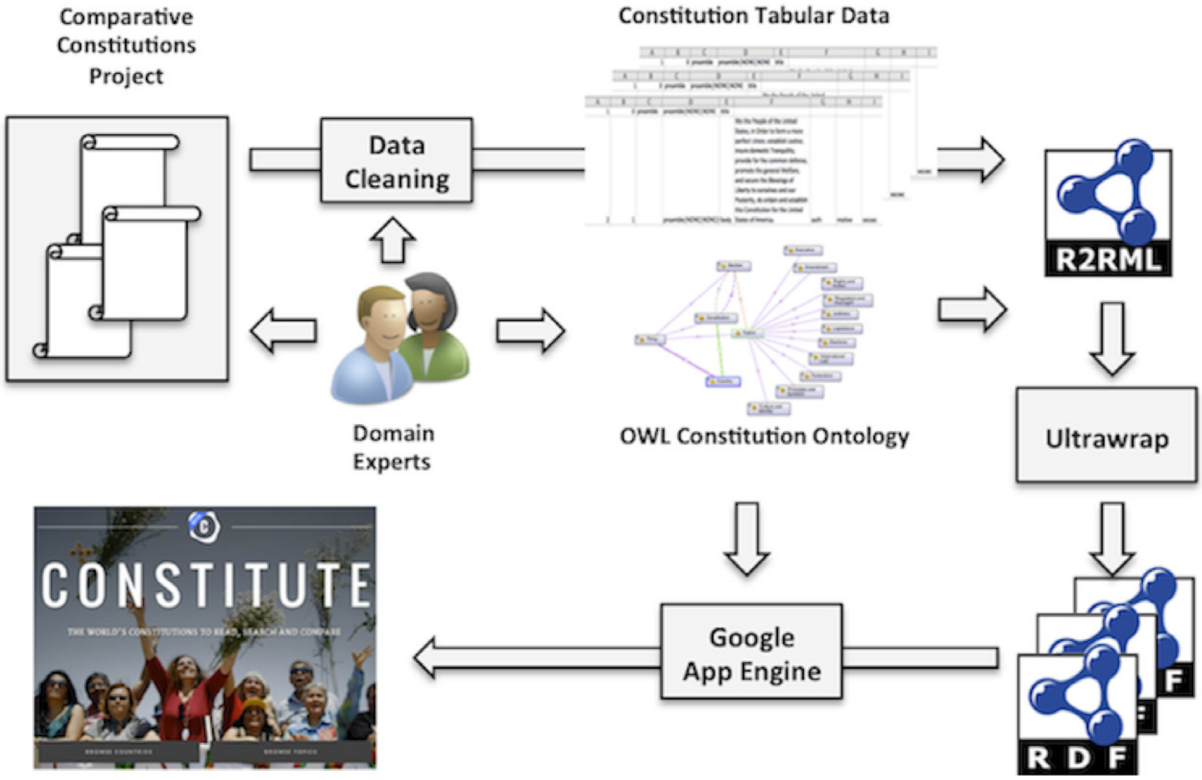

Comparative Constitutions Project
Data Cleaning
Constitution Tabular Data
R2RML
Domain Experts
OWL Constitution Ontology
Ultrawrap
CONSTITUTE
THE WORLD'S CONSTITUTIONS TO READ, SEARCH AND COMPARE
Google App Engine
RDF

Fig. 1. Architecture of constitute.

rule of law and democracy. Perhaps disturbingly, leaders seek to remake constitutions more often than one might think. On average, constitutions are replaced every 19 years and amended in smaller ways nearly every year [2]. Sometimes, this re-design corrects prior and unanticipated problems or allows laws to conform to evolving understandings and conditions. At other times, this process is self-serving, meant to lengthen the rule or enhance the power of the ruling elite. Regardless, at any given moment, constitutional redesign is going on in some part of the world.

In short, the redesign of national constitutions is one of the most consequential acts of public life and it recurs with relative frequency. It is also one whose process is amenable to real improvement.

2. The solution: Constitute

Constitute reproduces the search and analysis process of constitutional design, or at least the idealized version of that process. The application contains the constitutional text in force in nearly every independent state as of September 2013.[1] Each constitution is indexed with roughly 330 topics drawn from the conceptual inventory (described below) of the *Comparative Constitutions Project* (CCP). Users can extract the text for any one of these topics across the full sample of constitutions, or across a set of constitutions filtered by region or by the date the constitution was enacted. Users can pin individual excerpts to a sub-panel of the interface, and then export the compiled excerpts as a.pdf document or directly into the Google Drive environment as a "Doc". Excerpts can also be exported as.csv files for more systematic analysis. A standard collaborative use of Constitute is to extract a representative set of excerpts (perhaps 15 or so) on any given topic – e.g., the right to privacy – for further analysis or to share with fellow drafters or drafting committee members. Users can also view the full text of constitutions on the site.

Constitute is a semantically enabled search portal, built using Semantic Web technologies. First, the relationships among the constitutional topics and geographic regions were conceptualized in an OWL ontology, called the Constitution Ontology. Second, each tabular representation of the 191 constitutions is integrated and mapped to the OWL ontology by converting them to RDF using a combination of Direct Mapping, R2RML, and Capsenta's Ultrawrap. Finally, the OWL ontology enables the navigation and search of the constitutional text through the topic hierarchy and subclass reasoning. We anticipate that Constitute is the first of multiple applications that exploit the underlying linked data.

