Definition Disambiguation in the US State Regulations

Chuntain Chi cc2234@cornell.edu

Dylan Walker djw332@cornell.edu

Haonan Peng hp325@cornell.edu

Project Advisors:

Sara Frug

Cornell Law School

sara.frug@cornell.edu

And

Sylvia Kwakye

Cornell Law School

sbk22@cornell.edu

Course Advisor:

Renee Milligan

Department of Computer Science

ram25@cornell.edu

Table of Contents

1. Introduction

1.1 The US State Regulations

1.1.1 What are the State Regulations?

The US State Regulations are the official legal regulations for each of the 50 states in the US. Each individual regulation is developed by the legislative body of its respective state and acts as the official codification of laws that govern that state.

The Legal Information Institute (LII) at Cornell University is an organization that publishes many legal corpora including the US State Regulations. The Legal Information Institute works to increase accessibility to legal information by developing systems to empower users from outside the legal profession to more easily access and understand the laws that govern them.

1.1.2 How are the State Regulations organized?

Each state regulation follows a hierarchical structure in which documents are grouped into different entities such as Titles, Chapters, and Sections. What differentiates the US State Regulations from most federal codifications is that the corpora contained in the State Regulations are uniquely assembled for each of the 50 states. As a result of this there is no consistent nomenclature and hierarchical structure across the states. Each state has its own organization structure. This state-wise discrepancy is evident in both the overarching document organization but also in the standards used to write and format each document.

While each state may follow a different hierarchy pattern, the overall structure for every state is organized into levels that are typically divided by content. The diagram below illustrates the organization pattern for the state of New York:

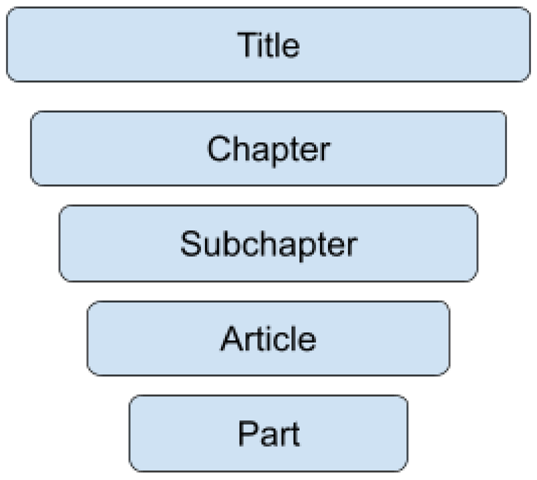


Figure 1: Organization of New York State Regulations

1.2 Definitions in The US State Regulations

Quick motivation and examples of defined terms within these state regulations that highlight where our work will be useful (ex. Act). May be helpful to get some images of the website definition pages from Sara for this. This chapter can be dedicated to showing off the summary statistics we collected (duplicates, # unique terms, terms repeated across states).

- The meaning of defined terms within the state regulations can differ from standard intuition and dictionary definitions. Within the scope of a given defined term, unless otherwise specified, that term shall be interpreted as it is defined in the corpus.
- Definitions for a given term can vary widely across different sections of the same state regulation or across different states and in most cases differ from the common definition of the term.
- When reading through a state regulation it is often difficult to reliably keep track of every defined term and where it applies, making the content of these documents less accessible. If a reader needs to look up a definition they must manually navigate to the corresponding definition section where the term is defined, which itself can be difficult to find if the term has a large scope.

1.3 Definition Disambiguation

Definitions within legal corpora can be tricky and often differ from colloquial meanings as terms can be defined with different meanings across states or within the same state.

State	Term	Definition
Florida	Equine	Any member of the family Equidae, including horses, mules, asses, and zebras.
North Carolina	Equine	Any member of the family Equidae, including horses, mules, asses, and zebras.
Kansas (Agency 106 - title 23)	Written notice	any paper or electronic document relevant to a matter that is filed with the Commission.
Kansas (Agency 106 - title 25)	Written notice	a written notification which is either hand-delivered, facsimile-transmitted or sent by certified mail.

Table 1: Examples of the Same Term Defined Differently

As seen in table 1, the term "Equine" is defined differently in different states. Zebras are viewed as a type of equine in Florida but not in North Carolina. The term "Written notice" is defined differently in the same state of Kansas, even within the same agency.

