

# Similarity Analysis on Government Regulations

Gloria T. Lau  
Stanford University

Dept. of Civil & Environmental Eng.  
Stanford, CA 94305-4020  
glau@stanford.edu

Kincho H. Law  
Stanford University

Dept. of Civil & Environmental Eng.  
Stanford, CA 94305-4020  
law@stanford.edu

Gio Wiederhold  
Stanford University

Computer Science Dept.  
Stanford, CA 94305-9040  
gio@db.stanford.edu

## ABSTRACT

Government regulations are semi-structured text documents that are often voluminous, heavily cross-referenced between provisions and even ambiguous. Multiple sources of regulations lead to difficulties in both understanding and complying with all applicable codes. In this work, we propose a framework for regulation management and similarity analysis. An online repository for legal documents is created with the help of text mining tool, and users can access regulatory documents either through the natural hierarchy of provisions or from a taxonomy generated by knowledge engineers based on concepts. Our similarity analysis core identifies relevant provisions and brings them to the user's attention, and this is performed by utilizing both the hierarchical and referential structures of regulations to provide a better comparison between provisions. Preliminary results show that our system reveals hidden similarities that are not apparent between provisions based on node content comparisons.

## Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *linguistic processing*; H.2.8 [Database Management]: Database Applications – *data mining*; J.1 [Administrative Data Processing]: Law.

## Keywords

Regulations, Similarity Analysis, Legal Informatics, Text Mining.

## 1. INTRODUCTION

Government regulations are an important asset of our society; ideally, they should be readily available and retrievable by the general public. Industrial productivity can be greatly increased if tools are provided to aid in locating and understanding regulations. For instance, building designers, although more knowledgeable than the general public, have yet to search through the continuously changing provisions and locate the relevant sections related to their projects, then resolve potential ambiguities in their provisions. Inspectors have to go through a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGKDD '03, August 24-27, 2003, Washington, DC, USA.  
Copyright 2003 ACM 1-58113-737-0/03/0008...\$5.00.

similar evaluation process before a permit can be approved.

The inherent nature of multiple issuing agencies also deserves attention. Regulations are typically specified by Federal as well as State governmental agencies and are amended and regulated by local counties or cities. These multiple sources of regulations sometimes complement and modify each other, and at times contradict one another. Designers often turn to reference handbooks that are independent of governing bodies; as a result, the regulations, amending provisions and interpretive manuals together create a massive volume of semi-structured documents with potential differences in formatting, terminology and context.

## 1.1 The Need for Regulatory Information Management

To illustrate some of the research issues in legal informatics and the need for it, we present two examples below. The first example shows two provisions regulating curb ramps in accessible parking stalls [9]. The California Building Code (CBC) [7] allows curb ramps encroaching into accessible parking stall access aisles, while the Americans with Disabilities Act (ADA) Accessibility Guidelines [1] disallows encroachment into any portion of the stall. Here one provision is clearly more restrictive than another, making compliance a non-trivial task without the knowledge of the existence of related provisions.

### Example 1

#### ADAAG Appendix

##### A4.6.3 Parking Spaces

... a curb ramp opening must be located within the access aisle boundaries, not within the parking space boundaries.

#### CBC

##### 1129B.4.3 Equivalent facilitation for parking arrangements

Exceptions: Ramps located at the front of accessible parking spaces may encroach into the length of such spaces ...

Example 2 presents two directly conflicting provisions from the ADAAG and the CBC. This conflict is due to the fact that the ADAAG focuses on wheelchair traversal while the CBC focuses on the visually impaired when using a cane, and is captured by the clash between the term “flush” and the measurement “½ inch lip beveled at 45 degrees”. In his interpretive manual to California accessibility regulations, Gibbens [9] points out that “when a state or local agency requires you to construct the California required ½ inch beveled lip, they are requiring you to break the federal law”, and this clearly should be brought to the user's attention.