3. Constitute architecture

Fig. 1 represents the architecture of Constitute. The authors created the Constitution Ontology in OWL, which represents the taxonomical relationship between constitutional topics, subtopics, and geographic regions. Subsequently, data from the CCP is cleaned and converted to a plain-text tabular format to which topic tags for the relevant provisions are attached. Mappings between the tabular data and the Constitution Ontology are represented in R2RML, which is then used to generate the RDF using Ultrawrap. Finally, the RDF data and Constitution Ontology are used to create a search portal built on top of Google App Engine.

3.1. The comparative constitutions project

Constitute leverages data and conceptual resources developed by the authors of the CCP. The central component of the CCP is a set of quantitative data that codifies the content (and meaning)

[1] Certain countries whose constitutional order is composed of multiple documents (e.g. the United Kingdom) are presently excluded.

of national constitutions since 1789, with continuing updates that reflect ongoing constitutional events.[2]

Much of the CCP data, which includes textual interpretation, is not revealed on *Constitute*, which for the most part leaves textual interpretation to the user. However, three elements of the CCP infrastructure are critical. The first of these is a complete census of each country's constitutional history. This accounting is, of course, an enormous historical inquiry in its own right. The CCP's authors have identified 742 new constitutions and 2544 formal amendments written in independent countries between 1789 and 2013. A second resource is the texts themselves. The CCP's authors have located, digitized, and (sometimes) translated to English the text associated with each of the changes identified in the chronology. At present, the CCP repository includes 707 of the 742 texts and 2183 of the 2544 amendments to these systems. Not all of this historical work is built into Constitute, which currently includes texts of only those constitutions that were in force as of September 2013. Future updates to Constitute will add the historical constitutions, which are particularly important to scholars, if not drafters. An updated version of the site allows side-by-side comparisons between any two constitutions, including some prior constitutions in a country's series of texts.

A third critical element is the CCP conceptual framework. In order to interpret the constitutional text, CCP investigators developed and refined a conceptual frame that includes over 600 attributes of constitutions. That frame was then converted to a highly "guided" survey instrument, which the project's coders use to read and interpret the texts. Constitutions are coded by two independent coders— typically, Political Science graduate students and Law students. The two codings are reconciled by a third coder in the event of any disagreement. A large number of interpretive questions are adjudicated on a message board, in which the project's principal investigators render judgments that serve as precedent for future cases. Importantly, for Constitute at least, the coders record not only the meaning of a given provision, but also *where* – e.g., which Article or Section – that meaning is found. So, for example, CCP data will specify not only that the Albanian constitution identifies an official religion, but also that it does so in Article 10, Sub-section 1. These variables form the base CCP data, from which the investigators extract the *topic data* for use in *Constitute*. The *Topic data* for *Constitute* includes 330 high level topics – e.g., official religion – that are drawn from the over 600 attributes included in the CCP *base data*. This conceptual work informs the OWL ontology, which we describe in more detail below.

The topic data for each constitution are represented as a set of tuples, which includes the constitution name, a short label for the topic, a topic code, and a numerical reference to an organization header. A typical entry might read "['Albania_2008', 'Official religion', 'offrel', '10.1']". This means that Section 10.1 of the Albanian constitution of 2008 has been tagged with the topic *official religion*. The topic data is then used to attach topic tags to excerpts from each constitutional text.

3.2. Text preparation

Text preparation begins with a domain expert, typically a Political Science graduate student, selecting a Constitution and downloading an uncorrected OCR scan of its text from the CCP repository. Subsequently, the domain expert cleans the document by fixing typos, errant line breaks, and bad characters. He or she then identifies all organizational headers – e.g. "Chapter" or "Article" –, any titles attached to those headers, and itemized lists in the text. Next, a Python script – customized with regular expressions to reflect each constitution's format – identifies parent–child relationships between the Constitution's organizational headers, and segments each document according to its organizational structure. The script then uses preexisting CCP data to label paragraphs of text associated with each header tag, and generates a tabular representation of the document. Each line in this tabular file consists of a header name, that header's position within the document's organizational hierarchy, and the title, text, and tags (if any) associated with that header. The domain expert then updates and corrects these tabular files as necessary. The tabular data is represented in Microsoft Excel so the domain experts can make use of the track-change features across versions. It took, roughly, 10 domain experts 9 months to clean and generate 180 constitution .xls files.

3.3. Constitution ontology

The search function of Constitute makes use of the Constitution Ontology, which is in the form of an OWL file and currently consists of two main parts: Topics and Geography. The topics component describes the taxonomical relationships of the constitutional topics (synonyms and classes) from the CCP conceptual frame. For example, the topic "Freedom of Religion" is a subclass of both "Religion" and "Civil and Political Rights". Additionally, "Religion" is a subclass of "Culture and Identity" while "Civil and Political Rights" is a subclass of "Rights and Duties". In addition, the ontology describes the origins of the topics used on Constitute, linking each topic to the question, or questions, from the CCP's survey instrument that it is based upon.

The geography component is an import and extension of the FAO Geopolitical Ontology.[3] Extensions here include the elaboration of geographic sub-regions (e.g. the Balkans and Middle East), the addition of newly independent countries (e.g., South Sudan and Kosovo) and, finally, the addition of synonyms for country names—e.g. Holland as a synonym for the Netherlands.

Fig. 2 depicts a portion of the Constitution Ontology. The authors are extending and refining the ontology on a Wiki platform (see below), which ultimately includes edits from users. Ongoing revisions include the addition of attributes (of both constitutions and countries) and the further specification of relationships among topics (see below).

