Knowledge Graphs for Legal Domain

Student Name: Udit Bhati
Roll Number: 2019281

Student Name: Vidhi Sharma
Roll Number: 2019286

Student Name: Ayush Garg
Roll Number: 2019030

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering

on May 11, 2022

BTP Track: Research

BTP Advisor
Dr.Raghava Mutharaju
Dr.Vikram Goyal

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **"Knowledge Graphs for Legal Domain"** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr.Raghava Mutharaju and Dr.Vikram Goyal**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

..... **Place & Date: IIITD, May 11, 2022**
Udit Bhati, Vidhi Sharma, Ayush Garg

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..... **Place & Date: IIITD, May 11, 2022**
Dr.Raghava Mutharaju and Dr.Vikram Goyal

Abstract

Our world is becoming increasingly digital, making it necessary for governments to optimize their operations digitally. So, government and lawyers are interested in concepts such as providing legal information via the web and standardized machine-consumable legislation. The idea of 'Rules as Code' allows the writing of legislative and executive rules such as laws, acts, bills, legislation, regulation, judgements, and policy, in a language machines can understand. The legal texts are complex and voluminous, making it difficult for citizens to access them and understand their true intent. Machine consumable legislative rules make it easy for citizens and businesses to understand their obligations. It allows a more consistent application of rules. We will be exploring Akoma Ntoso (AKN) XML and its applications to convert selected master legislation into a machine-consumable format that supports version control, testing, cross-linkages to other documents, internal linkages, updation, document comparison, and automated deployment of data. Also, we will work to reduce the mismatch between rule intent and implementation.

Keywords: Akoma Ntoso, Machine-Readable, single source, knowledge graphs, legislation, document structure, amendment, AKN.

Acknowledgments

We would like to thank our BTP adviser, Dr.Raghava Mutharaju, and co-advisor Dr.Vikram Goyal for their constant support and supervision of our work. We would also like to thank APG Law Foundation and their members who had been constantly helpful in acquiring the domain knowledge, and also helping in evaluating our results.

Work Distribution

- 21 Jan' 2022 - 10 Feb' 2022 : Problem Statement
- 10 Feb' 2022 - 5 March' 2022 : Literature Survey
- 5 March' 2022 - 15 March' 2022 : Discussion with Legal domain Experts and Exploring Akomo Natso XML
- 15 March' 2022 - 30 March' 2022 : Created single source for certain sections of PMLA
- 30 March' 2022- 15 April' 2022 : Analysis and testified by the domain experts
- 15 April' 2022 - 1st May' 2022 : Worked on 4 more sections of PMLA
- 1st May' 2022 - 11 May' 2022 : Conclusion

Contents

Chapter 1

Introduction

Single source for laws consolidates all the legislations across India, including state laws, central laws, ordinances, and amendments. A single source can ensure that all legal information is detailed, easily understandable, and accessible consistently. The purpose of an initiative such as the single source for laws is to enable all the stakeholders dependent on the legislation to understand and interact with the laws that impact their lives. It can help people outside of government to understand their responsibilities, duties, rights and obligations. It can optimize the legal operations of the government by creating a stronger alignment between the intentions with which a rule is made and its practical implementation.

Benefits of Initiating a Single Source for Laws:

- Digital solutions can be built over a single source for laws to improve access and comprehension.
- The single source can be a starting point to a long line of digital tools in the direction and justice sector.

Chapter 2

Motivation and Problem Statement

2.1 Motivation

The legislative and executive branches of the Indian government adopt a vast body of rules, including various laws, acts, legislation, decrees, judgements, executive orders, regulations, decisions, policies, etc. Books, PDFs and multiple publications that state legal information in detail are voluminous and intimidating to read. Also, inadequacies in legislative drafting and ambiguities in policies make these rules prone to multiple interpretations and amendments. The legal information is not easily accessible to the general public. Law-making, dissemination, and collation complexity make the laws difficult to understand. So, the Indian legislature needs a Single Source for all legal information which is detailed and easily accessible.