We do not want to provide users of our website and tools with mixed-up definitions, nor want to see incorrect information used in legal contexts. Hence, our project aims to disambiguate definitions of terms in different regulation under different usage contexts.

1.4 What is Scoping?

The **scope** of a defined term refers to the segments of the corpus for which that definition is to be applied to that term. Every defined term in the state regulations has some associated scope that designates the extent of that term's usage within the regulation. These scopes vary in range from single parts or sections to entire titles. Let's take a look at an example of some defined terms from California under the following location:

Title 1 - Division 1 - Chapter 1 - Article 1 - Section Cal-Code-Regs-Tit-1

The following definitions shall apply to the regulations contained in this chapter:

(1) "APA" means the part of the California Administrative Procedure Act appearing in California Government Code, Title 2, division 3, part 1, chapter 3.5, commencing with section 11340, which generally governs the adoption, amendment, or repeal of regulations by California state agencies.

(2) "Certificate of compliance" means a statement by the head of the rulemaking agency that the agency has complied with the provisions of Government Code sections 11346.2 through 11347.3 prior to the expiration of the effective period of the emergency regulations. After the adoption of emergency regulations, this statement is submitted to OAL for review along with the regulatory text and the rulemaking file.

In this example we see that the terms "APA" and "Certificate of compliance" are both defined in this document. In the header of the document we see that all definitions presented in the document "shall apply to the regulations contained in this chapter". This scoping phrase tells the reader the precise boundaries of where these definitions apply within the regulation.

For this reason being able to identify and disambiguate scoping language is crucial in increasing the accessibility of these documents. Our work on scoping aims to alleviate this issue by systematically pairing defined terms with their scoping language so that accurate definitions can be assigned each time a defined term appears in the text.

1.5 Project Goals

- Develop a system to identify scoping language
- Disambiguate scoping language
- Test other approaches to disambiguation than scoping
- Mark up defined terms in the underlying XML

2. Problem Overview

2.1 Previous Work - Definition Extraction

The data corpus we worked on is the state regulations for all 50 states in XML format. Students from previous semesters performed definition extraction on all 50 states using the same data, and their results are provided to us as dictionaries storing terms with definitions in CSV format. An example entry of these files is provided in Table 2:

idnum	datapath	Tag	Method	Term	Definition
Agency 100 - Article 25 - section 1	filepath	subsect	cues	Office	any place intended for the practice of the healing arts in the state of Kansas.

Table 2: Example Entry from Previous Work

Where idnum is the unique ID number of a regulation's XML file; datapath is the file path to where the data file is stored; tag is the section place in which the term is extracted from in the XML file; method is the method used to extract the term and definition pair; term is the term extracted and definition is the definition extracted corresponding to the term.

Previous work provides us with information about where a term is defined or where a new definition of a term appears for the first time in the regulations of one state. However, we want to know where such a definition applies. For the example in Table 2, we want to know whether this definition of "Office" as "any place intended for the practice of the healing arts in the state of Kansas" only applies to the term "office" in section 1, or if it also applies to the uses of "office" in section 2. Does it cover article 25, or even the entire agency 100? Our work in this project is to try to find the correct definition to every occurrence of terms defined.

2.2 Problem Statement & Task Breakdown

Following the previous work on definition extraction, our team was tasked with marking up defined terms within the US State Regulations. Primarily this involved developing solutions in three major domains:

1. Gathering Summary Statistics on the State Regulations
2. Scope Identification and Disambiguation
3. XML Markup

Furthermore, we break these areas of research into the following set of tasks. Each of these tasks is addressed in the subsequent sections.

Gathering Summary Statistics

- Collect statistics on term frequency and diversity
- Measure the frequency and diversity of scoping language
- Compare and contrast statistics across state corpora

Scope Identification and Disambiguation

- Establish a baseline set of scoping phrases
- Derive a system for identifying scoping language in XML
- Disambiguate scoping language

XML Markup

- Identify defined term instances in XML
- Create a unique "definiendum" XML tag for definition markup
- Create a unique identifier to reference each definition
- Assign definitions to term instances using scope
- Markup term instances in XML

3. Definitions

3.1 Definition Insights

Duplicate Definitions:

- On average for each state corpus 31.99% of all terms are defined more than once
    - “Commissioner” defined 314 times in Connecticut - most of any defined term
    - “Department” is the most defined term in 21 states
        - 154 times in New Mexico
        - 132 times in Alaska
        - 9 times in Missouri

