CTEI-2016-08

CTEI WORKING PAPERS

TOWARDS AN INTEGRATED DATABASE OF INTERNATIONAL ECONOMIC LAW (IDIEL) DISPUTES FOR TEXT-AS-DATA ANALYSIS[a]

Wolfgang ALSCHNER[b]§

Aleksander UMOV[c]

Keywords: International economic law, disputes, WTO, investment, database, text-as-data, reference detection

§ Corresponding author

[a] The research leading to these results has received funding from the SNSF project "Convergence versus Divergence? Text-as-data and Network Analysis of International Economic Law Treaties and Tribunals"

[b] Post-doctoral Researcher in International Law, Graduate Institute of International and Development Studies & World Trade Institute

[c] Graduate Student in Computer Science, Technische Universität München

Centre for Trade and Economic Integration (CTEI)

The Centre for Trade and Economic Integration fosters world-class multidisciplinary scholarship aimed at developing solutions to problems facing the international trade system and economic integration more generally. It works in association with public sector and private sector actors, giving special prominence to Geneva-based International Organisations such as the WTO and UNCTAD. The Centre also bridges gaps between the scholarly and policymaking communities through outreach and training activities in Geneva.

www.graduateinstitute.ch/ctei

Towards an Integrated Database of International Economic Law (IDIEL) Disputes for Text-as-data Analysis

Wolfgang ALSCHNER[a,1] and Aleksander UMOV [b]

[a] *Post-doctoral Researcher in International Law, Graduate Institute of International and Development Studies & World Trade Institute*

[b] *Graduate Student in Computer Science, Technische Universität München*

Abstract. This paper introduces an infrastructure for the analysis of legal metadata and textual data on international investment and trade disputes. The developed database architecture consists of three main components: (1) a WebCrawler of two key web sites for international economic law dispute information; (2) a document analyzer to transform PDFs into text files, identifying structure and footnotes within document, finding references to other disputes and storing texts as XML; and (3) multiple user interfaces to allow different user types to access the data. The architecture allows users to launch metadata queries and/or to investigate textual corpora. It therefore provides a versatile new framework for international economic law research from various angles and disciplines.

Keywords. International economic law, disputes, WTO, investment, database, text-as-data, reference detection

1. Introduction

Scholars pursuing text-as-data research in international economic law face serious obstacles in obtaining adequate data. Existing non-subscription online databases primarily provide full texts in pdf format that are optimized for human inspection, but not for computational analysis. Conversely, corpora specifically conceived for computational analysis tend to be research-question-specific and thus limited in scope, time or subject matter making them less valuable for other researchers. As a result, there is a need for databases that combine the best of both worlds being general in scope and open to other researchers, yet tailor-made for the computational analysis of textual corpora. As part of a Swiss National Science Foundation (SNSF) funded project,[2] we have built such a database for international economic law disputes. This paper describes the purpose, structure, implementation and deployment of the database.

[1] Corresponding Author, Center for Trade and Economic Integration (CTEI), Graduate Institute of

[2] SNSF project "Convergence versus Divergence? Text-as-data and Network Analysis of International Economic Law Treaties and Tribunals".

2. Background

Inspired by other efforts to create integrated systems for legal data analysis [1] and in parallel to ongoing research collecting and processing information on international economic law treaties [2], our project aims at building a database of international economic law dispute data. It covers two subject matters: (1) international trade law disputes, information of which is obtained from the World Trade Organization (WTO) web site (www.wto.org) and (2) international investment law dispute data collected from www.italaw.com and supplemented by information from the International Centre for Settlement of Investment Disputes (ICSID) web site (https://icsid.worldbank.org).

The database caters to two complementary research needs. First, it enables the continuous and holistic analysis of metadata provided by the source web sites. By centrally storing information on disputes and documents it facilitates analysis of disputes across subject matters, disputants, time and applicable law. To account for new data and evolving research questions, the database was designed to allow for regular updates and possible extensions. Second, the database provides an infrastructure for the text-as-data analysis of a wide range of dispute settlement related texts from judicial awards to communications by the party offering data for a wide array of future research questions. This combination of comprehensive metadata and marked-up textual data thus sets this database apart from other research databases on international economic law disputes.

