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, like those from federal, state, and local offices, 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 based on concepts generated by knowledge engineers. Our similarity analysis core identifies relevant provisions and brings them to the user's attention, and this is performed by utilizing both the structure and referencing of regulations to provide a better comparison between provisions. Preliminary results show that our system reveals hidden similarities between provisions that are not identified using traditional comparison techniques.

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.

General Terms

Algorithms, Documentation, Theory, Legal Aspects.

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. Curious citizens are entitled to and thus should be provided with the means to better understand government regulations. In addition to the general use by the public, regulations are reviewed and used by industry designers, planners and inspectors. 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 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 compliment and modify each other, and at times contradict one another. Designers often turn for resolution to reference handbooks that are independent of governing bodies, such as the California Disabled Accessibility Guidebook (CalDAG) [12] by Gibbens. As a result, the regulations, amending provisions and interpretive manuals together create a massive volume of semi-structured documents with possible 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 [12]. The California Building Code (CBC) [9] 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 2 below 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 [12] 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.

Example 1

> ADAAG Appendix
>
> *A4.6.3 Parking Spaces*
>
> …The parking access aisle must either blend with the accessible route or have a curb ramp complying with 4.7. Such a curb ramp opening must be located within the access aisle boundaries, not within the parking space boundaries. Unfortunately, many facilities are designed with a ramp that is blocked when any vehicle parks in the accessible space. Also, the required dimensions of the access aisle cannot be restricted by planters, curbs or wheel stops.
>
> CBC
>
> *1129B.4.3 Equivalent facilitation for parking arrangements*
>
> …Pedestrian ways which are accessible to persons with disabilities shall be provided from each such parking space to related facilities, including curb cuts or ramps as needed. Ramps shall not encroach into any parking space. EXCEPTIONS: 1. Ramps located at the front of accessible parking spaces may encroach into the length of such spaces when such encroachment does not limit the capability of a person with a disability to leave or enter a vehicle, thus providing equivalent facilitation…

Example 2

> ADAAG
>
> *4.7.2 Slope*
>
> Slopes of curb ramps shall comply with 4.8.2. The slope shall be measured as shown in Fig. 11. Transitions from ramps to walks, gutters, or streets shall be flush and free of abrupt changes. Maximum slopes of adjoining gutters, road surface immediately adjacent to the curb ramp, or accessible route shall not exceed 1:20.
>
> CBC
>
> *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 edge for persons with visual impairments.

1.2 A High-Value Application Domain for KDD Tool

A knowledge discovery and data mining (KDD) tool for legal documents is valuable for the industry, particularly for small businesses. Small companies simply do not have the resources and cannot afford to hire lawyers or specialists to do compliance check for projects and developments, and thus often suffer from fines for regulation violations. The sheer volume of regulations from different governing bodies makes it difficult for small businesses to locate relevant information, which in turn hinders the growth of such companies that have to devote their already-limited resources on compliance checks or budgets for penalties. Therefore, a tool for regulatory document analysis could help small businesses to locate related provisions, and thus makes understanding of regulations easier. In addition, tools that group similar or conflicting provisions together significantly shorten the process of compliance check against the complicated set of regulations.

Other than the application on legal documents, the techniques developed for regulations can be generalized to other domains as well. Regulatory documents are semi-structured; they follow a strict hierarchy of parent and child provisions. Also, as shown in Examples 1 and 2, provisions in regulations are heavily cross-referenced. This diversion from generic documents leads to our proposal of a similarity analysis system that utilizes the document structure to achieve a better comparison than that of traditional textual comparison techniques in the field of Information Retrieval (IR). Thus the application can be extended to other semi-structured documents, e.g., traditional textbooks organized chapter by chapter, with sections and subsections within each chapter, or software user manuals that are often cross-linked as much as regulations.

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. As suggested above, conflict analysis is anticipated as well but will not be discussed in this paper; Section 6 gives a brief discussion on future tasks.