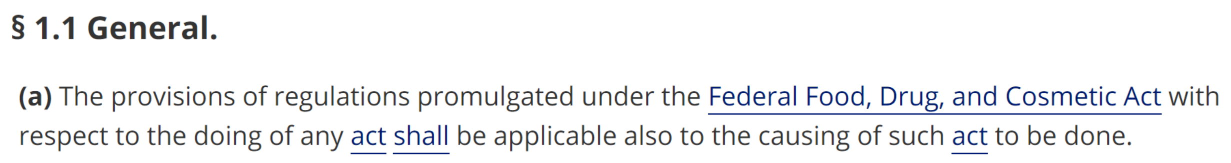
The average duplicate definition percentage further stresses the importance of our work. For terms with multiple definitions, this project will facilitate the determination of right definitions for each occurrence of those terms.

- Only 38.94% of all terms are accompanied by scoping language

    The limited amount of scoping language in the corpus showcases the difficulty of our task. Most of the term occurrences are accompanied without any scoping language, requiring other innovative means to find the corresponding definition.

3.2 Terms with Multiple Definitions

One of the problems for matching the correct definition to term occurrence is to only define the true legally-related occurrences. Attached below is an excerpt from the Code of Federal Regulations.

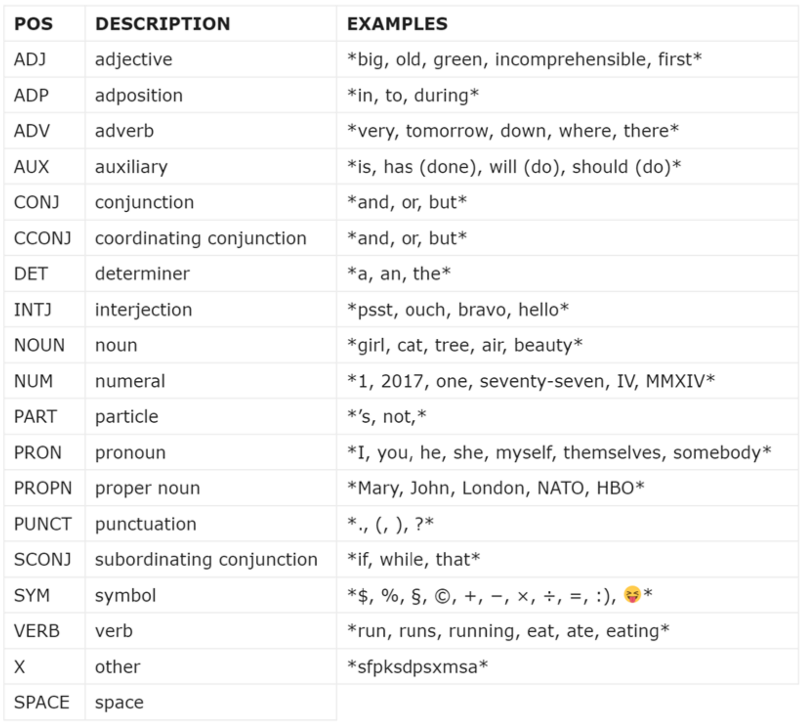


§ 1.1 General.

(a) The provisions of regulations promulgated under the Federal Food, Drug, and Cosmetic Act with respect to the doing of any act shall be applicable also to the causing of such act to be done.

In this example, there are three occurrences of the term “act”. However, only the first occurrence has a legally-related sense, and it is also the only occurrence of “act that needs to be defined here. To tackle this problem, we have devised two approaches.

POS (Part-of-Speech) tag exploration:
First, we explored the POS (Part-of-Speech) tag. We assumed that finding the POS tag for the term occurrence can indicate if the occurrence needs to be legally-defined. In order to efficiently extract the POS tag of term occurrences, we used the spaCy library pipeline. Attached below is a list of all possible POS tags generated by spaCy pipelines.