Legislative rules are written in natural languages that are human-readable but not machine-consumable. Many formats (such as HTML, PDF, XML, JSON etc.) have been explored to make a single source of information for all legal texts. Some of the issues faced were -

- inconsistent formats
- no metadata texts
- difficult for other IT tools to access
- text is not human-readable
- text is unsearchable
- no version control
- no interoperability
- requires manual updation
- no linked cross-references

- no linked internal references

These issues highlight the importance of standardization. It is essential to explore a standardized format that allows the machine-consumable rules to be created, understood, and accessed in a consistent way. Such tools can enable streamlining for accessing legal information better and allow free access to the law.

Many such tools are being developed, piloted, and deployed at the international level, such as the Senate of Italy[**Senate Italy**], European Parliament [**EuParliament**], Luxembourg Gazette [**Luxembourg Gazette**] and The National Archives of UK [**Archive Uk**].

2.2 Problem Statement

We aim to digitally create a single source of Indian laws that is reliable, easily accessible and consistent with the new rules and amendments. The India Legislative Markup (ILM) should be an open standard developed on Akoma Ntoso standards and should have attributes which would enable the Government to provide official sanctions to the authenticity and reliability of the information.

Requirements and Objectives of India Legislative Markup (ILM) –

- **Document Comparison** - For tracking changes in the law
- **Timeline** - Display what the law was at a given point in time
- **Machine-Readable Format** - laws should be machine-consumable
- **Capable of Quick Updation** - Any officer of Government could upload data
- **Display of Gazette Version** - Link source file with original PDF
- **Linkage to Court Decision** - Linked to any court order pertaining to any clause of the legislation
- **Version Control** - Reliable and seamless
- **Document Types** - Standard to apply to Acts, Bills, Amendments, Regulations, Notifications, Circulars, Directives, judgements etc.

Chapter 3

Related Work

3.1 Case Study of DataLex

The Case Study on DataLex[**1**] explores an exciting approach called 'Rules as Code', which includes the conversion of a legal text (rules, laws, legislations, judgements, regulations), initially in natural language, into a representation that is readable, consumable and processable by a machine. This approach intends to digitalize the legislative system. A program developed on the 'Rules as Code' approach can be programmed to input a user situation requiring legal aid, and produce conclusions that state the actual legislative intent of the legal text on the user input.

The case study suggests that there is a crucial difference between what legislative rules 'say' and what they are taken to 'do'. That is one of the reasons why the success of machine learning in the legal domain has been limited. DataLex avoids trying to illustrate what the rules do and instead represents what they say. Therefore, DataLex works while considering that the resources available for AI developments will be limited[**2**]. DataLex uses the yscript language to represent legal information and legal texts. It has quasi-natural-language syntax and is easy to explore and understand. The yscript interpreter has some sets of rules that makes it capable of producing reports on legal conclusions and legal documents. The ylegis pre-processor helps convert legal text into a yscript rule-set to represent the legislation's structure.

The case study also explores the limitation of legislation as codes. Legislation as codes cannot interpret all legislative terms accurately. So, DataLex provides linkages to the source documents where those legislative terms are used. This helps users to understand what those terms mean. While further research and development are needed for all aspects of the DataLex approach, they demonstrate its potential and value[**2**].

3.2 Study on Knowledge Graphs

Aidan Hogan et al.[**3**] view knowledge graphs as a graph of data intended to accumulate and convey knowledge of the natural world, whose nodes represent entities of interest and whose edges represent relations between these entities. The use of Knowledge Graphs opens up a wide range of tools and techniques to integrate and extract valuable information from diverse data sources. Knowledge graphs also introduce some inductive methods to infer additional knowledge extracted and accumulated by the knowledge graph through reasoning. Knowledge graphs offer diverse structure and granularity through schema, identity, and context representations.