3. Architecture

The architecture designed in this project consists of three core modules (data storage, document and metadata retrieval, and document analysis) as well as a user interface module. There is a focus on modularity and extensibility for the whole system, as well as for each module. During the project, the database system was developed based on this architecture using Python.

3.1. Data Storage

The central part of this project is a relational database containing all of the collected information. Different types of disputes and documents, including their metadata, as well as linkages between them are supported by the data model. All other modules of this project retrieve and store their data in this database.

3.2. Document and Metadata Retrieval

To retrieve the large amount of data necessary for building up the envisioned database from the source web sites mentioned above, a web crawling and scraping module was implemented. This module retrieves all available dispute and document metadata by accessing the web sites and stores it in the database. Additionally, some pre-processing is conducted on the metadata and all document PDF URLs are retrieved for later use.

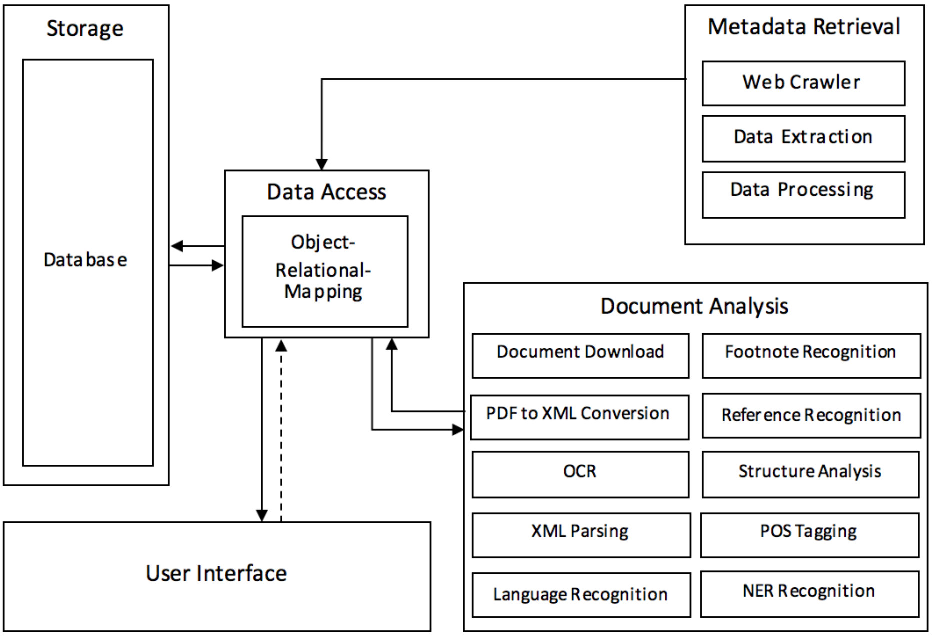


Figure 1. Reference architecture for the Integrated Database of International Economic Law Disputes

3.3. Document Analysis

The document analysis module is by far the most complex module in this project. It is built in Python in a modular fashion with extensibility in mind and provides various features for extracting data out of PDF documents. Features include:

- Parsing PDF documents and converting them to an XML format
- Detection of image-based PDFs and OCR text recognition
- Inclusion of various natural language processing tools, such as part-of-speech tagging (POS) and named entity recognition (NER)
- Footnote recognition
- Paragraph identification
- Headings and structure recognition
- Reference recognition and resolution

3.4. User Interface

There are currently three different ways for users to access the data contained in the database. Each caters to a different skillset.

- A simple web interface which allows browsing the available data.
- A Python Object-Relational-Mapping (ORM) interface which allows users familiar with the Python programming language to build programs based on the data available in the database.

- An extensive database web interface which allows to access the database using SQL queries. This interface is ideal for analyzing relations between data contained in the database. It allows to filter results and to export them in various formats (CSV, JSON, XML).