3.4. RDF generation

The XLS representation of each constitutional text was loaded into a Microsoft SQL Server database with one table per constitution. By loading the spreadsheets into a relational database, the mappings between the constitution data and the Constitution Ontology could be represented in R2RML. R2RML, the Relational Database to RDF mapping language [3], in conjunction with the Direct Mapping [4], are two recently ratified standards by the W3C to expose relational databases to the Semantic Web. Capsenta's Ultrawrap,[4] a productized version of a research prototype [5], was used to convert the constitutional data into RDF. Ultrawrap supports both W3C mapping standards, and both extract, transform, and load (ETL) of triples into a triplestore and SPARQL execution on relational data. First, the Direct Mapping created an initial default mapping represented in R2RML. Subsequently, the R2RML was edited to use terms from the Constitution Ontology. In this current phase of Constitute, Ultrawrap generates periodic ETL dumps of the constitution data as RDF. Fig. 3 shows an example RDF graph from Constitute and how it is linked to DBpedia.

[2] http://comparativeconstitutionsproject.org/.

[3] http://www.fao.org/countryprofiles/geoinfo/geopolitical/resource/.

[4] http://www.capsenta.com.

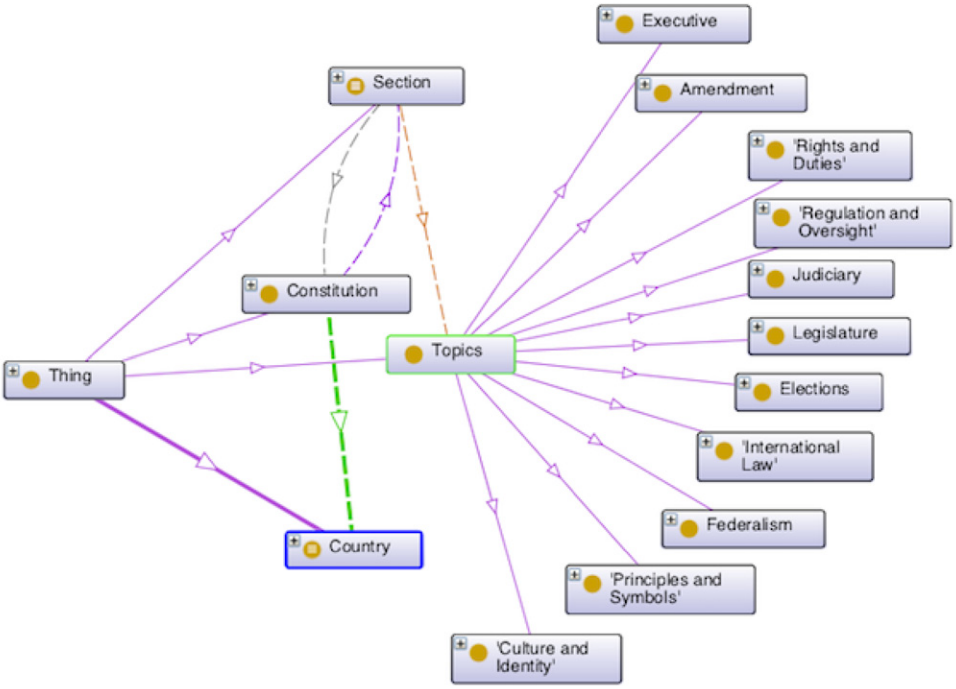


Fig. 2. The constitution ontology.