POS	DESCRIPTION	EXAMPLES
ADJ	adjective	*big, old, green, incomprehensible, first*
ADP	adposition	*in, to, during*
ADV	adverb	*very, tomorrow, down, where, there*
AUX	auxiliary	*is, has (done), will (do), should (do)*
CONJ	conjunction	*and, or, but*
CCONJ	coordinating conjunction	*and, or, but*
DET	determiner	*a, an, the*
INTJ	interjection	*psst, ouch, bravo, hello*
NOUN	noun	*girl, cat, tree, air, beauty*
NUM	numeral	*1, 2017, one, seventy-seven, IV, MMXIV*
PART	particle	*’s, not,*
PRON	pronoun	*I, you, he, she, myself, themselves, somebody*
PROPN	proper noun	*Mary, John, London, NATO, HBO*
PUNCT	punctuation	*., (,), ?*
SCONJ	subordinating conjunction	*if, while, that*
SYM	symbol	*$, %, §, ©, +, −, ×, ÷, =, :), 😝*
VERB	verb	*run, runs, running, eat, ate, eating*
X	other	*sfpksdpsxmsa*
SPACE	space	

We made several simplifications to accelerate our exploration. Based on the assumption that Noun tag and Verb tag will be the most frequent ones, we focused on Noun tag, Verb tag, and Unk (Unknown) tag for our exploration. The Unk (Unknown) tag is a blanket tag that covers any tag other than Noun and Verb.

We obtained the following data from running the custom script on all 50 state corpus:

- POS tags are mostly Noun tag or Unk (unknown) tag
    - Total Noun tag occurrences account for 52.5% of all defined term occurrences.
    - Total Verb tag occurrences account for only 2.8% of all defined term occurrences.
    - Total Unk (unknown) tag occurrences account for 44.7% of all defined term occurrences.

Unfortunately, the exploration is not successful for determining the right term occurrence to assign a legal definition. Most of the tags are Noun tags or Unk tags, and they don't provide a clear distinction between multiple occurrences of the same term.

Word Vector Similarity Comparison:
Then, we devised another approach to explore the similarity comparison of different word vectors. For the pairs of word vectors in similarity comparisons, we categorized them into In-scope and Out-scope comparison.

For instance, for the term "kingfish" in the Florida corpus, we extracted the definition of "kingfish" from Chapter 68B-12. Also, with the given scoping language, we know that this definition of "kingfish" applies for all occurrences in the chapter. For the In-scope comparison, we will extract the context of another occurrence of "kingfish" from the same chapter. In this case, the context will be everything in the "subsect" section of the XML file from which this new occurrence is extracted.

Then, the extracted context will be the input of a program to generate a disambiguated sense of the term "kingfish" using spaCy library. Finally, to calculate the similarity, we will use a word vector similarity calculation in the spaCy library.

Similarly, the process of calculating an Out-scope similarity is mostly the same. The only difference is that the Out-scope similarity will be calculated using a context of the term "kingfish" from a chapter other than Chapter 68B-12.

However, the result of this approach turns out to be disappointing as well. The result is for the entire 50-state corpus is shown below:

- In-scope and Out-scope similarity are close
    - Average In-scope Similarity is about 73.9%
    - Average Out-scope Similarity is about 73.5%

The average In-scope and Out-scope similarity is too close to be of any use for the intended task.

4. Scope Identification and Disambiguation

Here we can highlight most of the work by Dylan and Chuntian on taking the extracted definitions and identifying their scope. This should describe our approaches, experiments, and results for each of the subparts of this problem. This chapter should include parsing XML to find the extracted terms in the text, searching in the text for scoping language, an example of our list of scoping phrases, as well as why/how we could only use "well-defined" scoping. Lastly we should describe how identifying these scopes can help us assign the correct definitions.

4.1 Scoping Phrases

One unique problem posed by our work is that our system is designed to accommodate all 50 state regulations. As described previously, each of these regulations follows different standards in both organization and writing. This results in a lack of consistency when considering the scoping language used by different states. Each state has a unique organization hierarchy as shown above in Section Figure 1 including keywords such as "Title", "Chapter", "Section", "Division", "Agency" etc. Additionally each state is written by different authors under different standards and thus they tend to introduce scoping using a wide variety of phrasing.

These issues call for a robust set of scoping phrases that can generalize well to all 50 regulations. To construct such a set we progressively developed a comprehensive list of scoping phrases seen throughout XML definition pages from each state. We made each phrase "templated" by removing any specific hierarchy keywords, allowing these keywords to later be matched during searching. Below is an example of some of these phrases:

- When used in
- As used in this
- Under the provisions of this
- In this
- For purposes of this
- For the purpose of this
- Within the scope of this
- For purposes of the application under this
- As defined in
- As specified in

In the next section we describe how we use this predefined list of scoping terms to identify scoping language within the XML.