4. Data Model

On the metadata level, the data model is built around disputes and documents, their attributes, and relations between them. All available metadata from the sources was taken into consideration for the design. Additionally, extracted metadata such as footnotes and references are also contained in this data model.

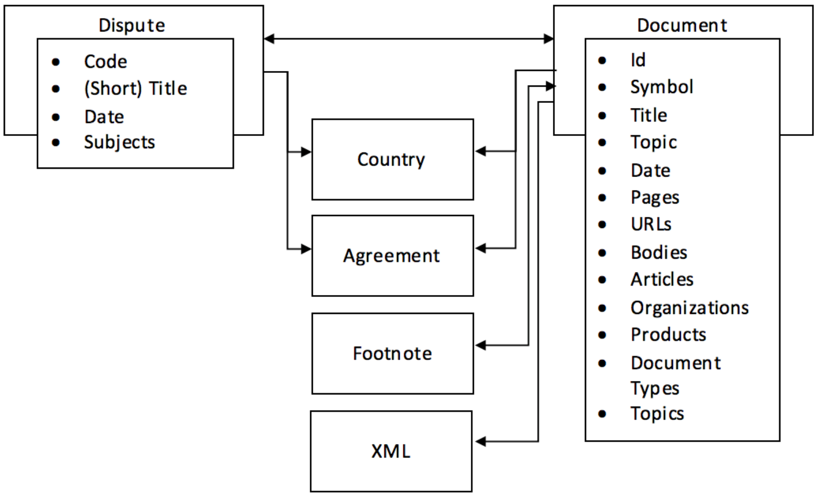


Figure 2. Simplified Data Model of WTO Metadata

On the document level, the data model is a nested tree structure containing pages, headings, paragraphs and footnotes. This structure allows to store the documents in an accessible XML format and to work with them for text analysis.

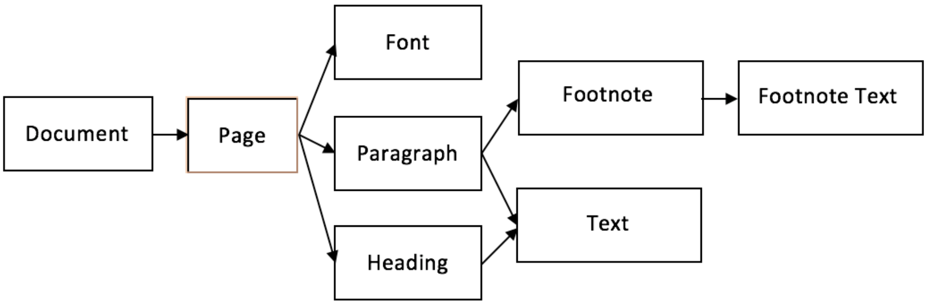


Figure 3. Document Data Model

5. Methodology

To scrape the necessary data, we analyzed the HTML documents of the web sites we were interested in, and built appropriate CSS and XPATH selectors. Additionally, we made use of regular expressions to extract data from the crawled web sites. For example, on the WTO's web site, the links to multi-part documents were stored inside the on-click attribute of an <a> tag as shown in figure 1.

```
<div class="hitEnFileLink><a onclick="var
w=window.open('http://...', ...)"
class="FEFileNameLinkResultsCss" /></div>
```

Figure 4. Example of interesting data stored non-trivially inside an attribute.

```
onclick = document.css('div.hitEnFileLink
a::attr(onclick)').extract_first()
match = re.search(r"^var w=window\.open\('([^']*)", onclick)
if match:
        url = match.group(1)
```

Figure 5. Data Extraction approach for the example in Figure 2.

By combining these two methods, we could extract all the data we needed from the web pages.