In this paper, we describe a regulatory document mining system that utilizes the structure of regulations to enhance a similarity comparison between sections. A brief literature review is

presented in Section 2; feature extraction, which is one of the key elements of the proposed regulation analysis model, follows in Section 3. Our similarity analysis is presented in Section 4, and preliminary results are shown in Section 5. Section 6 gives a brief discussion on future tasks.

**Example 2**

<p><u>ADAAG</u>  4.7.2 Slope  ... Transitions from ramps to walks, gutters, or streets shall be flush and free of abrupt changes ...</p> <p><u>CBC</u>  1127B.5.5 Beveled lip  The lower end of each curb ramp shall have a ½ inch (13mm) lip beveled at 45 degrees as a detectable way-finding ...</p>
---

**2. RELATED WORK**

To aid legal research, one can use traditional textual comparison techniques from the field of Information Retrieval (IR), such as the Boolean model or the Vector model [2], with most being bag-of-word type of analysis (i.e. word order insensitive). This is insufficient since it ignores the structure of regulations, namely that 1) regulations are organized into deep hierarchies, 2) sections are heavily cross-referenced, and 3) terms are well defined within regulations. A good similarity analysis tool for a legal corpus should make use of the structure of regulations mentioned above to provide a better comparison. In addition, traditional IR techniques do not cater to our future development of conflict identification (which is not discussed in this paper). Assuming that the contents of the conflicting sections are related, conflict analysis builds upon a solid similarity comparison between documents, and it requires a deeper understanding of documents rather than the traditional bag-of-word type of similarity analysis. Feature extraction provides some help to this end.

Feature extraction is an important step in repository development when the data dimension is large. It is a form of pre-processing, e.g., combining input variables to form a new variable, and most of the time features are constructed by hand based on some understanding of the particular problem being tackled [3]. Automation of this process is also possible; in particular, in the field of information retrieval, software tools exist to fulfill “the task of feature extraction ... to recognize and classify significant vocabulary items” [3]. The IBM Intelligent Miner for Text [8] and the Semio Tagger [17] are both examples of fully automated key phrase extraction tools.

In addition to comparing the body text of provisions, the heavily referenced nature of regulations provides extra information about provisions, and link analysis [5] is the natural improvement to the similarity measure. Academic citation analysis [4] is closest in this regard; however the algorithm cannot be directly transported to our domain. Citation analysis assumes a pool of documents citing one another, while our problem here are separate *islands* of information where within island documents are highly referenced; across islands they are not. We are therefore in search of a different algorithm that will better serve our needs.

**3. REPOSITORY DEVELOPMENT**

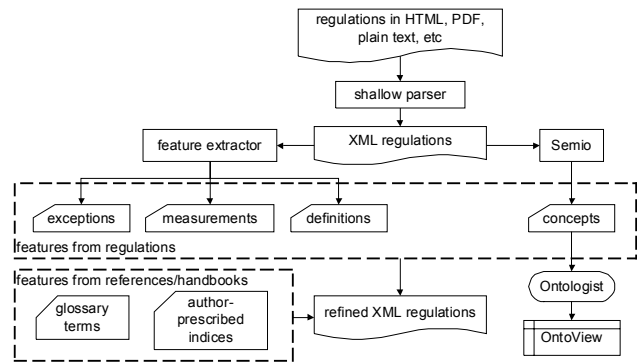
In order to develop a prototypic system, we focus on accessibility regulations, whose intent is to provide the same or equivalent access to a building and its facilities for disabled persons. Our

corpus currently includes two Federal documents: the ADAAG [1] and the Uniform Federal Accessibility Standards (UFAS) [18]. In addition, parts of the International Building Code (IBC) [11] is included to reflect the similarity and dissimilarity between federal and private agency mandated regulations. Related sections from the British Standard BS8300 [6] are included as well to show the difference between American and European regulations.

**3.1 Data Consolidation and Categorization**

Before regulations can be compared, documents are consolidated to a unified format and features that identify similarity are extracted as shown in Figure 1. As for data format conversion, it suffices to say that a shallow parser is developed to consolidate different documents into eXtensible Markup Language (XML) [19] for its capability to handle semi-structured data. The hierarchy of regulations is maintained by properly structuring the XML tags, for example, Section 1.1 is a child node of Section 1, and is thus structured as a child element of Section 1 in the XML tree. References are extracted as well; please see Section 3.2.4 for an example of a reference tag.