The study[**3**] introduces two types of knowledge graphs: open knowledge graphs and enterprise knowledge graphs. The public has free access to the content of open knowledge graphs. A few examples of open knowledge graphs are DBpedia, Freebase and Wikidata. Enterprise knowledge graphs of some prominent industries include Bing, Accenture, Google, Amazon, eBay, Uber, Facebook, and LinkedIn. Various applications of enterprise knowledge graphs are also discussed in this study.

This paper[**3**] explores different graph data models, graphs datasets, graph patterns and various schemas commonly used in practice. It also gives a detailed description of the role of schema, identity, and context in knowledge graphs. It introduces multiple methods for the creation, enrichment, quality assessment, refinement, and publication of knowledge graphs. It explains the representation and extraction of knowledge using inductive and deductive techniques. Also, many directions for future research and development have been mentioned and explored.

Chapter 4

Research Approach

4.1 Methodology

4.1.1 Methodology1 (Using WebNLG Triples)

While still in the process of the understanding the legal domain , before the interactions with the APG law team , we worked on the understanding the IIITD laws. We tried to make one source destination for the all the laws using the WebNLg text to triple converter. We need to first of all extract the data , then in the given data we need to extract the sentences which are good for the formation of the triples For this Data to feed in the model: Triples (WebNLG triples) to text using a good accuracy model , Classify good or bad sentences .Now, we got the training data. (good or bad sentences) . Create a model, train the above data on that model and classify the test data(input data) as good or bad. This methodology had lot of the drawbacks such as we didnt have the data in the proper format in the first place as well. Then It would be very difficult to get the triples for the entire legislature , which keeps on changing with time to time. Moreover we researched and found other technologies such as datalex and the AKN which had higher values in the given domain.

4.1.2 Methodology2 (Using AKN)

AKN Structure

AKN is an open legal XML standard for parliamentary, legislative and judicial documents. Promoted by UNITED NATIONS Department for Economics and Social Affairs (UN/DESA) in 2004 Adds systematic metadata to documents using ontologically sound approaches.The purpose of Akoma Ntoso is to define an XML representation of the parliamentary, legislative, legal and judiciary documents. The standard captures the universal concepts that are found in this kind of documents. First very important point, the standard is document-centric. The schema is applicable on each major type of document in this domain: legislation, debate, judgment, All Akoma Ntoso documents share the same root element akomaNtoso, under which the specific

document type is selected. In Akoma Ntoso, all document types share the same partitions of document namely coverPage, preface, preamble, conclusions... Not all are mandatory.

The only part that is more specific is the main content as its structure varies: a hierarchical structure for act or bill (called body), an open structure for doc (called mainBody) A special kind of document is the collection document that may contain one or more AKN documentType (including other collections). The documentCollection is the generic type for document that is a collection of documents. Akoma Ntoso uses Internationalized Resource Identifiers (IRIs) as standard mechanisms for referring to documents, languages and concepts on the World Wide Web. A good IRI also has an identification purpose, i.e. it provides a way to universally refer to that resource in a manner that does not change with time, computer systems or software versions. Akoma Ntoso gives a lot of importance to IRIs, and provides rules to systematically specify IRIs for all documents, concepts of the ontology, and even for the markup language itself.