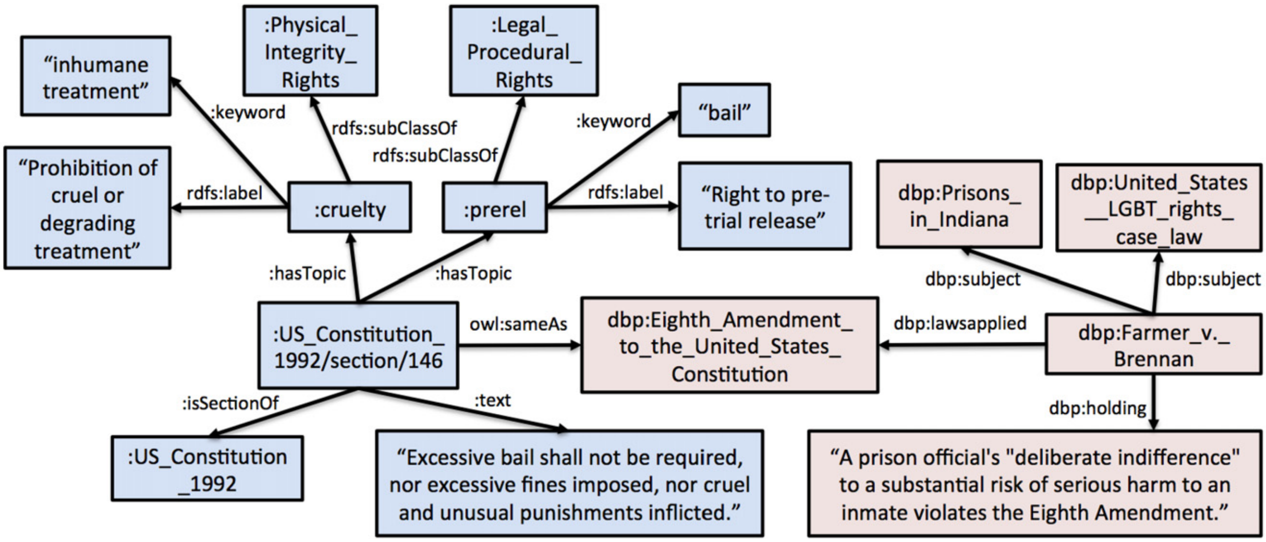

"inhumane treatment"
:Physical_ Integrity_ Rights
:Legal_ Procedural_ Rights
"bail"
:keyword
rdfs:subClassOf
rdfs:subClassOf
:keyword
"Prohibition of cruel or degrading treatment"
rdfs:label
:cruelty
:prerel
rdfs:label
"Right to pre-trial release"
dbp:Prisons_ in_Indiana
dbp:United_States __LGBT_rights_ case_law
:hasTopic
:hasTopic
dbp:subject
dbp:subject
:US_Constitution_ 1992/section/146
owl:sameAs
dbp:Eighth_Amendment_ to_the_United_States_ Constitution
dbp:lawsapplied
dbp:Farmer_v._ Brennan
:isSectionOf
:text
dbp:holding
:US_Constitution _1992
"Excessive bail shall not be required, nor excessive fines imposed, nor cruel and unusual punishments inflicted."
"A prison official's "deliberate indifference" to a substantial risk of serious harm to an inmate violates the Eighth Amendment."

Fig. 3. Constitute RDF graph (in blue) linked to DBpedia (in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.5. *Search portal*

The Constitute search application was built using Google App Engine,[5] Python, and the RDFlib library.[6] Free-text search is powered by indexing raw constitutional text using the Google App Engine search API. Semantic search is powered by the RDF triples and OWL ontology, which is stored in Google's DataStore.[7]

Google's DataStore supports a entity–property-value data model which enables the storage of RDF. Data objects are considered entities, which can have one or more properties, which in turn can have one or more values.[8] Entities stored in the DataStore include constitutional topics, geographic regions, constitutional sections, and the text of each section. An index of the topics enables retrieval of the sections of a constitution from a topic search.

Constitute implements subclass inference in the search box. The assertions implied by the subclass entailments in the ontology are materialized. For example, in Fig. 4 a user types "Religion". The autocomplete field then suggests topics that are semantically related to the Religion, such as "Separation of church and state", a subclass of Religion.

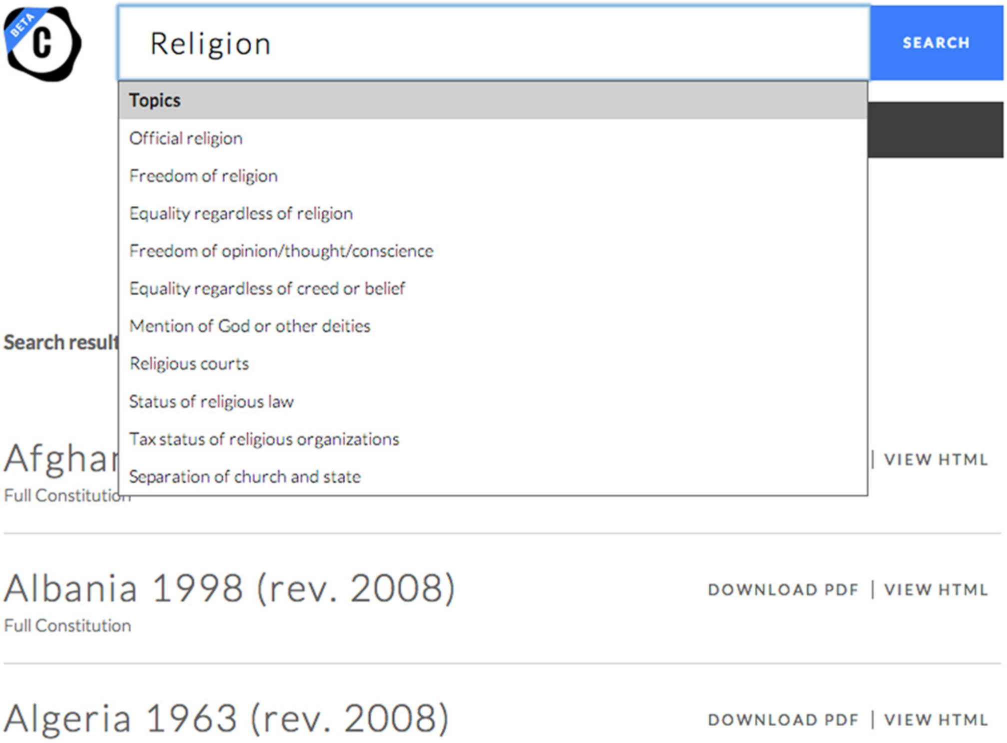
Users can also browse topics from a fixed menu, which displays subclass relationships from the Constitution Ontology. See Fig. 5.

[5] https://developers.google.com/appengine/.

[6] https://github.com/RDFLib.

[7] https://developers.google.com/datastore/.

[8] https://developers.google.com/datastore/docs/concepts/entities.



Religion

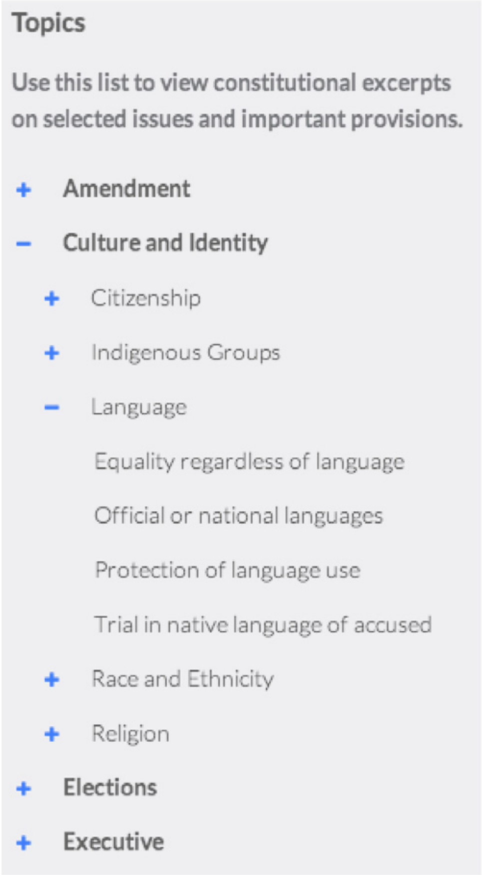
Fig. 4. Subclass reasoning in topic search: user searches for "religion" and retrieves related topics that are specified as subclasses of religion.