2. RELATED WORK

Guidance in the interpretation of legal documents has existed as long as legal documents themselves. Reference materials and handbooks are merely the byproducts of the many sources of regulatory agencies and the ambiguity of regulatory documents. For instance, CalDAG is a handbook written for compliance guidance with the accessibility code. It claims to "sort out and explain the differences between the ADA & Title 24 that all California professionals must understand and apply to comply with both laws" [12].

Despite the fact that interpretive guidelines have long existed, the introduction of information technology to aid legal interpretation is rather new. The recent increase in network capacity has given rise to the proposal of a web-based broker for regulations [17]. Data mining techniques, in particularly text mining algorithms, are sought to perform classification and clustering on legal documents [27]. Most of the recent research focuses on enhancing the search and browse aspects of the legal corpus, whose targeted users are legal practitioners.

To aid legal reasoning and interpretation, most knowledge bases develop upon a rule-based system or a network representation. However, rule-based systems have limited scalabilities, and in particular logic programming does not deal with the ambiguities of legal issues. Graph or network representations, on the other hand, require knowledge engineers and domain experts to create the representation structure themselves, which is often a difficult and subjective task [27]. In our development, we try to avoid assumptions involving the interpretation of regulations or the structure of the model.

2.1 Feature Extraction

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 [5]. 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" [5]. The IBM Intelligent Miner for Text [11] and the Semio Tagger [25] are both examples of fully automated key phrase extraction tools.

Apart from reducing the effect of the curse of dimensionality [4], feature extraction in text mining identifies important phrases by pulling together terms to form concepts. This captures the sequencing information of terms, and experiments have shown that phrases can convey more important information than the terms separated. For example, as pointed out in [15], “joint venture is an important term in the Wall Street Journal database, while neither joint nor venture are important by themselves. In fact, in a 800+ Mbytes database, both joint and venture would often be dropped from the list of terms by the system because their idf weights were too low”.

2.2 Similarity Analysis

As mentioned above, regulatory documents are organized into deep hierarchies and sections in regulations are heavily cross-referenced. With this in mind, a brief overview of related textual and structural analysis algorithms is given below.

Text document comparison, in particular similarity analysis between generic documents, is widely studied in Information Retrieval (IR). Techniques such as the Boolean model and the Vector model exist [3], and most of these are bag-of-word type of analysis (i.e. word order insensitive). This type of model cannot capture synonymic information without the help of thesauri; Latent Semantic Indexing (LSI) [10] fills the gap between word and concept. LSI uses an algorithm called Singular Value Decomposition (SVD) to reduce the dimension of term space into concept space; the claim is that synonyms that represent the same concept are mapped onto the same concept axis. In our project, LSI will be used as the control to compare with our experimented result.

Since a strict boolean term matching model ignores synonyms which can convey important information at times, work has been done to resolve terminological heterogeneity. As shown in [20], 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.

The heterogeneity of different data structures and their implied comparisons have been widely studied in the field of database management systems. In particularly, semantic interoperations between sources of information are enabled by a well-defined ontology mapping system [19]. However, as pointed out above, all regulations follow a strict hierarchical structure regardless of their source. In addition, the terminologies used in each regulation are well defined, which makes the use of an ontology matching system unnecessary. In the future, if more free-form texts are added to the corpus, or if the relationships between provisions become more complicated than parents and children, an ontology matching system can be handy.

In addition to comparing the body text of provisions, the heavily referenced nature of regulations provides extra information about provisions, and link analysis [7] is the natural improvement to the similarity measure. Academic citation analysis [6] 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 Americans with Disabilities Act Accessibility Guidelines (ADAAG) [1], and the Uniform Federal Accessibility Standards (UFAS) [2]. In addition, Chapter 11 of the International Building Code (IBC) [14], titled Accessibility, is included to reflect the similarity and dissimilarity between federal and private agency mandated regulations. Related sections from the British Standard BS8300 [8] and the Scottish Technical Standards [24] 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) [26] for its capability to handle semi-structured data. The hierarchy of regulations is maintained by properly structuring the XML tags, for example, Section 3.4.1 is a child node of Section 3.4, and is thus structured as a child element of Section 3.4 in the XML tree.