4.2 Identifying Scoping Language

Now that we have established the setup for our scoping problem let's discuss how we go about identifying scoping language. For a given state regulation we first run the definition extraction system on the entire corpus. This produces a set of definitions that we take to be the ground truth. These definitions are stored in a CSV file containing the information on the term and definition pair, as well as the location at which it was extracted (stored as a file ID number). For convenience in our development we modified the extraction code to output an additional column storing the exact file path in the corpus.

Using this CSV file we compile a list of all documents that contain definition items within a given state corpus. We then search the text of each of these XML documents for scoping language from our predefined list (Section 4.1). Once we have found a piece of scoping language within the document we extract the sentence it is contained in for parsing. Lastly we search the sentence for special keywords such as "Title" or "Chapter" corresponding to the different hierarchical elements of the state corpora.

We can then compare any found keywords to the ID number of the current file in order to produce the scope. For example if we have the scoping phrase "As used in this chapter" used within *Title 1 - Chapter 2 - Section 8* then we set the scope as the entirety of *Title 1 - Chapter 2.*

We make the assumption that the presence of any such keywords implies that they are to be used as the scope. For example if we find the keyword "Chapter" in the scoping sentence we assume that scope is solely the current chapter. We also currently make the assumption that scopes span only entire hierarchical elements. This assumption is not always the case and may be pursued in future work. For example in the case of "As used in this section through section XYZ" where the scope encompasses multiple specific sections. In this case our method will use the "Section" keyword and mark the scope to only include the current section, not all sections within the given range.

4.3 Ambiguous Scopes

Across the 50 states there are many different ways in which scoping language is applied to defined terms. For the purposes of our work we divided scoping language into two categories, namely "well-defined" and "ambiguous" Scopes. In this subsection we will discuss the difference between these two types of scopes within the context of our work.

A piece of scoping language is **well-defined** if the range of the scope is clearly stated and easily interpretable. These are the terms that are introduced using a scoping phrase from our predefined list and containing keywords that clearly identify their range such as "Title" or "Chapter".

The following is an example of well-defined scoping language from the Connecticut Regulations. From this example we can see that the Connecticut term "Commissioner" has a clearly defined scope that extends across *Section 1H-8*. This can be resolved by our methods outlined in Section 4.2.

"*As used in this section: 'Commissioner' means the Commissioner of Motor Vehicles or his authorized representative."* - ***Connecticut, Title 1 Section 1H-8***

A piece of scoping language is **ambiguous** if the range of the scope is not easily interpretable and requires additional context and information than the scoping language itself. These are terms that either have no local scoping language or have scoping language that does not contain a resolvable keyword like "Title" or "Chapter".

The following is an example of ambiguous scoping language from the Michigan Regulations:

"As used in these rules: 'Department' means the Michigan department of health and human services." - ***Michigan, Licensing and Regulatory Affairs***

In this Michigan example we see that "Department" is defined using a strong leading scoping phrase "As used in these". However, the actual scope keyword provided "rules" is ambiguous in the context of the Michigan corpus, which only contains Titles and Divisions. As a result of this ambiguity we cannot easily resolve the intended scope using our current approach.

4.4 Handling Well-Defined Scopes

As stated above the well-defined scopes are easily resolvable using the scoping methods we outlined in Section 4.2. For these cases we use our proposed methods directly as stated. That is once we have extracted the well-defined scoping language and verified that it is well-defined we can use the keywords within the scope to identify the coverage of the scope within the hierarchy of the current state regulation. For example if we have the scope "When used in this chapter" for a definition extracted within *Title 3 - Chapter 7 - Section 1* then we use the keyword "chapter" to identify that the scope should be the entirety of *Title 3 - Chapter 7*.

4.5 Handling Ambiguous Scopes

As mentioned above, ambiguous scopes pose a difficult challenge for our current methodology in that we cannot easily resolve their intended scope. However we still need a method of assigning scopes to definitions of this type or else we will miss a significant amount of terms during markup.

Naturally this leads us to employ a default scope for any unresolvable scopes. This default scope should only be applied locally to where the definition was extracted. We infer that the definition is intended to apply to the lowest level parent scope available from the extraction. For example if we have an unresolved scope for a definition in *Title 3 - Chapter 7 - Section 5 - Document 1* then we will infer the scope of the document to be *Title 3 - Chapter 7 - Section 5*.