5.1. PDF to XML conversion

We analyzed multiple PDF to XML conversion solutions regarding their quality, including *PDFBox*, *pdf2json*, *pdf2xml*, *Adobe Acrobat* and *pdftohtml* (from poppler-utils). The analysis was conducted using manual inspection of multiple resulting XML files. Adobe Acrobat provided the best results, followed by pdftohtml. Since we focused on using open-source software, we decided to use pdftohtml. The resulting XML consists of an array of page-tags, each containing fontspec and text tags. The fontspec tags define the fonts that are used in the document, specifying their ID, size, font family and color. There is no built-in support for bold, italic, and underscored fonts. Each text tag has five attributes, four that define the position and size of the text box in the PDF, and one that specifies which font is used. The `pdftohtml --xml` conversion automatically extracts all contained images and links to them in the resulting XML file. The images are then passed to tesseract for optical character recognition (OCR). The recognized text is then inserted into the XML file inside the image tag. This allows the system to work with both image- and text-based PDFs.

5.2. XML Conversion and Handling

Lxml is used for parsing the resulting XML files. We also considered Python's built-in xml.etree.ElementTree and BeautifulSoup. ElementTree was not able to parse invalid XML files, which pdftohtml produced rarely, but regularly. BeautifulSoup, on the other hand, managed parsing invalid XML files very well, but did not manage to parse large XML files. With lxml, all XML files could reliably be parsed. After parsing, an XML file can be worked with as a Python object in a nested-tree structure. For each XML tag, it is possible to iterate over all or specific subtags, to access attributes, to move or delete it and to change its name, attributes or value. Finally, the tree can be written back to disk as an XML file.

5.3. Multi-Part Document Handling

As some documents from the WTO consisted of multiple PDF files, the conversion also resulted in multiple XML files. In that case, the first step after parsing an XML file is to merge multiple XML files into one. To do this, the pages of all but the first documents are appended to the first document. Additionally, the used fontspecs are compared, deleting duplicate fontspecs and changing the ID of new fonts that collide with existing ones. To reflect these changes, the font attribute of all affected text tags are also updated.

5.4. Cleaning Up

In this step, all data we were not interested in is removed from the document. This includes empty text blocks, tables of contents and page numbers. Tables of contents are recognized using regular expressions looking for the characteristic dot lines. Page numbers are found by looking at the first text block from the top and from the bottom. If one of them contains a number and the next page also contains a number at the same place that is higher by one, it is assumed to be a page number block. Finally, for WTO documents only, the characteristic page headers containing the document symbol are deleted.

5.5. Structure Analysis

To be able to analyze the structure of documents, we implemented support for paragraph and heading recognition. Text blocks that are not too far apart vertically and horizontally and use a similar font are assumed to be in one paragraph. The numbers at the beginning of each paragraph are parsed and assigned as an attribute to the paragraph. This allows text analysts to link to specific paragraphs. Additionally, the different font and numbering styles of headings are used to define heading types. Based on the order in which different heading types appear, it is possible to determine the hierarchy of these types. This approach however, is hindered by badly formatted documents or documents with missing pages. Therefore, it is currently not supported by default.

5.6. Reference Recognition

One of the most interesting and challenging features of this project was to recognize references to other documents and disputes. To do this, we first focused on identifying footnotes, as this is where most references can be found. This was done using a mix of font size/style analysis and text position analysis. Using this approach, we were able to identify almost all footnotes. The next step was to analyze the text found in the footnote. Here, we chose different approaches for trade law disputes and investment law disputes.

5.6.1. Trade Law Disputes

The first approach is to look for WTO document symbols (such as WT/DS375/R) using regular expressions. This covers a large share of references already. Additionally, by constructing a list of all short titles of WTO disputes and comparing it to the footnote's text, references to disputes without a document symbol can be resolved. By identifying keywords such as "Appellate Body report" or "Panel report" close to the short title in the footnote, these references can even be matched to a single document. This is done by accessing the database and looking up the document type in relation to the dispute.