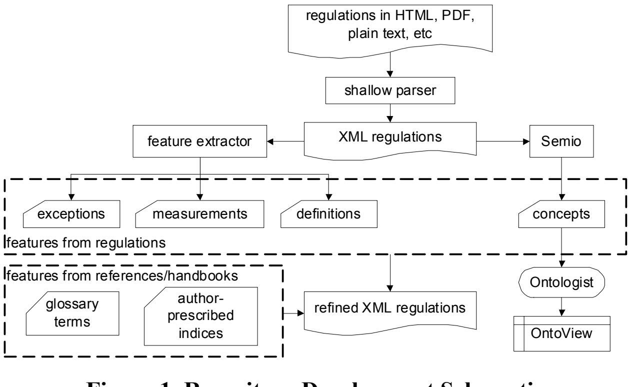
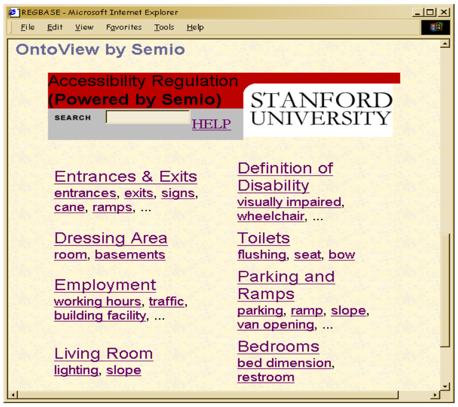


Figure 1. Repository Development Schematic

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 [13] 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. It also provides a concept latching tool to help knowledge engineer to categorize the concepts and create a taxonomy. Documents are thus clustered according to the taxonomy, and users can click through the structure to view relevant provisions classified with concepts. Figure 2 below shows a sample taxonomy generated using Semio.


REGBASE - Microsoft Internet Explorer
File Edit View Favorites Tools Help

OntoView by Semio

Accessibility Regulation (Powered by Semio)

SEARCH HELP

STANFORD UNIVERSITY

Entrances & Exits
entrances, exits, signs, cane, ramps, ...

Definition of Disability
visually impaired, wheelchair, ...

Dressing Area
room, basements

Toilets
flushing, seat, bow

Employment
working hours, traffic, building facility, ...

Parking and Ramps
parking, ramp, slope, van opening, ...

Living Room
lighting, slope

Bedrooms
bed dimension, restroom

Figure 2. OntoView by Semio

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 might 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 such as Semio Tagger 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 3.2.5.

3.2.1 Concept Tag

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-word 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 Semio identifies 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 [21] before matching. Below is an example of a concept and its count.

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

3.2.2 Author-Prescribed Indices

Semio extracts key phrases from the corpus by linguistic analysis and other techniques; these machine-generated phrases are a good measure of important concepts in the body text of 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.

To start out, index terms from Chapter 11, Accessibility, of the IBC [14] are tagged against the repository. Again the terms and the body texts are both stemmed to increase the number of matches, and the syntax is identical to a concept tag except that the element name is replaced with indexTerm. Below is an example of the indexTerm tag.

```
<indexTerm name="valet parking" num="1" />
```

3.2.3 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 Semio concepts. In addition, the definition of a term gives the meaning to a term, which is useful for comparisons.

```
<definition>
<term> Accessible </term>
<definedAs> Describes a site, building, facility, or portion thereof
that complies with these guidelines. </definedAs>
</definition>
```

Similarly, engineering handbooks always define the important terms used in the field in the glossary. 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" [16]. The difference between definition and glossaryDef is that definition comes from the regulation itself, while glossaryDef comes from sources other than the regulation.

```
<glossaryDef>
<term> Return Head </term>
<definedAs> The continuation of a molding, projection, etc., in an
opposite direction. </definedAs>
</glossaryDef>
```

3.2.4 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”.