Different types of Document Format

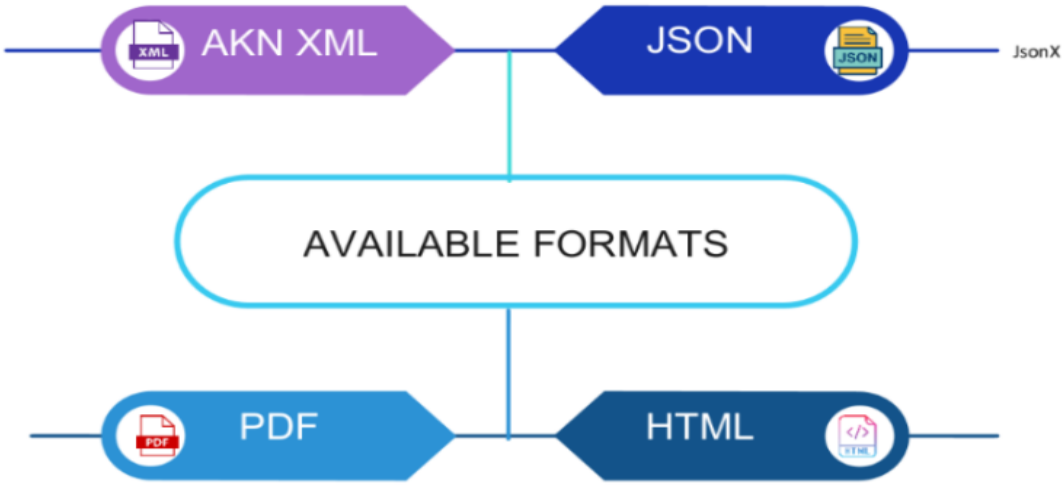

Figure 4.1: Different Formats

Akn benefits over other formats

Why Akoma Ntoso (AKN) XML not HTML/pdf/JSON?

- **HTML** - There is limited support for representing structure, and semantics related to legislation, isn't really explicitly part of the schema. HTML by its nature and use imposes few structural rules and even those are not imposed very strictly.
- **PDF** - No support for structure, nor for semantics related to the structure.
- **JSON** - JSON is great for representing data structures, but not suited for representing documents and their content. It does not distinguish between attributes (i.e. local metadata) and content itself, we need to introduce a named structure called "content" and assume that all other properties are equivalent of attributes.

4.1.3 Flow of Action

- **Choosing and collating the laws** - Picking a master legislation can be a useful start in terms of choosing a law to first collate and consolidate the laws at various levels. The related amendments and subordinate legislation's to the master legislation like rules, notifications, circulars, etc., will be collated and sourced from the various departments and Ministries.

- **Building for machine-readability** - Once a master legislation is chosen and collated, there needs to be a process to make the same machine-readable. This would entail choosing a suitable methodology and technology application to make the consolidated legislation's, including the master legislation and their amendments, machine-readable.

- **Formation of Data Charts from the Legislation** Now for each section we prepare the Data charts which is in the tabular form, where each row corresponds to the various subsections of the Sections. Each column corresponds to the various amendments , notifications and other chnages in the Act.

- **Understanding AKN**

    1. Analyse AKN elements
    2. Understand the Description and Scope of AKN elements
    3. Understand the needs of Indian Legislative Markup
    4. AKN elements in context of the ILM needs
    5. Taking cues from European Union's adoption and improvisation of AKN for EU Legislations