Each topic is an ontological class. Clicking on a topic reveals a set of sub-topics. Constitutional excerpts are tagged with the leaf classes of the ontology. For example, clicking on "Official or national languages", will display all the excerpts of constitutions corresponding to that topic. Results can be further filtered by geographic regions and/or by the year in which the constitution was enacted.

Search results are delivered to the front-end as JSON and the client implemented using AngularJS.[9] Where possible, HTML5 browser-based caching is used. CSS3 media queries are used to provide responsive design across desktop and mobile browsers.

4. Constitute as an enabler

Constitute is designed for a very particular audience: constitutional drafters. However, one can imagine a broad set of educational, scholarly, and policy-making applications for the corpus of texts. In modern times, constitutional design is hardly limited to elites. Constitutions are often designed to be read and understood by both experts and citizens alike. Also, as the process of constitution making becomes more participatory (there are even efforts to crowdsource constitution-making, e.g. the Constitution UK project at the London School of Economics),[10] one can imagine some version of the search-and-analysis process at the citizen level. Most constitutional drafting processes include a consultative stage in the beginning and a ratification stage towards the end, in which ordinary citizens propose and approve, respectively, constitutional ideas. Obviously these proposal and evaluative functions require a clear set of textual reference points. Sometimes the reference point may only be the outgoing constitutional text. Prior to the Egyptian



Use this list to view constitutional excerpts on selected issues and important provisions.

Fig. 5. The constitution ontology in constitute.

[9] https://angularjs.org/.

[10] See http://blogs.lse.ac.uk/constitutionuk/ for more information.

constitutional referendum in January 2014, the CCP's authors exploited the ontology underneath constitute to build a visualization tool that facilitates comparisons of the 1971, 2012 and 2014 Egyptian constitutions.[11] Future version of *Constitute* will include such A-to-B comparisons.

Users of *Constitute* in the constitution-making sphere may very well represent a minority of users. The underlying data was collected by scholars in order to answer research questions about the origins and consequences of constitutional provisions. This sort of analysis and description of political institutions is very likely a staple of a basic civics education in most parts of the world. Of course, more sophisticated analysts of the corpus of texts may develop content-analytic techniques to extract insights and meanings from the texts and even build additional web applications that utilize the data. Constitute will facilitate such endeavors in three respects.

4.1. Exposure and analysis of the data

The data underlying *Constitute* can be directly queried using the site's SPARQL endpoint.[12] This allows access to both the metadata associated with each text and the tagged textual data. These data can be used by anyone to perform in-depth analyses of in force constitutions.[13]

For example, suppose one is interested in understanding the age of all constitutions currently in force. This piece of metadata (enactment date) is returned on Constitute search results (e.g., in Fig. 4, Albania's constitution is listed with an enactment date of 1998). Alternatively, one could retrieve this information – and other pieces of metadata – by submitting the following query to the SPARQL endpoint:

```
SELECT ?const ?country ?year
WHERE {
 ?const :isConstitutionOf ?country .
 ?const :yearEnacted ?year .
}
```

The query returns data containing the name of each constitution on the site, the country where each constitution is in force, and the year each constitution was enacted. One might then produce a simple graphic such as the histogram in Fig. 6, which depicts the distribution of enactment dates for constitutions in force as of September 2013.

For most users, metadata on constitutions is probably less interesting than the actual texts of those constitutions. Fortunately, the texts are also searchable via the SPARQL endpoint. For example, suppose one is interested in comparing the environmental provisions in North and South America to those in Africa. One could search for "protection of environment" on Constitute, filter the results by the relevant regions, and carefully read all of the relevant provisions. Alternatively, one could perform the following SPARQL query:

```
SELECT ?const ?country ?region ?sectionType
?sectionText ?childType ?childText
WHERE {
 ?const :isConstitutionOf ?country .
 ?country :isInGroup ?region .
 ?section :isSectionOf ?const .
 ?section :hasTopic ontology:env .
 ?section :rowType ?sectionType .
 OPTIONAL {?section :text ?sectionText}
 OPTIONAL {
  ?childSection :parent ?section .
  ?childSection :rowType ?childType
 }
 OPTIONAL {
  ?childSection :parent ?section .
  ?childSection :text ?childText
 }
}
```

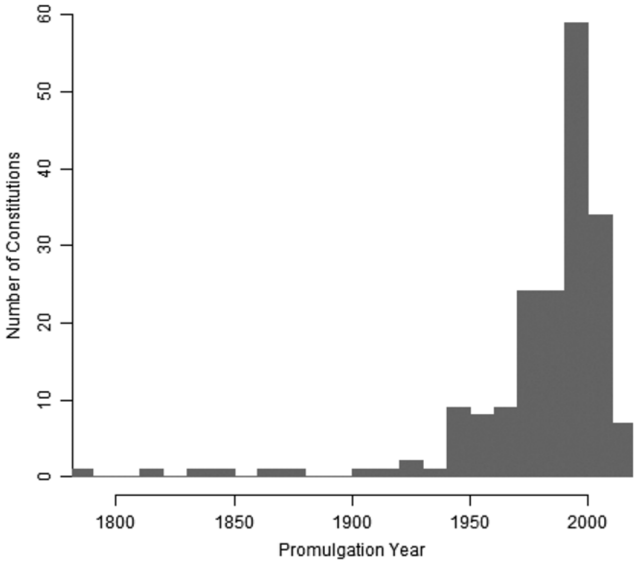


Fig. 6. Promulgation dates of constitutions in force on September 1, 2013.