As shown in Figure 1, after the documents are parsed into XML format, features are extracted and added to the corpus as described in Section 3.2. Besides reading regulations based on its natural hierarchy, users might find it helpful to browse through an ontology [10] with documents categorized based on *concepts* as well. Semio Tagger is one of several software products that provide such a capability. It first identifies a list of noun phrases, or concept, that are central to the corpus through linguistic analysis. It also provides a concept latching tool to help knowledge engineer to categorize the concepts and create a taxonomy. Documents are thus clustered based on the taxonomy, and users can click through the structure to view relevant provisions classified according to concepts.



**Figure 1. Repository development schematic**

**3.2 Feature Extraction**

This process extracts from regulations the identified features that signal related or similar sections. Some of the features can be applied generically on other sets of regulations, while some are specific to the domain of accessibility; for instance, numeric measurements only make sense in the domain of disabled access code but not in human rights law. In addition, what defines evidence in a certain domain of regulations is also subjected to the knowledge engineer’s judgment. In this context, we strive to be as generic as possible, and all of the extracted features can be easily extended to other engineering domains as well.

Two different sources of features, namely features from within the regulation corpus and features from outside (like those from reference books or engineering handbooks), are extracted with the help of software tools and parsers developed for this task. As shown in Figure 1, features from within the corpus include exceptions, measurements, definitions and concepts, and features from outside domain, for example, engineering handbooks and references, provide domain-specific glossary terms and author-prescribed indices. Each of the features will be discussed in the following sections with an example to illustrate the idea. An example with complete mark-up of the features is shown in Section 3.2.4.

### 3.2.1 Concept and Author-Prescribed Index Tags

The traditional Boolean model or Vector model in IR provides a mechanism for text analysis. Indexing the texts using all of the words, except stopwords (which are very common terms), generates a huge multi-dimensional space with one axis representing one word. Using singular value decomposition, in short SVD, as the dimensional reduction tool, similar words are pulled together as one reduced axis. However, it is still computationally intensive to perform SVD, and the initial sparseness of the matrix is destroyed after dimension reduction. In order to seek an alternative to the bag-of-words vector model and the SVD technique, we use concepts or key phrases, which are relatively simpler compared to traditional index terms and allow us to capture sequencing information on words.

To extract noun phrases from the corpus, the software tool Semio Tagger is used to extract a list of concepts that are identified as important. In our case, the ADAAG and the UFAS together generate just over a thousand concepts. Each provision is tagged with its concepts along with the corresponding count of appearances of the concept (num) as shown below. To increase the number of matches, our system stems both the concepts and the texts in the provision with Porter's Algorithm [14] before matching. Below is an example of a concept and its count.

```
<concept name="stationary wheelchair" num="2" />
```

Concepts are machine-generated phrases that represent a good measure of important terms in the provisions. Another source of potentially important phrases comes from author-prescribed indices at the back of reference books or even the regulation itself; this type of human-written information sometimes can be more valuable than machine-generated phrases. Therefore, index terms from the accessibility chapter of the IBC [11] are tagged against the repository with identical syntax as a concept tag.

A strict Boolean concept or index term match ignores synonyms which can convey important information at times, and work has been done to resolve terminological heterogeneity. As shown in [13], a relatively high accuracy of concept matching is obtained by combining dictionary-based and context-based heuristics. As our corpus grows and so does the list of extracted concepts, matching techniques similar to this can be used to help consolidate the vocabulary, which also aids our future development of conflict identification.