We first identify 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" />
```

In addition, range (e.g., 2 to 3 inches) and area (e.g., between 2 and 3 lbf) measurements are identified, and an area measurement tag is shown as follows:

```
<measurement unit="lbf" size1="2" size2="3" quantifier="min" />
```

3.2.5 Examples with Complete Mark-up

Presented below are two examples with the complete set of feature mark-ups. The first example comes from the ADAAG definition section, and it shows the section hierarchy in addition to the extracted definition, concept and indexTerm tags. The second example is a typical provision from the UFAS, which contains exception, measurement and ref tags in addition to the body text regText tag. All of the extracted information are capsulated in a regElement node for each section. The selected provisions tend to be rather lengthy to illustrate most of the mark-ups in a single provision, and therefore only excerpts of the body text and the mark-ups are shown below.

Example 3

```
Original Section 3.5 from ADAAG
3 Miscellaneous instructions and definitions
  …
  3.5 Definitions
    …
    ACCESSIBLE.
    Describes a site, building, facility, or portion thereof that
    complies with these guidelines.
    …
    CLEAR.
    Unobstructed.
    …
Refined Section 3.5 in XML format
<regElement name="adaag.3" title="miscellaneous instructions
and definitions">
  …
  <regElement name="adaag.3.5" title="definitions">
    <concept name="accessible means" num="2" />
    <indexTerm name="facility" num="1" />
    <definition>
      <term> accessible </term>
      <definedAs> Describes a site, building, facility, or portion
      thereof that complies with these guidelines.</definedAs>
    </definition>
    <definition>
      <term> clear </term>
      <definedAs> Unobstructed. </definedAs>
    </definition>
    …
  </regElement>
</regElement>
```

Example 4

```
Original Section 4.6.3 from the UFAS
4.6.3 Parking Spaces
  Parking spaces for disabled people shall be at least 96 in
  (2440 mm) wide and shall have an adjacent access aisle 60
  in (1525 mm) wide minimum (see Fig. 9). Parking access
  aisles shall ...
  EXCEPTION: If accessible parking spaces for vans designed
  for handicapped persons are provided, each should have ...
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.

A schematic is shown below in Figure 3 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. The process starts with an initial similarity score obtained by feature matching. Then the immediate surrounding nodes are compared as well to modify their initial score. The influence of the not-so-immediate surroundings of nodes A and B is taken into account by a process called Reference Distribution. The entire process together produces a reliable set of scores, and below threshold pairs of provisions are discarded as dissimilar pairs. Details of each process follow in Sections 4.1 through 4.2.

As Section 4.3 shows, a control experiment is implemented using LSI techniques to assess system performance. Each provision is represented by a vector of words, or concepts if SVD is performed; pairwise comparison of sections can be obtained from the cosine similarity of vectors, or other similarity measures as discussed in Section 8.5 in [18]. Note that the ranking, not the actual similarity score, will be compared with that of our system. We anticipate that, through the utilization of the structure of regulations and the addition of domain-specific knowledge from feature extraction, our system will perform better than a bag-of-word type of comparison such as LSI; or at the very least, provide additional useful information in comparison and ranking.