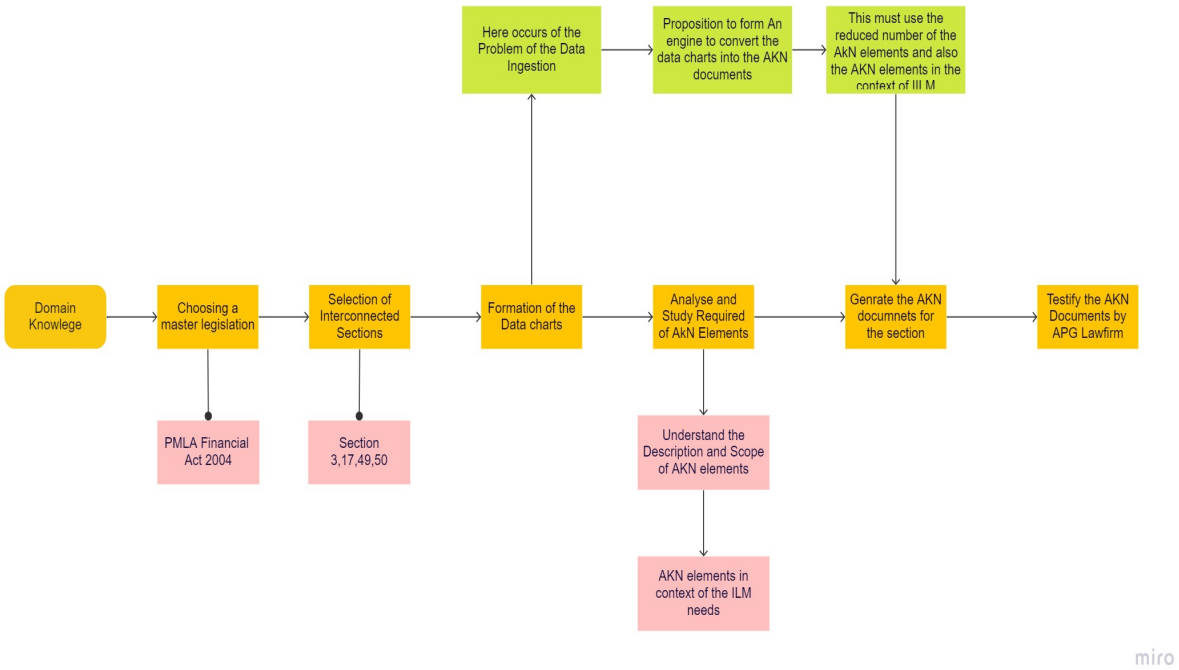

Figure 4.2: The complete flow of work

Chapter 5

Work Done

5.1 Exploring AKN

5.1.1 Document Types

In Akoma Ntoso, there are six basic document structures. However, there are twelve actual document types. This is because some of the document types share their document structure with other document types.

Hierarchical structure

This is the document structure for documents that have a well-defined and regular document hierarchy. It has two types:

1. bill — A proposal for law, regulations, or other normative text.
2. act — An enacted law, regulation, or other normative text.

Amendment structure

This is the document structure for documents that amend other documents. It has only one types:

1. amendment — A proposed or adopted amendment.

Collection structure

This is the document structure primarily used for documents that are a composition of other documents. It has three types:

1. amendmentList — A collection of amendment documents.

2. documentCollection — A more generic composition of other documents.

3. officialGazette — A specific composition of documents that forms the official publication of the legislative body.

Debate structure

This is the document structure This is a document structure that models a recorded debate in the chamber or committee hearing. It has one types:

1. debate — The transcript of a chamber or committee debate.

Judgment structure

This is the document structure for documents that have a well-defined and regular document hierarchy. It has two types:

1. bill — A proposal for law, regulations, or other normative text.

2. act — An enacted law, regulation, or other normative text.

Open structure

This is a catch-all structure that is very flexible and is intended to be used to model documents that are used in the legislative process and don't have the same level of formality or uniformity found in the other document types. It has three types:

1. doc — A generic document type intended to be used whenever another more suitable document type does not exist.

2. debateReport — A summary or report about a debate.

3. statement — A statement from a chamber or other legislative body that either has no or has limited legal effect.

5.1.2 Document Structure

1. meta: mandatory (metadata): identification, references

2. coverPage: Optional

3. preface: document name, number, authors or sponsors, long and short title.

4. preamble: an enacting formula (or clause) which declares the source from which the law claims to derive its authority. enacting formula is preceded and/or followed by recitals.