### 3.2.2 Definition and Glossary Tags

In regulation documents, there is often a designated section in an early chapter that defines the important terminologies used in the code, such as Section 3.5 in the ADAAG. These human-

generated terms are more likely to convey key concepts than machine extracted ones such as concepts shown above. In addition, the definition of a term gives the meaning to a term, which is useful for comparisons; an example is shown below.

```
<definition>
<term> Clear </term>
<definedAs> Unobstructed. </definedAs>
</definition>
```

Similarly, engineering handbooks always define in the glossary the important terms used in the field. For instance, the Kidder-Parker Architects' and Builders' Handbook provides an 80-page glossary that defines "technical terms, ancient and modern, used by architects, builders, and draughtsmen" [12]. The difference between definition and glossary tags is that definition comes from the regulation itself, while glossary term comes from sources other than the regulation. Again the syntax is similar to that of definition tags.

### 3.2.3 Measurement Tag

In accessibility provisions, measurements play a very important role; in particular, they define most of the conflicts. For instance, one provision might ask for a clear width of 10 to 12 inches, while another one might require 13 to 14 inches. It is therefore crucial to identify measurements and the associated quantifiers if there is any. In our context, measurement is defined to be length, height, angle, and such. They are numbers preceding units. Quantifiers are noun phrases that modify a measurement, like "at most", "less than", "maximum" and so on. These can be reduced to a root of either "max" or "min", for example, "at most" and "less than" are maximum requirements, thus both reduce to "max".

Our parser first identifies numbers followed by units, like the number 2 followed by the unit lbf as in 2 lbf. The quantifier is an optional attribute in the measurement tag and is identified if it appears in the vicinity of the measurement. Negation, if appearing right in front of the quantifier, is extracted as well and the final quantifier is reduced to its root "max" or "min"; an example is shown below.

```
<measurement unit="lbf" size="2" quantifier="max" />
```

### 3.2.4 Example with Complete Mark-up

An example is presented below with the complete set of feature mark-ups that shows a typical provision from the UFAS, which contains exception, measurement, ref, concept and indexTerm tags in addition to the body text regText tag. All of the extracted information is encapsulated in a regElement node for each section.

#### Example 3

```
Original Section 4.6.3 from the UFAS
4.6.3 Parking Spaces
... at least 96 in ... and an adjacent access aisle...
EXCEPTION: If accessible parking spaces for vans...
Refined Section 4.6.3 in XML format
<regElement name="ufas.4.6.3" title="parking spaces">
  <concept name="access aisle" num="3" />
  <indexTerm name="accessible circulation route" num="1" />
  <measurement unit="inch" size="96" quantifier="min" />
  <ref name="ufas.4.5" num="1" />
  ...
  <regText> Parking spaces for disabled people ... </regText>
  <exception> If accessible parking spaces for ... </exception>
</regElement>
```

## 4. SIMILARITY ANALYSIS

As pointed out in the Introduction, it is rather difficult for anyone to locate any desired material within the jungle of regulations available. Even upon finding a relevant provision for a particular design scenario, clients have to search multiple codes with multiple terms to locate yet more related provisions if there are any. Thus, our goal is to provide a reliable measure of relatedness for pairs of provisions, and to suggest similar sections of a selected provision based on a similarity measure. Here, since a typical regulation can easily exceed thousands of pages, we do not attempt to compare a full set of regulations against one another; rather, a section or a provision from one set of regulation is compared with another section from another set, such as a comparison between Section 4.3(a) in ADAAG and Section 3.12 in UFAS.