Additionally, we use this same default scoping method for any defined terms that are completely missing scoping language. In these cases we have no reference so we make the same assumption that the definitions are meant to apply only within the lowest level parent scope from where they were extracted.

5. XML Markup

5.1 Markup Overview

5.1.1 Markup Goals & Standards

Before going into the details of our markup process design and implementation we must first establish the overall goals and standards of the markup process. In this section we will clearly state our markup objectives as well as the standards we must uphold when processing delicate legal information.

Markup Goals

- Find all defined term instances in a given state regulation
- Modify XML documents to indicate defined terms
- Include unambiguous references to corresponding definitions

Markup Standards

As mentioned before legal documents contain sensitive information and their content cannot be modified in any way when displayed to the user. For this reason we must follow strict standards for marking up XML to ensure that no information is lost or corrupted by our system. We focus on 3 key standards by which we developed our methods.

- All Markup definitions should use the "definiendum" tag (see Section 6.1.2)
- XML structure should be maintained for all non-definition elements
- Spacing and content should be consistent between original and markup documents

5.1.2 The Definiendum Tag

During the markup process we need a unique tag that denotes our defined terms and any necessary information for linking them with their associated definitions. To do this we create the special “definiendum” tag. This tag includes three main attributes, the ID, the number of occurrences, and an optional attribute markup = “no”. In summary here are the 3 attributes you may come across:

1. ***ID*** - unique identifier of the corresponding definition
2. ***numOccur*** - Which number occurrence of the definition the current instance is
3. ***Markup = “no”*** - Denotes a definition was matched using a default scope.

Let's briefly discuss the purpose and usage of each of these attributes.
For each definition we need to have a succinct and unique way of referencing that definition. To handle this we assign each definition a unique ID using a hash function that gives us a permanent reference. When we come across a term instance we can then reference this ID to denote the corresponding definition in a compact way.

The ***numOccur*** attribute helps us to distinguish between the different instances of a given definition. During processing, each new instance of a given definition will be numbered ordinally so that we can reference each instance independently, especially those that occur in the same document.

Lastly, as mentioned in Section 4.3 some defined terms have ambiguous scoping language from which we cannot easily infer their scope. As mentioned in that section we work around this by assigning them a default scope to the lowest parent level scope. However we still want to be aware that these scopes are not certain, so for any mark ups of these terms we include an additional ***markup = “no”*** attribute to denote that the matched definition came from an inferred default scope.

Now that we have established our markup goals and the definiendum tag that we will use to mark up terms in XML, let’s discuss how we implemented the actual markup process. Firstly, we need a way of representing the two main entities related to definition assignment, namely Defined Terms and Term Instances

5.2 Representing Defined Terms

A **defined term** is a pair of entities that describe a unique definition within a given state corpus. Each defined term consists of a term and a corresponding definition. For our purposes we use the previous work on definition extraction as our methodology for gathering defined term candidates.

To represent these defined terms we create a class structure with the following attributes to fully capture all relevant components of a particular defined term:

- **term -** Term being defined
- **definition -** Corresponding definition to be used
- **idnum -** Unique identifier corresponding to document location
- **ID -** Unique identifier for this defined term
- **scope -** Scope of this defined term
- **scopeID -** Identifier code for the scope of this definition
- **default_scope -** True/False denoting if default scope is being used
- **nOccur -** Number of times this defined term is used in the text

These attributes allow us to properly represent each definition. Most importantly, we have the term-definition pair that the user will see and interact with, as well as the idnum of the file the definition was extracted from. Each of these three entities are derived from the definition extraction process. We introduce the unique identifier code in order to reference definitions during markup. Additionally we include the scope of the definition in order to assign definitions to new term instances. We include the default scope and number of occurrences attributes for markup purposes.

5.3 Representing Term Instances

A **term instance** is a unique appearance within the text of a given defined term. These are the exact elements we wish to markup.

In order to properly conduct the markup process we need to establish all of the necessary information. To do this we created a class structure with the following attributes:

- **term -** The term being referenced
- **definition -** The corresponding definition of this term instance
- **definitionID -** Unique identifier code of the corresponding definition
- **idnum -** Unique identifier corresponding to document location of the instance
- **start -** Denotes the starting index of the term within its XML element
- **end** - Denotes the ending index of the term within its XML element
- **default_scope -** True/False if definition was assigned using a default scope

- **nInstance** - Denotes which appearance of the defined term a given instance is

We only initialize a new term instance object when a term has been found and paired with an associated definition (see Section 6.5). However, our framework does support initializing instance objects without an already paired definition. This case is not currently in use but could be used in the future for terms that do not find a matching definition.