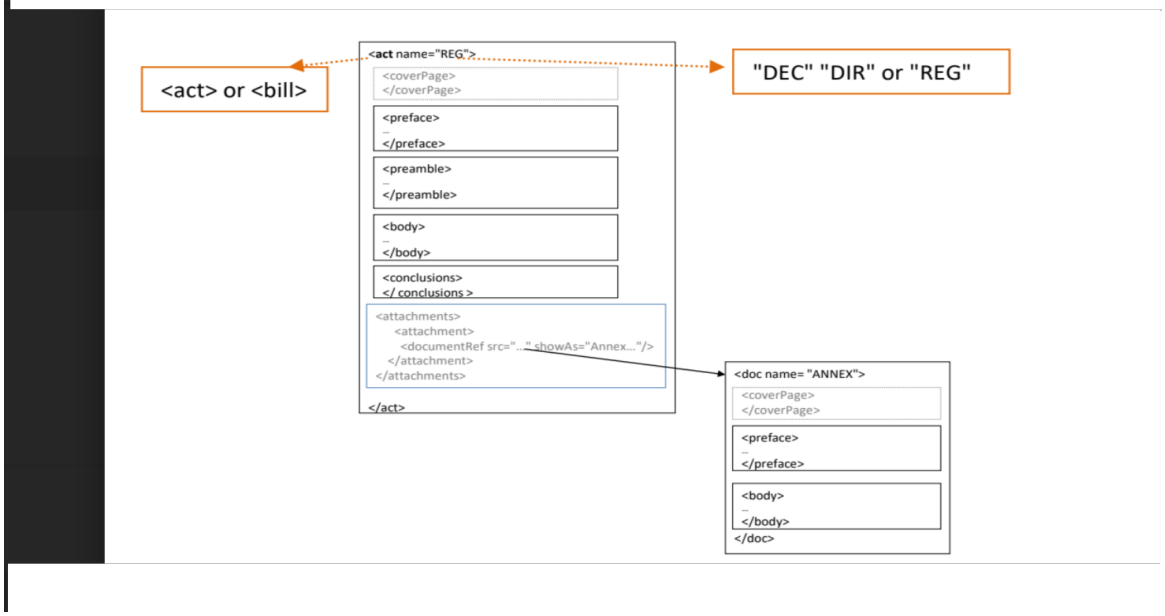

Figure 5.1: f

5. body: includes amendmentBody, collectionBody, debateBody, header, judgementBody, mainBody conclusions: concluding matters such as dates and signatures

6. attachments: annexes, appendices, related bills, or treaties

7. components: schedules or tables which are appended to the main document.

5.2 Amendment in Akoma Ntoso

There are many different amendment styles in Akoma Ntoso.

1. Cute and Bite - It specifies specific changes to be made to the text in insertions and deletions.

2. Restatement in Full - The section (most often) that is to be modified is restated entirely with the changes shown (sometimes) as strikes and insertions. Usually found in state legislation in the United States when amending codes.

3. Side-by-side - The original next of a provision is shown in the left column, and the proposed text is shown on the right with changes sometimes denoted somehow.

```
<mod>
  On <ref>page 1, line 10</ref>,
  strike
  <quotedText>majority</quotedText>
  and insert
  <quotedText>plurality</quotedText>
</mod>
```

Figure 5.2: Cute and Bite

```
<mod>
  Section 1234 is amended to read:
  <quotedStructure>
    <section>
      <num>1234</num>
      <content>
        <p>If a <del>majority</del>
        <ins>plurality</ins> of the
        parties to the arbitration
        agreement are citizens of a
        State or States that have
        ratified or ….</p>
      </content>
    </section>
  </quotedStructure>
</mod>
```

Figure 5.3: Restatement in Full

```
<mod eId="mod_1">
 <quotedStructure eId
="mod_1__qst_1"
    refersTo="#initialVersion">
  (2) In numerous resolutions, the
  European Parliament has expressed
  its concern at the destruction of
  forests and the consequences for
  forest peoples.
 </quotedStructure>
 <quotedStructure id=" mod_1__qst_2"
    refersTo="#newVersion">
  (2) In numerous resolutions, the
  European Parliament has expressed
  its concern at the destruction of
  forests and the consequences for
  forest peoples<ins>, in particular
  indigenous peoples</ins>.
 </quotedStructure>
</mod>
```