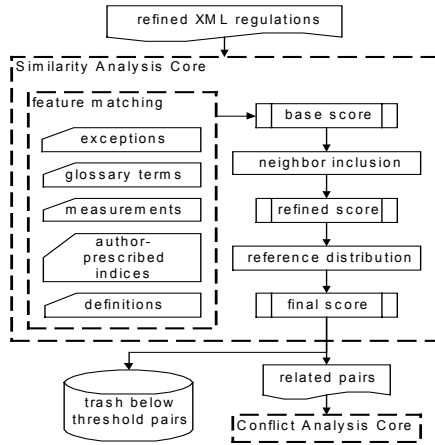


Figure 2. Similarity analysis core schematic

A schematic is shown below in Figure 2 for the similarity analysis core, which takes as an input the parsed regulations and the associated features, and produces as a result a list of the most similar pairs of provisions. The dissimilar pairs are discarded while the most related pairs form the analysis basis for conflict identification (which is not discussed in this paper). The goal of the similarity analysis core is to produce a similarity score, denoted by  $f \in (0, 1)$ , per pairs of provisions. As mentioned in Section 2, our system combines feature matching with the structure of regulations to provide a better comparison in a legal corpus. The process starts with an initial similarity score obtained by feature matching, and the neighboring nodes are compared as well to modify their initial score. The influence of the not-so-immediate neighbors is taken into account by a process called Reference Distribution. Details of each process follow in Sections 4.1 through 4.2.

### 4.1 Base Score $f_0$

The base score  $f_0$  is a linear combination of the scores  $f_i$  from each of the features  $i$ . Scores from features can be weighted differently but for now equal weights are assigned to all features as in Equation 1. The scoring scheme for each of the features essentially reflects how much resemblance can be inferred between the pair of sections based on that particular feature. Here we take concept matching as an example.

$$f_0 = (\sum_{i=\text{features}} f_i) / \# \text{ features } i \quad (1)$$

Concepts are used exactly like the index terms in the vector model [15], where the degree of similarity of documents is evaluated as the correlation between their index term vectors that represent the weights for each index term in the document. The regulations are indexed against these concepts. Each provision is represented as a  $k$ -entry vector where  $k$  is the total number of concepts. A technique similar to the  $tf \times idf$  measure [16] is used for normalization, where term frequency ( $tf$ ) is replaced by concept frequency for intra-cluster similarity, while the inverse document frequency ( $idf$ ) remains the same to account for inter-cluster dissimilarity. The formula to compute the  $idf$  component is taken to be  $\log(n/n_i)$  where  $n$  is the total number of sections, and  $n_i$  is the number of sections the particular concept appears. For two sections, the similarity score  $f_{concept}$  is obtained by comparing concepts given by the cosine similarity between the two concept vectors. Since the cosine similarity is normalized, it always produces a score between 0 and 1. Scoring schemes for other features follow the same idea.

### 4.2 Refined Scores

Score refinement utilizes the tree structure of regulations to refine the base score  $f_0$  between provisions in order to obtain a better and more complete comparison. The immediate neighbors of a node, i.e., the parent, siblings and children of a provision A, are collectively termed the  $p_{sc}$  of A. To help define the terms in a solid sense, we take sections A and B as our point of comparison. By comparing the neighbors of A and B, additional similarity evidences might be revealed; therefore section A itself is first compared with  $p_{sc}(B)$ , and vice versa, to produce the score  $f_{s-p_{sc}}$  based on the initial score  $f_0(A, B)$ . The next refinement takes into account the comparison between  $p_{sc}(A)$  and  $p_{sc}(B)$ , which gives the score  $f_{p_{sc}-p_{sc}}$ . The final score  $f_{rd}$  comes from reference distribution, which compares the referenced sections. Each step is briefly discussed in the follow sections.

Before discussing the details of each refinement techniques, it is crucial to understand the assumption here: we are only interested in increasing the identified similarity but not reducing it. Thus, in the following sections we only consider neighbors or referenced sections that already have higher similarity scores than the pair of interest. The validity of this assumption is built upon what we intend to achieve, and in the case of legal informatics we aim to provide the end user with related provisions and possibly conflicting ones in the future. As a result, it is best to include as much evidence as possible to increase the number of matches, which explains why we are only interested in increasing the similarity score but not decreasing it. For instance, if two sections are entirely the same, but embedded in two completely different neighborhoods, it is important not to decrease their similarity score such that the end user is presented with all relevant provisions.