This query will return each excerpt from the full set of in-force constitutions that is tagged with "protection of environment". Then, one simply has to subset the data by region, and analyze it using some appropriate text analytic software. For instance, one could create a word cloud, such as that in Fig. 7. The word cloud reveals that many more words are devoted to the environment in the Americas than in Africa, almost twice as many. One can also detect differences in the way the environment is discussed in constitutions between the two regions. Although constitutions from both regions use words like "protect" and "right", words like "conservation" and "preservation" are much more likely in the Americas, as are mentions of specific natural resources like "water" and "land".

Not only can one use Constitute to detect differences in the way constitutional topics are addressed across regions, but one could also look for similar differences over time. For instance, one might suspect that socioeconomic rights are addressed differently in constitutions written before and after the Cold War. This is a hypothesis that can be tested using the corpus of textual data underlying the Constitute site. Moreover, one could apply much more sophisticated text mining methods to the Constitute corpus, e.g. topic models or even network analysis. *Constitute*, in this sense, represents the textual version of the coded CCP data. The CCP data allow scholars to analyze spatiotemporal trends in the prevalence of very specific constitutional provisions. Constitute allows scholars to not just analyze the likelihood that some constitutional provision will be present across contexts but also *how* that provision is addressed in different contexts.

[11] http://comparativeconstitutionsproject.org/comparing-the-egyptian-constitution/.

[12] http://data.constituteproject.org.

[13] A tutorial for creating Figs. 6 and 7 in R is available at http://constitution-unit.com/2013/12/11/directly-querying-the-constitute-data/.

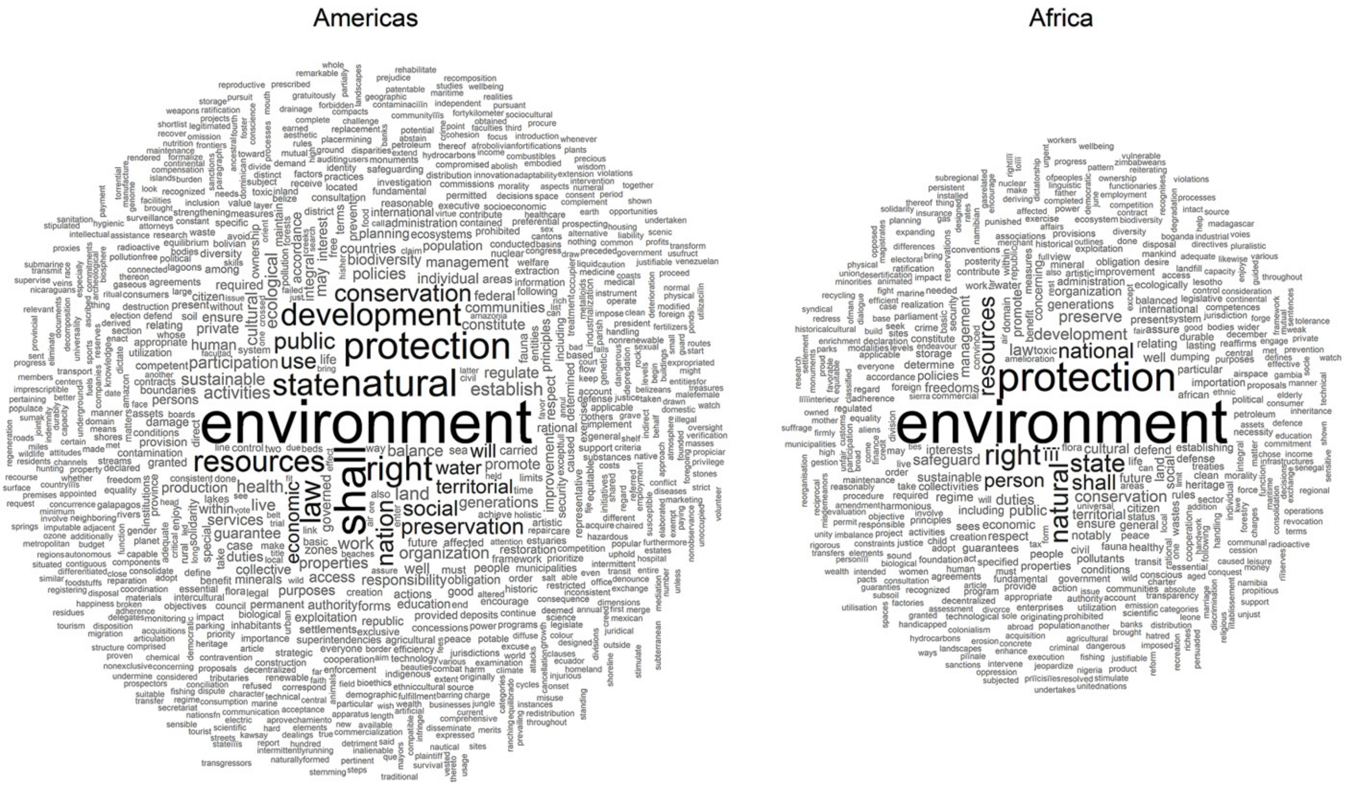


Fig. 7. Constitutional provisions on the environment.