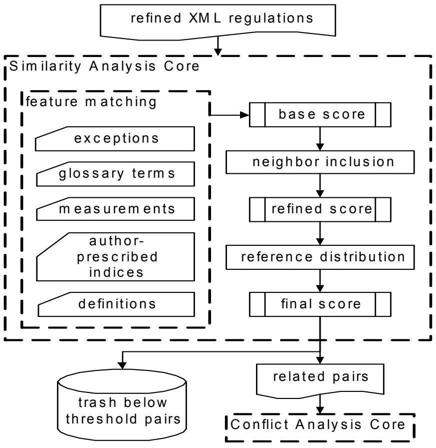


Figure 3. The Similarity Analysis Core Schematic

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 will take concept matching as an example to illustrate the basic idea.

$$f_0 = (\Sigma_{i = \text{features}} f_i) \,/\, \#\text{ features } i \qquad (1)$$

Concepts are used exactly like the index terms in the vector model [22], 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 [23] 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 *psc* 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 *psc*(B), and vice versa, to produce the score $f_{s\text{-}psc}$ based on the initial score f_0(A, B). The next refinement takes into account the comparison between *psc*(A) and *psc*(B), which gives the score $f_{psc\text{-}psc}$. 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 $f_{s\text{-}psc}$ 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. *psc*(B), and vice versa, to obtain $f_{s\text{-}psc}$(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

Set $S = f_0(\text{A, psc(B)}) \cup f_0(\text{psc(A), B})$
$N = \text{sizeof}(S)$
$\delta_{GT} = \Sigma_{s>f_0(\text{A, B})}\,(s - f_0(\text{A, B})),\ s \in S$
$\alpha_{s\text{-}psc}$ = discount factor of update
if (N != 0) $f_{s\text{-}psc}(\text{A, B}) = f_0(\text{A, B}) + \alpha_{s\text{-}psc} \times (\delta_{GT} / N)$
else $f_{s\text{-}psc}(\text{A, B}) = f_0(\text{A, B})$

Here, set S is the set of similarity scores between section A and *psc*(B), and between *psc*(A) and section B. The total δ_{GT} sums over all s in set S which is greater than the original score; thus δ_{GT} / N represents the average greater-than score. Clearly α is always less than one, following our intuition that self-self comparison is more important than self-*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 *psc-psc* on sections A and B. Here *psc*(A) is compared against *psc*(B) to refine f_0(A, B), which implies that another layer of indirection is inferred and thus the weight of *psc-psc* should be less than that of *s-psc*. We have

Set $S = f_{s\text{-}psc}(\text{psc(A), psc(B)})$
$N = \text{sizeof}(S)$
$\delta_{GT} = \Sigma_{s>f_{s\text{-}psc}(\text{A, B})}\,(s - f_{s\text{-}psc}(\text{A, B})),\ s \in S$
$\alpha_{psc\text{-}psc}$ = discount factor of update
if (N != 0) $f_{psc\text{-}psc}(\text{A, B}) = f_{s\text{-}psc}(\text{A, B}) + \alpha_{psc\text{-}psc} \times (\delta_{GT} / N)$
else $f_{psc\text{-}psc}(\text{A, B}) = f_{s\text{-}psc}(\text{A, B})$

By separating the process of comparing *s-psc* and *psc-psc*, we are enforcing the intuition that the comparison between self (e.g., section A) and *psc* (e.g., *psc*(B)) should weigh more than that of *psc* (e.g., *psc*(A)) and *psc* (e.g., *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 other documents as shown in Figure 4.

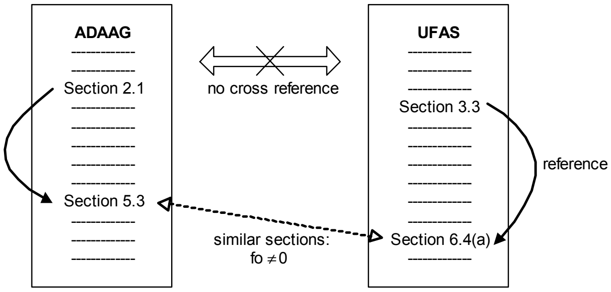


Figure 4. Comparison of Section 2.1 from ADAAG with Section 3.3 from UFAS

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. The above figure illustrates the idea; it is important to note that we are utilizing the self-reference structure but not the cross-references, which implies that neither the referees nor the referrers are the same for the two sections in interest. 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. From Figure 4, it is appropriate to claim that the similarity score between Section 2.1 in ADAAG and Section 3.3 in UFAS should be increased due to the similarity in the referenced sections. Indeed, this increase should be proportional to the similarity score between the referenced sections.