Once a definition is assigned we initialize the term instance object, storing the definition itself as well as the corresponding unique ID of the definition so that we can reference it using our definiendum tag during markup. Also, if the assigned definition uses a default scope then we make note of this during markup.

We also keep track of the instance number such that every term instance can be uniquely identified by its definition and instance number. For example if a given definition has already been assigned 4 times within the corpus and we encounter a new instance then that new instance will receive **nInstance** = 5. This value will then be stored in the ***numOccur*** attribute of the definiendum tag at markup time.

5.4 Identifying Term Instances

Now that we have established how we represent defined terms in both text and markup, let's discuss our methodology for the markup process. We process each state individually by first running the definition extraction code to get a list of all defined terms in a given regulation (Section 2.1). We then use our scoping methods to assign a scope to each definition (Section 4).

After pairing each definition with a scope we store all of the definitions into a dictionary that maps terms to a list of DefinedTerm objects corresponding to all candidate definitions for that term. Note that within a state regulation a single term is often defined more than once. This dictionary representation allows us to access all possible definitions under the same key. Below is an example dictionary entry from the Kansas corpus:

Example Dictionary Key	Example Dictionary Value
Applicant	1. **Definition:** A person seeking certification as an officer. **Scope**: *Agency 106 - Article 2* 2. **Definition**: A landowner or legal agent applying for financial assistance

	to construct or apply conservation or pollution control practices. **Scope**: *Agency 11 - Article 1*

Table 3: Example Dictionary Entry from Kansas Regulations

After forming this dictionary we begin identifying term instances. To do this we pass through each document of the regulation and cross-compare the document text with our dictionary of defined terms in order to identify all potential term instances. The basis for our matching process is a simple string search within the XML text. However naive string matching has some clear limitations such as plurals and different word senses. In Section 6.6 we discuss some of these challenges and how we handled them.

5.5 Assigning Definitions

After identifying all of the term instances we begin assigning definitions and marking up terms in the corpus. These two processes are done concurrently with the identification process to avoid a second pass through the text. In order to assign definitions we first go through every XML document in a given state regulation. For a particular document we use the following process to perform markup:

1. Identify all terms that have a definition that is in-scope for this document
2. For each element of the document, search for matches between the element text and the in-scope definitions
3. For each match assign definition based on matching scope
4. Markup each match with "definiendum" tag

For step 1 above we take advantage of the unique document idnum, from which we can determine which definitions are in-scope for that document. We scan over all definitions and compare their scopes to the current idnum to gather a list of all in-scope definitions. We then use standard XML parsing to examine each element and use a modified string matching approach to identify the term instances.

Once a match is confirmed we update the defined term object to increment its number of instances. We also create a term instance object for the found term and fill out its attributes using the corresponding definition. Lastly we use the created term instance object to markup the term with the definiendum tag.

5.6 Markup Challenges

5.6.1 Lemmatization

Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form. Generally speaking, we want to be able to recognize the same word in different forms (comparative forms, plurals, different tenses) as the same word. For instance, “walk”, “walking” and “walked” should be treated as the same word and have the same definition in the same scope.

This problem is solved by the following process, given a defined term that we want to find a matching to and a plain text extracted from the XML file.

1. Lemmatize the defined term string
2. Create a list of tokenized words from the text
3. Create a list of storing the positions of each token in the original text
4. Lemmatize the list of tokens in step 3 to get a list of lemmatized tokens

An example, where the term we want to search for is “facility” and the text extracted is “ fire protection and public safety facilities;”, and the results for each step is as follows:

1. Lemmatized Term: facility
2. List of Tokens: ['fire', 'protection', 'and', 'public', 'safety', 'facilities', ';']
3. List of Positions: [[(1, 5)], [(10, 20)], [(21, 24)], [(25, 31)], [(32, 38)], [(39, 49)], [(49, 50)]]
4. List of Lemmatized Tokens: ['fire', 'protection', 'and', 'public', 'safety', 'facility', ';']

It can be seen from the example that “facilities” is lemmatized into “facility” after step 4, and we can find a matching of the term in the text. Then, using the position indexes from step 3, we are able to store the starting and ending positions of the word “facilities” in its original text, which is position 39 to 49. Note that the positions are zero-indexed, but the first word starts with “1” in the original text because there is an extra space ahead. We want to preserve all spaces and other forms of spacing or other characters due to the preciseness of legal texts.