4.2. Linked data

Of course, one of the virtues of exposing the data as RDF is compatibility with an increasing rich set of data in similar form. For example, Wikipedia, and hence DBpedia, contains information on the constitution of almost every country, including information on the motivation for specific articles and amendments as well as their interpretations by the courts. For instance, there is an article in Wikipedia devoted to each amendment to the US constitution, and these pages often include references to important Supreme Court cases, which represent authoritative interpretations of those amendments. We have started to manually link US constitutional provisions from Constitute to the relevant data in DBpedia (see Fig. 3).

After linking Constitute with DBpedia, the following SPARQL query can find all the US Supreme Court cases on the topic of cruelty:

```
SELECT ?lawLabel ?abstract ?subjectLabel ?holding
WHERE {
 ?x :hasTopic :cruelty.
 ?x owl:sameAs ?dbpX .
 ?law dbp:lawsapplied ?dbpX .
 ?law rdfs:label ?lawLabel .
 ?law rdfs:comment ?abstract .
 ?law dcterms:subject ?subject .
 ?subject rdfs:label ?subjectLabel.
 ?law dbp:holding ?holding .
}
```

To non-specialists, it may not be obvious what the Eighth Amendment means by "excessive bail and fines" or "cruel and unusual punishment". Constitute already links the text of the US Constitution to a large set of topic tags, which documents the "inventory" of the US Constitution. In linking to court decisions and other linked content, we can provide authoritative and contextualized interpretations of constitutional provisions that may be written in more abstract terms. In sum, linking to DBpedia and other open linked data enables users to discover the meaning of complex constitutional provisions and to explore the interaction between legal drafting and legal interpretation.

Linking between Constitute and DBpedia is on-going and experimental work. The authors are in the process of building more robust links between Constitute and DBpedia. Future versions of Constitute will reuse data from DBpedia to populate more elaborate entity pages about the constitutional topics.

Other, more specialized data offers even more promise. For example, the Comparative Law project at Washington University catalogs decisions by a large sample of constitutional courts.[14] These decisions review the constitutionality of legislation and typically refer to particular sections of constitutional text. It is only natural that an analyst of these decisions would want to view the relevant constitutional text, just as analysts of constitutional text would benefit from reading an authoritative interpretation of that text by the court. In short, these sorts of combinations – greatly facilitated by the standardized form of the data – represent opportunities for even sharper insights.

4.3. Data and conceptual enrichment

In any domain there are many ways to classify phenomena. The Constitution Ontology is but one conceptualization of constitutions and their topics. Sometimes an authoritative classification system makes sense, if only to facilitate communication and knowledge

[14] http://cerl.wustl.edu/project/compLaw.php?pg=overview.

accumulation (e.g., the Linnean taxonomy in biology). Sometimes, however, alternative conceptualizations can be layered in ways that enrich our understanding, which is a model we imagine here. For instance, the CCP authors happen to classify the "right to work" as a "social right" rather than an "economic right" and "right to culture" as a "minority right" rather than a "social right". Other domain experts might prefer the alternative classification of these two rights. Or, some might prefer a more nuanced categorization scheme. For example, some might propose an elaboration of the category of social rights that distinguished between those that are merely aspirational and those that are explicitly justiciable.

Formalizing the ontology in OWL provides the transparent infrastructure for exactly this kind of refinement. Web-based ontology editing tools such as Protege allow for something close to conceptual collaboration, or at least the merging of multiple classification schemes. Suppose, for instance, that one prefers a simple categorization scheme, perhaps using three traditional categories of rights: civil and political, socioeconomic, and group rights. A collaborative ontology that does not limit elements to one category could integrate such friendly amendments. Even in the same categorization scheme, elements might pertain to multiple categories, as we describe above.

Another possibility is that someone wants to refine or expand the data itself – not just the ontology – to accommodate alternative or refined conceptualizations. New species of constitutional provisions evolve continually, and the ontology (and corresponding data) should as well. Recently, for example, a legal researcher requested that we add a topic for "Independent Audit Agencies", bodies that responsible for ensuring that governments follow ethical accounting practices. Fulfilling such a request is relatively straightforward. Provided the individual has already collected data on which provisions are associated with the new topic, one simply adds the topic to the ontology and add triples to the data indicating which sections of which constitutions address the new topic.

A relatively large set of individuals are interested or invested in the content of constitutions. Together, an international collective of domain experts has significantly more knowledge of constitutions than the handful of domain experts that work on the CCP. Including this additional knowledge into the data underlying Constitute will undoubtedly improve the site. Fortunately, the use of RDF and an ontology makes this possible by rendering conceptual decisions transparent and facilitating the integration of alternative conceptualizations.

5. Design choices and lessons learned

5.1. XML vs. RDF

Initially, the authors planned to encode the data underlying Constitute using XML. The switch to RDF was prompted by three, interrelated features of the framework. The first is the *consistency* of the syntax. As long as the data are modeled as a graph in RDF, it does not matter what syntax is chosen. In XML, on the other hand, one must decide upon a schema to be used to define such relationships, and as a result, there are a multitude of ways that one can mark-up each subject–predicate–object relationship. On its face, the ability to adapt the XML schemas to the specific applications seems like a benefit of XML. However, for *Constitute*, the level of heterogeneity in the structure of constitutional texts made developing a schema that was both parsimonious and valid extremely difficult. The switch to RDF eliminated the need to find a schema that was suitable for every constitutional text and allowed the authors to focus on other, more important aspects of the project, such as developing the ontology.