5.6.2. Investment Law Disputes

Due to a missing central administration entity for all investment law disputes, documents in these kinds of disputes were harder to analyze. As with trade law disputes, the first step was to identify document symbols (e.g. ARB(AF)/07/4). However, this approach was not as successful. Thus, we focused on trying to match the names of the disputes. Since there is a lot of variation in the citation style for investment law disputes, simple string matching was not possible. A more complex approach for matching dispute names had to be developed during this project. First, since the term *versus* abbreviated as "v." commonly separates the two disputing parties in the middle of the dispute name we use it to split the string. Then, for each party, a self-developed fuzzy matching algorithm based on the Levenshtein distance is conducted. Finally, if the aggregated distance is below a given threshold, a match is accepted. The source code for the scorer function of this algorithm can be found in the appendix.

6. Examples

The database allows users to investigate research questions that would otherwise take months of manual labor to answer. For example, researchers may be interested how precedent connects different international investment agreements (IIAs). Investment disputes are filed under one of more than 3000 IIAs. In spite of this fragmentation, arbitral tribunals and parties routinely refer to investment case law rendered under other IIAs as precedent. In order to trace how precedent connects distinct agreements, researchers in the past would have to manually (1) determine under which treaty a decision was rendered, (2) locate references to other awards in that investment decision, (3) identify which precedent is being referenced and (4) determine under which treaty that decision was rendered. One SQL query now does the same work in a matter of seconds.[3] The query generates over 14000 connections and reveals the North American Free Trade Agreement (NAFTA) as the most interconnected IIA.

[3] select sourcedispute.case_treaties as source, targetdispute.case_treaties as target from document, dispute as sourcedispute, dispute as targetdispute, footnote, dispute_reference where document.dispute_id =

7. Issues

Several issues could not be fully resolved, of which two stand out. First, investment documents come in a myriad of different formatting, since unlike at the WTO they are not generated through a single institution. This made structure recognition or footnote detection particularly difficult. Second, both main data sources contain PDFs of images. While the OCR can convert these images into texts with varying success (depending on the quality of the original image), structural information (e.g. font style) is partially lost in that process. Better OCR software may resolve the issue.

8. Evaluation

We created a WTO citation network to evaluate the performance of the database by replicating earlier research published by Joost Pauwelyn in 2015 using a different dataset [3]. We investigated the number of outward and inward citations per WTO dispute report and found that out of the top-ten most citing and most cited Appellate Body disputes five disputes also featured in the top-ten list of Pauwelyn's research article. Remaining differences are due to a range of factors including different counting methodology, different years investigated, but also some shortcomings in our data. The reference recognition failed to detect citations in five Appellate Body reports and 24 Panel reports despite the existence of references therein. Future work will investigate possible causes. Overall, however, this preliminary evaluation suggests that the database allows us to satisfactorily engage in the type of analysis envisaged.

9. Conclusion and future work

In its current form, the database serves users with some knowledge of SQL queries or programming. Yet, in order to make it accessible for lawyers without either of these skills, we intend to build an expandable online architecture to run queries and access information in a user-friendly manner. That architecture will originally link the two databases resulting from this project. In a second step, it will be extended to encompass affiliated datasets created by other researchers. The ultimate goal is to enable legal users to conduct metadata but also textual analysis on the entirety of international economic law data from treaties to disputes and from trade to investment issues.

References

[1] B. Waltl, M. Zec and F. Matthes, A Data Science Environment for Legal Texts, *Legal Knowledge and Information Systems: JURIX 2015* (2015).

[2] W. Alschner & D. Skougarevskiy, Mapping the Universe of International Investment Agreements, *Journal of International Economic Law* **19**(3) (2016).

[3] J. Pauwelyn, Minority Rules: Precedent and Participation Before the WTO Appellate Body, *Judicial Authority in International Economic Law* (eds. J. Jemielniak, L. Nielsen & H. P. Olsen) (2015).