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

```
Set $S = f_{psc\text{-}psc}(\text{ref (A), ref (B)})$
$N = \text{sizeof}(S)$
$\delta_{GT} = \Sigma_{s > fpsc\text{-}psc(A,\ B)}\ (s - f_{psc\text{-}psc}(A, B)),\ s \in S$
$\alpha_{rd}$ = discount factor of update
if $(N\ != 0)\ f_{rd}(A, B) = f_{psc\text{-}psc}(A, B) + \alpha_{rd} \times (\delta_{GT} / N)$
else        $f_{rd}(A, B) = f_{psc\text{-}psc}(A, B)$
```

4.3 LSI: the control

A traditional LSI approach [10] is used as the control. A term-document matrix [A] is populated with the *tf*×*idf* measure of the index term in the document, while documents here represent the entire corpus of sections from both regulations. We have

$$a_{ij} = tf_{ij} \times \log\ (\ n\ /\ n_i\ ) \qquad (2)$$

where tf_{ij} is the term frequency of term i in document (section in our framework) j, and the log term is the inverse document frequency (*idf*) with n being the total number of documents, and n_i being the number of documents with term i. In addition, each document vector is normalized to length one. SVD is then performed on the [A] matrix to map index terms to concept space, and also to reduce noise. We have

$$[\text{A}] = [\text{P}][\text{Q}][\text{R}]^{\text{T}} \qquad (3)$$

The diagonal [Q] matrix is then partly zeroed out for dimension reduction. For some $s << r = \text{rank}[\text{Q}]$, we take only the largest s singular values from [Q] and zero out the rest to form $[\text{Q}_\text{s}]$. We then have

$$[\text{A}_\text{s}] = [\text{P}_\text{s}][\text{Q}_\text{s}][\text{R}_\text{s}]^{\text{T}} \qquad (4)$$

with $[\text{P}_\text{s}]$ and $[\text{R}_\text{s}]$ being the corresponding stripped out version of the original matrix as part of Q is zeroed out. The document-to-document (doc-doc) similarity matrix is given by

$$[\text{A}_\text{s}]^{\text{T}}[\text{A}_\text{s}] = [\text{R}_\text{s}][\text{Q}_\text{s}]^2[\text{R}_\text{s}]^{\text{T}} \qquad (5)$$

Indeed, since we are solely interested in comparing different documents but not self-comparisons, we only need the upper right hand quadrant of the doc-doc similarity matrix.

5. RESULTS

Preliminary results are obtained by taking the score from concept match as the base score, and the discount factor α is taken to be 0.5 for all cases. Sections from different regulations are randomly selected for comparison to assess system performance.

First, to justify for neighbor inclusions within our system, we compare results from f_0 and $f_{s\text{-}psc}$ and some improvement is observed. For instance, Example 5 below shows that Section 4.1.6(3)(d) in ADAAG is concerned with doors, while Section 4.14.1 in UFAS deals with entrances. As expected, the concept match in f_0 could not identify the similarity between door and entrance, thus $f_0 = 0$. With $f_{s\text{-}psc}$, the system is able to infer some relatedness between the two sections from the neighbors in the tree, and thus results in a nonzero score for $f_{s\text{-}psc}$.

To illustrate the similarity between American and British standards, we compare UFAS with BS8300. Example 6 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 ($f_{psc\text{-}psc} = 0.471$). As shown in Figure 5, similarities in neighboring nodes in the regulation trees imply a higher similarity between the compared Sections 4.13.9 and 12.5.4.2.