The second advantage of RDF is its flexibility. The use of an ontology greatly enhances the flexibility of the underlying data and its conceptual architecture. In particular, the authors needed the ability to enrich the data continually. In XML, depending on the scope of such changes, enrichment might require changing the XML schema, which would imply re-encoding each constitutional text to apply the new schema. In RDF, we only need to change the ontology, either adding or updating the relevant RDF triples. Since we are uncertain what the future holds for Constitute, the flexible data model offered by RDF was a key selling point.

The third reason for switching to RDF is the ability to link to other linked open data. Representing the constitution data in RDF enables the creation of links between constitutions through shared constitutional topics that are represented by URIs. Such linkages create the possibility of inferring taxonomical relationships between constitutional topics as well as the ability to perform subclass reasoning, as shown in Fig. 4. Additionally, it opens the possibility of linking the constitutional data with other datasets in the LOD cloud, such as DBpedia and New York Times.

5.2. Unstructured topics list vs. ontology

Before Semantic Web technologies were used, the relationships between the constitutional topics were not formally represented in any format. After receiving a brief introduction on ontologies and Protege, the CCP team initiated the task of creating the Constitution Ontology. For the CCP team, it was very intuitive to create the taxonomical relationships between the constitutional topics and import the FAO Geopolitical Ontology. The ontology consists of classes, subclasses, object properties and datatype properties. Starting from the constitutional topics, the Constitution Ontology was created in approximately 150 h. Future work involves refactoring the ontology in order to adhere to best practices. For example, it may be preferable to employ SKOS instead of subclasses in OWL. Additionally, the CCP team did not necessarily employ standard vocabulary; subsequent revisions with harmonized labels will enhance connectivity with other projects in the LOD cloud.

6. Usage

In the few months since its launch, *Constitute* is already having an impact. Our primary evidence for this is usage statistics. Since its launch in mid-September 2013, Constitute has received 230,000 unique visitors (as of June 2014). 168,181 of these visitors were within a month of the launch event. After the initial burst of visitors, we have been averaging about 250 new visitors per day (from mid October 2013 to June 2014). The vast majority of new visitors, approximately 70%, have been from Europe and the Americas. In Africa, Constitute is most popular in Northern Africa, having 2360 new visitors between mid October 2013 to June 2014, which is unsurprising given the fact that new constitutions have recently been promulgated in Egypt and Tunisia.

In terms of our target audience—constitutional drafters, we know that drafters in Tunisia were using Constitute, although we do not know how much they used the site. We have also heard reports that Constitute is being used in Yemen and Libya, where NGOs are using Constitute to train potential constitutional drafters. In addition, the United Nations Department of Political Affairs, which is the UN department primarily responsible for consulting on constitution-making, has embedded Constitute into an internal site used by its advisors.

This is very preliminary evidence of its impact. In the near future, we will be performing more rigorous impact assessments that provide information not just about who is using *Constitute* but also how they are using it and how effective it is. In addition, we anticipate that Constitute is the first of multiple applications that will exploit the underlying linked data.

7. Next steps

The features and functionality of Constitute have been stable since September 2013 and the site has been employed by a broad and growing set of users worldwide. The team envisions a set of incremental improvements over the next two years. Version 1 of the application was focused on converting the constitution data into RDF and the creation of a search portal that supports light-weight reasoning. Version 2 will extend Constitute in at least two directions relevant to the Semantic Web Community. First, the data will be released as Linked Data and the authors will elaborate a set of standard use cases for users, including sample SPARQL queries on the constitute data. Second, Constitute will be linked to other datasets such as DBpedia, the New York Times and other constitutional dataset such as the Comparative Law project. Third, version 2 will extend the semantic features of the search portal, including capabilities for exploratory search. We anticipate that interested users will develop the data, conceptual framework, and applications to fit any number of other objectives. Finally, future versions of Constitute will add the historical constitutions and allow side-by-side comparisons between any two constitutions.

Acknowledgments

Constitute is a partnership of the Comparative Constitutions Project and Google Ideas, with financial support from the Indigo Trust and the IC2 Institute at the University of Texas. Engineering and web-design support are provided by Psycle. The following organizations have made important investments in the Comparative Constitutions Project since 2005: the National Science Foundation (SES 0648288), the Cline Center for Democracy, the University of Texas, the University of Chicago, and the Constitution Unit at University College London. Juan F. Sequeda was supported by NSF Graduate Research Fellowship. Daniel P. Miranker was support by NSF Grant IIS 1018554.

References

[1] T. Ginsburg, Z. Elkins, J. Blount, Does the process of constitution making matter? Annu. Rev. Law and Soc. Sci. 5 (2009) 201–223.

[2] Z. Elkins, T. Ginsburg, J. Melton, The Endurance of National Constitutions, Cambridge, 2009.

[3] S. Das, S. Sundara, R. Cyganiak, R2RML: RDB to RDF mapping language, W3C Recommendation 27 September 2012, http://www.w3.org/TR/r2rml/.

[4] M. Arenas, A. Bertails, E. Prud'hommeaux, J. Sequeda, Direct mapping of relational data to RDF, W3C Recommendation 27 September 2012, http://www.w3.org/TR/rdb-direct-mapping/.

[5] J.F. Sequeda, D.P. Miranker, Ultrawrap: Sparql execution on relational data, Web Semant.: Sci. Serv. Agents World Wide Web 22 (2013) 19–39.