Figure 5.4: Side-by-side

5.3 Converting Documents to AKN

XML code for Original PMLA Act of 2005.

```
<akomaNtoso xmlns="http://www.example.org/PreventionOfMoney-LaunderingAct-2005">
    <act contains="originalVersion">
<meta>
<identification source="#somebody"></identification>
    <references source="...">
    <TLCReference xml:id="_REG" name="docType" href="http://www.example.org/originalVersion"
showAs="" shortForm=""/>
    </references>
</meta>
<preface>
    <p>
        <docType>Public Law</docType>61</p>
    <p>
        <docDate date="2005-07-01">Jul. 1, 2005</docDate>
    </p>
    <p> An <docType>Act</docType>
        <docTitle>Prevention of Money-Laundering Act,2002 </docTitle>
    </p>
</preface>
<body>
    <section id="3">
<num title="1">1</num>
    <heading>Offence of Money Laundering</heading>
    <paragraph id="sct1-par001">
        <num title="1">(1)</num>
        <content>
            <p>Whosoever directly or indirectly attempts to indulge or knowingly
assists or knowingly is a party or is actually involved in any process
            or activity connected with the proceeds of crime and projecting it as
untainted property shall be guilty of offence of money laundering.
            </p>
        </content>
    </paragraph>
</section>
</body>
</act>
</akomaNtoso>
```

Figure 5.5: Tree View of PMLA (Original 2005)

Code for the PMLA Amendment 2012

```
<akomaNtoso xmlns="http://www.example.org/PreventionOfMoney-LaunderingAct-Amendment-2012">
    <amendment>
        <meta>
            <identification source="#somebody"></identification>
            <references source="...">
                <TLCReference xml:id="_REG" name="docType" href="http://www.example.org/origina
showAs="" shortForm=""/>
            </references>
            <activeRef href="link to the affected document" />
</meta>
<preface>
</preface>
<amendmentBody>
<amendmentHeading>
<p><affectedDocument id="..." href="...">Section3(1):</affectedDocument>
        </p>
    </amendmentHeading>
    <amendmentContent>
```

```
        <mod eid="mod1">
            <quotedStructure eId="mod_1__qst_1" refersTo="#initialVersion">
                <p>Whosoever directly or indirectly attempts to indulge or knowingly
assists or knowingly is a party or is actually involved in any process
or activity connected with the proceeds of crime and projecting it as untainted property
shall be guilty of offence of money laundering.
                </p>
            </quotedStructure>
            <quotedStructure id="mod_1__qst_2" refersTo="#newVersion">
                <p>Whosoever directly or indirectly attempts to indulge or knowingly
assists or knowingly is a party or is actually involved in any process
or activity connected with the proceeds of crime <ins> including its concealment,
possession,acquisition or
use </ins> and projecting it as untainted property shall be guilty of offence of
money laundering.
                </p>
            </quotedStructure>
        </mod>
    </amendmentContent>
</amendmentBody>
</amendment>
</akomaNtoso>
```

Figure 5.6: Tree View of PMLA Amendment 2012

5.4 Referencing in AKN

5.4.1 How to Refer to documents or provisions

There are three types of references:

1. A reference to another document. This is called a document reference and is expressed as a URI to the document using the URL-based references of the AKN.

2. A reference to a provision within the current document. This is called an internal id reference and is expressed as #eId where eId is the @eId attribute value of the provision being referenced.

3. A reference to a provision within another document. This type of reference is called an external id reference. This is called an external id reference and can be expressed in two ways.

   (a) url#eId is a reference to the document containing the provision with a request to scroll to that element.

   (b) urlēId is referred directly to the provision.

5.4.2 Referencing Elements and Attributes

There are four elements that can be used for references in AKN.

1. a - Similar to the HTML element with the same name, used to refer non-legal texts.
2. ref - Use this element to make a legal reference to a provision or to a document.
3. rref - Use this element to refer to a range of provisions rather than a single provision. If used to refer to a range of provisions within another document wrap an ref around it.
4. mref - It is a grouping construct. Use this element to group a set of ref or rref elements together.

	<a>	<ref>	<rref>	<mref>	<affectedDocument>	<relatedDocument>
@href	Opt.	Req.	-	-	Req.	Req.
• URI pointer to a resource.						
@target	Opt.	-	-	-	-	-
• As with HTML, the browser window in which to display the referred to item.						
@from	-	-	Req.	-	-	-
• An id reference to the initial provision in a range of referenced provisions.						
@upTo	-	-	Req.	-	-	-
• An id reference to the last provision in a range of referenced provisions.						