Example 5

ADAAG

4.1.6(3)(d) Doors

(i) Where it is technically infeasible to comply with clear opening width requirements of 4.13.5, a projection of 5/8 in maximum will be permitted for the latch side stop.

(ii) If existing thresholds are 3/4 in high or less, and have (or are modified to have) a beveled edge on each side, they may remain.

UFAS

4.14 Entrances

4.14.1 Minimum Number

Entrances required to be accessible by 4.1 shall be part of an accessible route and shall comply with 4.3. Such entrances shall be connected by an accessible route to public transportation stops, to accessible parking and passenger loading zones, and to public streets or sidewalks if available (see 4.3.2(1)). They shall also be connected by an accessible route to all accessible spaces or elements within the building or facility.

Example 6

UFAS

4.13 Doors

4.13.1 General

…

4.13.9 Door Hardware

Handles, pulls, latches, locks, and other operating devices on accessible doors shall have a shape that is easy to grasp with one hand and does not require tight grasping, tight pinching, or twisting of the wrist to operate. Lever-operated mechanisms, push-type mechanisms, and U-shaped handles are acceptable designs. When sliding doors are fully open, operating hardware shall be exposed and usable from both sides. In dwelling units, only doors at accessible entrances to the unit itself shall comply with the requirements of this paragraph. Doors to hazardous areas shall have hardware complying with 4.29.3. Mount no hardware required for accessible door passage higher than 48 in (1220 mm) above finished floor.

…

4.13.12 Door Opening Force

BS8300

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 in accessible bedrooms should be easy to grip and operate by a wheelchair user or ambulant disabled person (see 6.5). Handles fixed to hinged and sliding doors of furniture and fittings in bedrooms should be easy to grip and manipulate. They should conform to the recommendations in 6.5 for dimensions and location, and the minimum force required to manipulate them.

Consideration should be given to the use of electronic card-activated locks and electrically powered openers for bedroom entrance doors.

COMMENTARY ON 12.5.4.2. Disabled people with a weak hand grip or poor co-ordination, find that using a card to open a door lock is easier than turning a key.

A wide angle viewer should be provided in doors to accessible bedrooms at two heights, 1050 mm and 1500 mm above floor level to allow viewing by a person from a seated position and a person standing.

Door furniture should contrast in colour and luminance with the door.

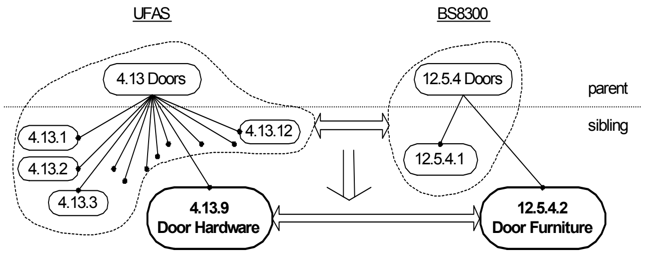


Figure 5. Score refinement based on neighboring nodes in tree

Comparing $f_{psc\text{-}psc}$ 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 $f_{psc\text{-}psc}$ if the outlinks have higher similarities between them. However, some improvement still exists; for instance, in Example 7 below, 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_{pscpsc} of 0.094 to f_{rd} of 0.31.

Example 7

UFAS

4.1.2 Accessible Buildings: New Construction

(4) Stairs connecting levels that are not connected by an elevator shall comply with 4.9.

Scottish Technical Standards

3 Stairs and ramps

3.17 Pedestrian Ramps

A ramp must have (a) a width at least the minimum required for the equivalent type of stair in S3.4; and (b) a raised kerb at least 100mm high on any exposed side of a flight or landing, except – a ramp serving a single dwelling.