#### 4.2.1 Neighbor Inclusions: Self vs. Psc

We use an empirical formula to update the score from  $f_0$  to  $f_{s-p_{sc}}$  based on the near neighbors in the regulation tree. Starting from  $f_0$ , the comparison between a pair of provisions (A, B) is first refined by comparing the self node, i.e. node A, with the immediate surrounding of the other interested node, i.e.  $p_{sc}(B)$ , and vice versa, to obtain  $f_{s-p_{sc}}(A, B)$ . Here we are only interested

in  $s$ -psc scores higher than what A and B already share in  $f_0$  in order to reveal greater similarity from the neighbors. We have

$$\begin{aligned} \text{Set } S &= f_0(A, \text{psc}(B)) \cup f_0(\text{psc}(A), B) \\ N &= \text{sizeof}(S) \\ \delta_{GT} &= \sum_{s \in S} (s - f_0(A, B)), s \in S \\ \alpha_{s\text{-psc}} &= \text{discount factor of update} \\ \text{if } (N \neq 0) \quad f_{s\text{-psc}}(A, B) &= f_0(A, B) + \alpha_{s\text{-psc}} \times (\delta_{GT} / N) \\ \text{else} \quad f_{s\text{-psc}}(A, B) &= f_0(A, B) \end{aligned}$$

Here, set  $S$  is the set of similarity scores between section A and  $\text{psc}(B)$ , and between  $\text{psc}(A)$  and section B. The total  $\delta_{GT}$  sums over all  $s$  in set  $S$  which is greater than the original score; thus  $\delta_{GT} / N$  represents the average greater-than score. Clearly  $\alpha$  is always less than one, following our intuition that self-self comparison is more important than self- $\text{psc}$  comparison.

#### 4.2.2 Neighbor Inclusion: Psc vs. Psc

Based on  $f_{s\text{-psc}}$ , the second refinement is to account for the influence of  $\text{psc-psc}$  on sections A and B. Here  $\text{psc}(A)$  is compared against  $\text{psc}(B)$  to refine  $f_0(A, B)$ , which implies that another layer of indirection is inferred and thus the weight of  $\text{psc-psc}$  should be less than that of  $s\text{-psc}$ . We have

$$\begin{aligned} \text{Set } S &= f_{s\text{-psc}}(\text{psc}(A), \text{psc}(B)) \\ N &= \text{sizeof}(S) \\ \delta_{GT} &= \sum_{s \in S} (s - f_{s\text{-psc}}(A, B)), s \in S \\ \alpha_{\text{psc-psc}} &= \text{discount factor of update} \\ \text{if } (N \neq 0) \quad f_{\text{psc-psc}}(A, B) &= f_{s\text{-psc}}(A, B) + \alpha_{\text{psc-psc}} \times (\delta_{GT} / N) \\ \text{else} \quad f_{\text{psc-psc}}(A, B) &= f_{s\text{-psc}}(A, B) \end{aligned}$$

By separating the process of comparing  $s\text{-psc}$  and  $\text{psc-psc}$ , we are enforcing the intuition that the comparison between self (e.g., section A) and  $\text{psc}$  (e.g.,  $\text{psc}(B)$ ) should weigh more than that of  $\text{psc}$  (e.g.,  $\text{psc}(A)$ ) and  $\text{psc}$  (e.g.,  $\text{psc}(B)$ ). Therefore the comparison threshold here is raised to  $f_{s\text{-psc}}$ .

#### 4.2.3 Reference Distribution

To understand the intuition behind reference distribution, we should note that regulations are heavily self-referenced documents, which contributes to the difficulty in reading and understanding them. Our documents, in particular ADAAG and UFAS, are heavily self-referenced but not cross-referenced: they do not reference each other or outside materials as much. For instance, sections in the ADAAG reference other sections in the ADAAG, but do not reference the UFAS or others.

With this understanding in mind, it is easy to explain the process of reference distribution. The hypothesis is that two sections referencing similar sections are more likely to be related and should have their similarity score raised. Therefore, the process of reference distribution utilizes the heavily self-referenced structure of the regulation to further refine the similarity score obtained from Section 4.2.2. One can visualize the problem as separate islands of information: within an island information is bridged with references; across islands there are no connecting bridges. Therefore, similarity score between the referee sections is increased due to the similarity in the referenced sections, and this increase is proportional to the similarity score between the referenced sections.