5.6.2 Subtoken Definitions

One additional challenge we faced in the XML markup process was with subtoken definitions because our current method is fully dependent on string matching. This occurs when a defined term appears as a substring or subtoken within a body of text. Below is an example output from our markup system in which we have a defined term “rec” that appears as a substring within the word “preceding”.

```
<designator> (A) </designator> During the 18-month period immediately
p<definiendum id=“351036315016590851 8” numOccur=“1”>rec</definiendum>
eding the license expiration date, the <definiendum id=“0346167350003974478”
numOccur=“1”>person</definiendum> completed at least 50 credits
```

Since we are using string matching the system identifies every instance of the string “rec” as a defined term. In this example we can see that this was an oversight of our original design. To combat this we used two subsequent approaches. The first approach was to search for the term with spaces appended to either side to ensure that no parts of the term appeared as a substring of any other tokens. This method appeared to work initially however we found that it was not robust enough for special cases, for example when the term starts or ends a sentence, or when the term appears in quotations. As a result of this we bolstered our method to check that the characters directly before and after the term instance are not alphanumeric. This allows our system to properly differentiate terms that appear next to punctuation versus those that appear as subtokens.

5.6.3 Nested Defined Terms

In some instances, we can have nested defined terms where one defined term appears as a word of phrase within the use of another defined term. This can introduce complications in both the identification/definition assignment process as well as in our markup scheme. The table below illustrates an example of a triple-nested definition from Kansas.

Defined Term	Definition
Attendant	a staff person or volunteer who provides direct supervision of a juvenile.
Attendant Care	one-on-one direct supervision of a juvenile who has been taken into custody. Attendant care shall not exceed 24 hours exclusive of weekends and court holidays.
Attendant Care Facility	a boarding home for children at which attendant care is provided.

Table 4: Example Nested Definition from Kansas Regulations

Our primary method for handling nested terms is to sort all defined terms for a given state from longest to shortest before starting the definition assignment and markup process. This ensures that when we go to assign definitions, we will always assign the outermost layer of the nesting to capture the most dominant term. Using the example above, consider we come across a sentence that uses the term “Attendant Care Facility”. While there are technically 3 defined terms here we only want to markup the most relevant one as marking up multiple can lead to contradictory or conflicting meanings. Thus processing the terms from longest to shortest will assert that we always process the outermost definition first. If an in-scope definition exists for this outermost term then we will assign that definition and mark it up in the XML with a “definiendum” tag (see Section 6.1). As we continue to process we skip over any “definiendum” tags to avoid marking up any terms already contained in a processed defined term.

The following example displays the results from our model after applying these changes to handle nested definitions. This is an example of an instance of the nested defined term “attendant care” in the Kansas corpus:

Original XML File:

“Attendant care” means one-on-one direct supervision of a juvenile who has been taken into custody. Attendant care shall not exceed 24 hours exclusive of weekends and court holidays.

Model After Handling Nested Terms:

"<definiendum id="3458623777646826528" numOccur="1" markup="no">Attendant care</definiendum>" means one-on-one direct supervision of a juvenile who has been taken into custody.

6. Future Work

For the future work, we think a more sophisticated scope extraction can improve the quality of the XML file markup. Due to time constraints, we have barely worked on machine learning techniques this semester. We think machine learning techniques can be beneficial when processing terms with ambiguous scoping language. Future improvements using machine learning techniques can include some training sets and testing sets to fine-tune a machine learning model to the scoping task.

Besides the technical advances, opinions from domain experts can also be valuable. For future work, future teams can consult domain experts to better interpret some of the ambiguous scoping languages.

In addition, for future work, future teams can look into ways to improve the efficiency of the current code base. We think there exists innovative ways to complete our task in a shorter time.

7. Conclusions

This semester, we have collected insightful statistics on defined terms, applied a variety of methods to identify scoping language, and developed a system to assign definitions to term instances.

In particular, the Markup program we developed can be used to insert definition tags into XML files under multiple circumstances. The system can work well for terms with a well-defined scoping language, terms with an ambiguous scoping language, as well as terms without any scoping language.