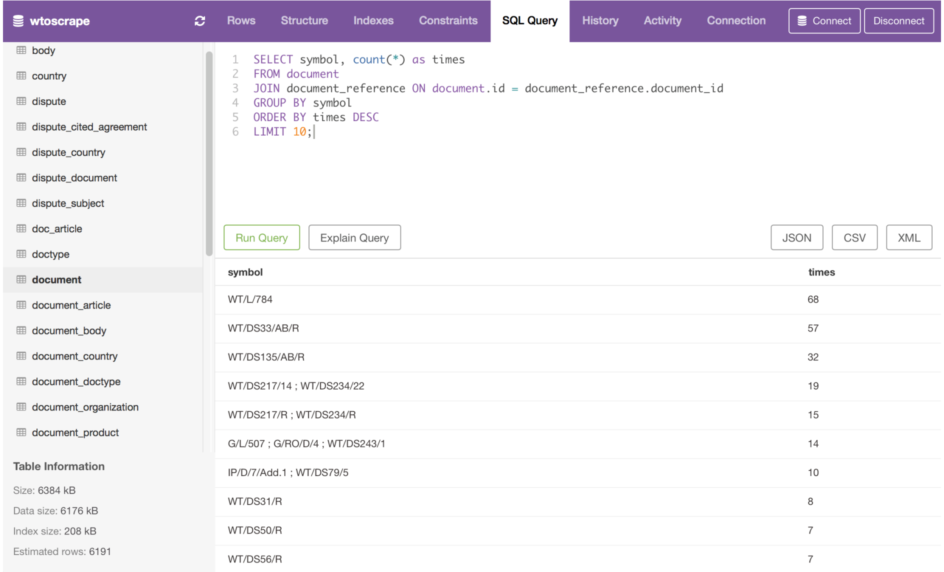
sourcedispute.id and footnote.document_id = document.id and dispute_reference.footnote_id = footnote.id and dispute_reference.dispute_id = targetdispute.id;

10. Appendix

Appendix 1. Scorer function for determining the similarity of two Investment Law Disputes short titles

```
from fuzzywuzzy import fuzz
import fuzzywuzzy
# Returns the similarity score of two dispute title strings on a scale of
0 to 100.
def scorer(x, y):
          # Weight for first party matching if nation party could not
confidently be matched.
          weightratio = 0.5
          # Threshold score of country matching to be considered correct.
          bmin = 70
          # Seperate the two parties in both strings by splitting at the
'vs.'.
          matchx = re.search(r' vs?\.? ([^,:]*)', x)
          matchy = re.search(r' vs?\.? ([^,:]*)', y)
          if not matchx or not matchy:
                    return 0
          xa = re.sub(r' {2,}', r' ', x[:matchx.start()].strip()).lower()
          xb = re.sub(r' {2,}', r' ', matchx.group(1).strip())
          ya = re.sub(r' {2,}', r' ', y[:matchy.start()].strip()).lower()
          yb = re.sub(r' {2,}', r' ', matchy.group(1).strip())
          # If the first party substring of one string is more than three
times longer than
          # the first party substring of the other, use partial matching,
else weigthed.
          len_ratio = float(max(len(xa), len(ya))) / min(len(xa),
len(ya)) if xa and ya else 0
          if len_ratio > 3:
                    scorea = fuzz.partial_ratio(xa, ya)
          else:
                    scorea = fuzz.WRatio(xa, ya)
          # Get the partial matching score of the nation parties.
          scoreb = fuzz.partial_ratio(xb, yb)
          # If it's higher than the minimum, we assume the country to be
correct and return
          # just the similarity of the first party.
          if scoreb > bmin:
                    return scorea
          # If it's lower, we return a weighted score.
          return int((scorea*weightratio+scoreb)/(1.0+weightratio))
```

Appendix 2. Screenshot of Database Web Interface



```sql
SELECT symbol, count(*) as times
FROM document
JOIN document_reference ON document.id = document_reference.document_id
GROUP BY symbol
ORDER BY times DESC
LIMIT 10;
```

symbol	times
WT/L/784	68
WT/DS33/AB/R	57
WT/DS135/AB/R	32
WT/DS217/14 ; WT/DS234/22	19
WT/DS217/R ; WT/DS234/R	15
G/L/507 ; G/RO/D/4 ; WT/DS243/1	14
IP/D/7/Add.1 ; WT/DS79/5	10
WT/DS31/R	8
WT/DS50/R	7
WT/DS56/R	7