We deploy a system similar to the  $s\text{-psc}$  and  $\text{psc-psc}$  process, replacing  $\text{psc}$  with  $\text{ref}$  which represents the set of outlinks from a section:

$$\begin{aligned} \text{Set } S &= f_{\text{psc-psc}}(\text{ref}(A), \text{ref}(B)) \\ N &= \text{sizeof}(S) \\ \delta_{GT} &= \sum_{s \in S} (s - f_{\text{psc-psc}}(A, B)), s \in S \\ \alpha_{rd} &= \text{discount factor of update} \\ \text{if } (N \neq 0) \quad f_{rd}(A, B) &= f_{\text{psc-psc}}(A, B) + \alpha_{rd} \times (\delta_{GT} / N) \\ \text{else} \quad f_{rd}(A, B) &= f_{\text{psc-psc}}(A, B) \end{aligned}$$

## 5. RESULTS

Preliminary results are obtained by taking the score from concept match as the base score, and the discount factor  $\alpha$  is taken to be 0.5 for all cases. Due to the volume of documents involved, a number of sections from different regulations are randomly selected for comparison to assess system performance.

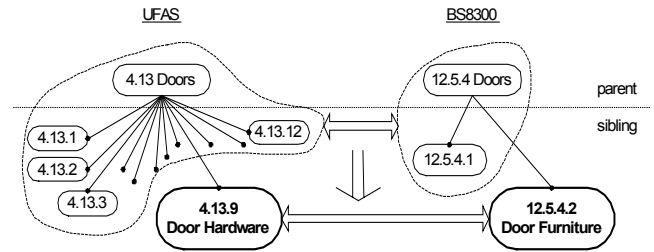


Figure 3. Score refinement based on neighboring nodes in tree

#### Example 4

<b>UFAS</b>
4.13 Doors
4.13.1 General
...
4.13.9 Door Hardware
Handles, pulls, latches, locks, and other...
...
4.13.12 Door Opening Force
<b>BS8300</b>
12.5.4 Doors
12.5.4.1 Clear Widths of Door Openings
12.5.4.2 Door Furniture
Door handles on hinged and sliding doors ...

#### Example 5

<b>UFAS</b>
4.1.2 Accessible Buildings: New Construction
(4) Stairs connecting levels that are not connected ...
<b>Scottish Technical Standards</b>
3 Stairs and ramps
3.17 Pedestrian Ramps
a raised kerb at least 100mm high on any ...

To illustrate the similarity between American and British standards, we compare UFAS with BS8300. Example 4 shows sections from the two regulations both focusing on doors. Given the relatively high similarity score between Sections 4.13.9 and 12.5.4.2 ( $f_0 = 0.425$ ), they are expected to be related, and in fact they are; Section 4.13.9 from the American code is titled “Door Hardware” while Section 12.5.4.2 from the British standard is titled “Door Furniture.” As the American and British phrasing is different, concept comparison does not pick up the match between “door hardware” and “door furniture”; however, by comparing the neighbors of the sections, we observe a higher similarity score ( $f_{\text{psc-psc}} = 0.471$ ). As shown in Figure 3, similarities in neighboring nodes in the regulation trees imply a higher similarity between the compared Sections 4.13.9 and 12.5.4.2.

Comparing  $f_{psc-psc}$  with  $f_{rd}$ , we find it difficult to observe any major improvements after neighbor inclusion. This is possibly due to the relatively high threshold in the algorithm:  $f_{rd}$  is only updated from  $f_{psc-psc}$  if the outlinks have higher similarities between them. However, some improvement still exists; for instance, in Example 5, both sections from the UFAS and the Scottish code are concerned about pedestrian ramps and stairs which are related accessible elements. Indeed, after reference distribution, these two provisions show a significant increase in the similarity score from  $f_{psc-psc}$  of 0.094 to  $f_{rd}$  of 0.31.