Figure 5.7: Optional and Required tags

5.4.3 Reference Metadata

Metadata records other types of internal references using the container in the meta element: original: reference to the original version of this document.

1. activeRef: reference to a document that this document amends.
2. passiveRef: reference to a document that makes modifications to this document.
3. attachmentOf: reference to a document that this document is attached to.

4. hasAttachment: reference to a document that is attached to this document. jurisprudence: reference to a document that provides a legal basis to this document.

Chapter 6

Evaluation /Future Goals

6.1 Evaluation

Once a particular Act is converted into the AKN formats, the AKN file was submitted to APG Law firm team. The personnel's at the above firm , had there members to testify the documents. All the tags and the document structure was thoroughly tested and reviewed. We successfully managed to give the AKN files for the Section3 covering all the amendments and its notifications and Section 17. The AKN file of the Section3 and all its amendments has been approved and the that of Section 17 is still under review.

6.2 Future Work

Now the whole idea of the Single Source of Legislation is that we have all our Acts available at one place. When We talk about the acts being in one place , this means that for a particular act, it is cross-linked with all the previous amendments and the notifications. For that we need to have a better format of the data representation so that data can be linked to all linked items. Now for this step we have already identified the AKN format as one of the format to do so. We have also successfully identified a lot of tags for different operations.But now we encounter a huge problem of the data ingestion.To solve this, we need to create the automated data Ingestion engine that would help us to automatically create the AKN files from our data charts. So our major goals from hereon are:

- Identify all the Sufficient tags to cater to all possible variation in the documents
- Create a automated process to convert the Data charts to AKN file format
- Explore the Ontology and its usage in our current Data

The various possible Applications for the same are:

- Explore all the changes in the document at one place.

- Retrieve information related to a particular section or a particular logic. This could be achieved using the introduction of some rule based system.
- View the timeline of the change of the document.

and many more..

Bibliography

1. AustLII Communities. 2022. DataLex. [online] Available at: http://austlii.community/.

2. Mowbray, Andrew and Greenleaf, Graham and Chung, Philip, Law as Code: Introducing AustLII's DataLex AI (November 16, 2021). UNSW Law Research Paper No. 21-81. [online] Available at: https://dx.doi.org/10.2139/ssrn.3971919.

3. Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G., Gutierrez, C., Kirrane, S., Gayo, J., Navigli, R., Neumaier, S., Ngomo, A., Polleres, A., Rashid, S., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S. and Zimmermann, A., 2022. Knowledge Graphs. ACM Computing Surveys, 54(4), pp.1-37. [online] Available at: https://arxiv.org/.

4. Button.is. 2022. Government Rules as Code – A Transformative Idea. [online] Available at: https://www.button.is/post/government-rules-as-code-a-transformative-idea.

5. Vitali, Fabio, and Monica Palmirani. "Akoma Ntoso: Flexibility and Customization to Meet Different Legal Traditions." Presented at Symposium on Markup Vocabulary Customization, Washington, DC, July 29, 2019. In Proceedings of the Symposium on Markup Vocabulary Customization. Balisage Series on Markup Technologies, vol. 24 (2019). [online] Available at: https://doi.org/10.4242/BalisageVol24.Palmirani01.

6. **[Archive Uk]** "The National Achives of UK". [online] Available at: http://leggovuk.s3-website-eu-west-1.amazonaws.com/texts/enactedepublished/akn/ukpga/index.htm

7. **[Senate Italy]** "Senate of Italy bulk of Akoma Ntoso XML files". [online] Available at: https://github.com/SenatoDellaRepubblica/AkomaNtosoBulkData

8. **[Eu Parliament]** "European Parliament". [online] Available at: https://joinup.ec.europa.eu/

9. **[Luxembourg Gazette]** "Luxembourg Gazette example of Akoma Ntoso file". [online] Available at: http://data.legilux.public.lu/eli/etat/adm/pa/2019/07/12/b2143/jo/fr/xml