Table Information

Size: 6384 kB

Data size: 6176 kB

Index size: 208 kB

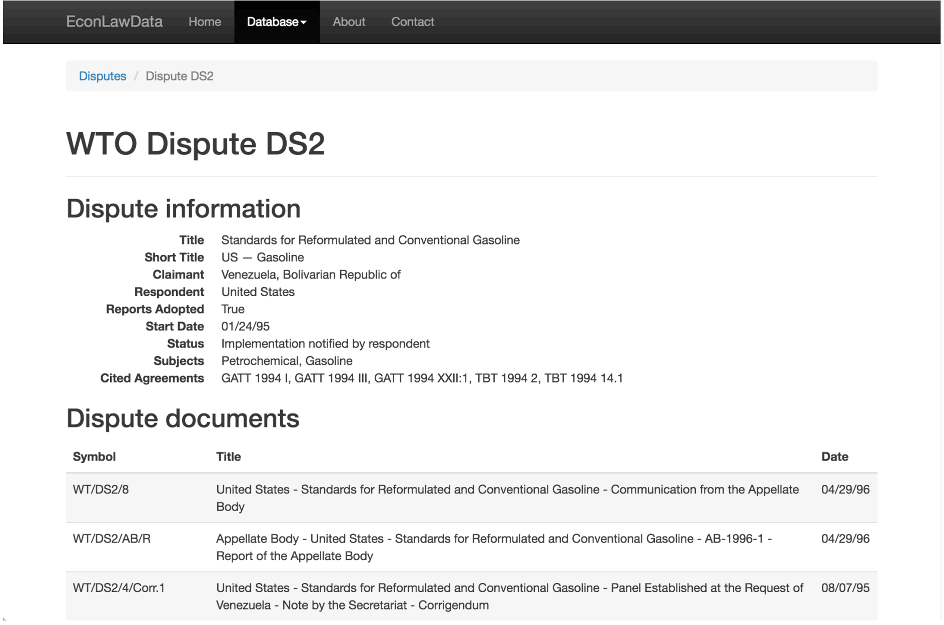
Estimated rows: 6191

Appendix 3. Example of Database Access in Python

```python
from connector import DatabaseConnector
from models import *
# Establish a database connection
con = DatabaseConnector()
# Create a session
session = con.Session()
# Retrieve all disputes over automobiles.
disputes =
session.query(Dispute).join(Dispute.subjects).filter(Subje
ct.name == "Automobiles")
print "There are %d disputes over automobiles." %
len(disputes.all())
# Sum up all pages in all documents in these disputes.
total_pages = 0
for dispute in disputes:
        # Access the documents of the current dispute.
        documents = dispute.documents
        # Sum up all pages of documents in this dispute.
        total_pages += sum([document.pages_en for document
in documents])
print "Overall, these disputes contain %d pages." %
total_pages
```

```
# Output:
# There are 23 disputes over automobiles.
# Overall, these disputes contain 4223 pages.
```

Appendix 4. Example of Simple Web Interface



WTO Dispute DS2

Dispute information

Title	Standards for Reformulated and Conventional Gasoline
Short Title	US — Gasoline
Claimant	Venezuela, Bolivarian Republic of
Respondent	United States
Reports Adopted	True
Start Date	01/24/95
Status	Implementation notified by respondent
Subjects	Petrochemical, Gasoline
Cited Agreements	GATT 1994 I, GATT 1994 III, GATT 1994 XXII:1, TBT 1994 2, TBT 1994 14.1

Dispute documents

Symbol	Title	Date
WT/DS2/8	United States - Standards for Reformulated and Conventional Gasoline - Communication from the Appellate Body	04/29/96
WT/DS2/AB/R	Appellate Body - United States - Standards for Reformulated and Conventional Gasoline - AB-1996-1 - Report of the Appellate Body	04/29/96
WT/DS2/4/Corr.1	United States - Standards for Reformulated and Conventional Gasoline - Panel Established at the Request of Venezuela - Note by the Secretariat - Corrigendum	08/07/95