## 6. CONCLUSIONS AND FUTURE TASKS

This project aims to develop an infrastructure for regulation management and comparative analysis. A repository is built by transforming regulations into XML format because of its capability to handle semi-structured data. After all regulations are in a unified format, features are extracted from the corpus semi-automatically, in addition to features from reference materials such as engineering handbooks. A taxonomy is developed on top of the concepts identified by a text mining tool to allow for easy viewing following the classification. With the repository fully functional online, users can browse through regulatory documents according to the document hierarchy or based on concept clusters.

We then perform a similarity analysis. It first computes a base score between pairs of provisions by combining similarity scores from each of the features. The base score is refined by taking immediate neighboring sections into account. Reference distribution is performed to further refine the scores according to the reference structure in the regulations. Preliminary results are obtained by comparing several sets of accessibility regulations, and we have provided examples to show that our system does reveal hidden relatedness between provisions through neighbor inclusion and reference distribution.

Once the prototype is thoroughly tested and evaluated on accessibility regulations, we anticipate the incorporation of environmental regulations to demonstrate scalability and practicality of the system. In addition, due to the existence of multiple sources of regulations and thus potential conflicts between them, conflict identification becomes the natural next step to a complete regulatory document analysis. In the long run, we plan to study the formal representation derived from structured texts in order to perform automated analysis of overlaps, completeness and conflicts.

## 7. ACKNOWLEDGMENTS

This research project is sponsored by the National Science Foundation, Contract Numbers EIA-9983368 and EIA-0085998. The authors would like to acknowledge a "Technology for Education 2000" equipment grant from Intel Corporation. We would also like to acknowledge the support by Semio Corporation in providing the software for this research.

## 8. REFERENCES

- [1] *ADA Accessibility Guidelines for Buildings and Facilities*. The Access Board, 1998.

- [2] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. ACM Press, New York, NY, 1999.
- [3] Bishop, C. *Neural Networks for Pattern Recognition*. Oxford University Press; Clarendon Press, New York, NY, 1995.
- [4] Bollacker, K.D., Lawrence, S. and Giles, C.L. CiteSeer: an autonomous web agent for automatic retrieval and identification of interesting publications. in *Proceedings of the 2nd International Conference on Autonomous Agents* (Minneapolis, MN, 1998), ACM Press, 116-123.
- [5] Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. in *Proceedings of the 7th International World Wide Web Conference* (Brisbane, Australia, 1998), 107-117.
- [6] *British Standard 8300*. British Standards Institution (BSI), 2001.
- [7] *California Building Code*. California Building Standards Commission, 1998.
- [8] Dorre, J., Gerstl, P. and Seiffert, R. Text mining: finding nuggets in mountains of textual data. in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, CA, 1999), 398-401.
- [9] Gibbens, M.P. *California Disabled Accessibility Guidebook 2000*. Builder's Book, Canoga Park, CA, 2000.
- [10] Hovy, E. Using an ontology to simplify data access. *Communications of the ACM*, 46 (1). 47-49.
- [11] *International Building Code 2000*. International Conference of Building Officials, 2000.
- [12] Kidder, F. and Parker, H. *Kidder-Parker Architects' and Builders' Handbook*. John Willey & Sons, London, UK, 1931.
- [13] Mitra, P. and Wiederhold, G. Resolving terminological heterogeneity in ontologies. in *Proceedings of Workshop on Ontologies and Semantic Interoperability at the 15th European Conference on Artificial Intelligence (ECAI)* (Lyon, France, 2002).
- [14] Porter, M.F. An algorithm for suffix stripping. *Program: Automated Library and Information Systems*, 14 (3). 130-137.
- [15] Salton, G. *The smart retrieval system - experiments in automatic document processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.
- [16] Salton, G. and Buckley, C. Term-weighting approaches in automatic retrieval. *Information Processing and Management*, 24 (5). 513-523.
- [17] *Semio Tagger*. Semio Corporation, 2002. <http://www.semio.com>.
- [18] *Uniform Federal Accessibility Standards (UFAS)*. The Access Board, 1986.
- [19] *Extensible Markup Language (XML)*. World Wide Web Consortium (W3C), 2003. <http://www.w3.org/XML>.