Performance comparison between our system and LSI is done through a user survey. Since it is impossible for our survey subjects to read the entire corpus of regulations, ten sections from the ADAAG and the UFAS are randomly chosen as our point of comparison. To facilitate understanding, contexts are given to our subjects for sections that are deep in the tree, for example, upon reading Section 12.5.4.2 from BS8300 in Example 6, titles of its parent and relevant grandparents are shown as well. We asked users to assign a ranking between each pair of provisions from ADAAG and UFAS based on their relatedness, and the average ranking obtained from users is regarded as the *true ranking*.

To obtain the difference between the true ranking and the machine predicted ones, we rank scores from our system and those obtained from LSI and compute the least square errors between the ranking vectors. Based on concept match as the initial score and a discount factor of 0.5 between sequential refinements, the least square error for our system is roughly 21%, while the error from results obtained using LSI is 23%. This reduction in error proves that our system, with only one feature implemented and a random selection of $\alpha = 0.5$, outperforms the traditional bag-of-word model, LSI. We believe that with the addition of domain-specific knowledge as features as suggested in Section 3.2, and a fine-tuned α value, our system will be able to imply more hidden similarities between provisions.

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, or evidences, 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 an text mining tool, such as Semio, 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. A list of the most related sections is produced as a result.

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. In addition, a user survey is used to compare our system performance with that obtained using LSI, and a relative reduction in error is observed based on the use of concept match as initial score and a discount factor of 0.5 between score refinements.

Once the prototype is thoroughly tested on accessibility regulations, we anticipate the incorporation of environmental regulations in the near future 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. Assuming that the contents of the conflicting sections are related or similar, 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. In this paper, we therefore combined different techniques to further utilize the characteristics of legal documents to improve similarity analysis result. 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] The Access Board. *ADA Accessibility Guidelines for Buildings and Facilities*, 1998.

[2] The Access Board. *Uniform Federal Accessibility Standards (UFAS)*, 1986.

[3] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. ACM Press, New York, NY, 1999.

[4] Bellman, R.E. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

[5] Bishop, C. *Neural Networks for Pattern Recognition*. Oxford University Press; Clarendon Press, New York, NY, 1995.

[6] 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.

[7] 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.

[8] British Standards Institution (BSI). *British Standard 8300*, 2001.

[9] California Building Standards Commission. *California Building Code*, 1998.

[10] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41* (6). 391-407.

[11] 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.

[12] Gibbens, M.P. *California Disabled Accessibility Guidebook 2000*. Builder's Book, Canoga Park, CA, 2000.

[13] Hovy, E. Using an ontology to simplify data access. *Communications of the ACM*, *46* (1). 47-49.

[14] International Conference of Building Officials. *International Building Code 2000*, 2000.

[15] Jones, K.S. and Willett, P. *Readings in Information Retrieval*. Morgan Kaufmann, San Francisco, CA, 1997.

[16] Kidder, F. and Parker, H. *Kidder-Parker Architects' and Builders' Handbook*. John Willey & Sons, London, UK, 1931.

[17] Liang, V.-C. and Garrett, J.H. Java-based environmental regulations broker. *Journal of Computing in Civil Engineering*, *14* (2). 100-108.

[18] Manning, C.D. and Schutze, H. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

[19] Mitra, P. and Wiederhold, G. An algebra for semantic interoperability of information sources. in *Proceedings of the 2nd IEEE Symposium on BioInformatics and Bioengineering* (Bethesda, MD, 2001), 174-182.

[20] 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).

[21] Porter, M.F. An algorithm for suffix stripping. *Program: Automated Library and Information Systems*, *14* (3). 130-137.

[22] Salton, G. *The smart retrieval system - experiments in automatic document processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[23] Salton, G. and Buckley, C. Term-weighting approaches in automatic retrieval. *Information Processing and Management*, *24* (5). 513-523.

[24] Scottish Executive. *Technical Standards*, 2001.

[25] Semio Corporation. *Semio Tagger*, 2002. http://www.semio.com.

[26] World Wide Web Consortium (W3C). *Extensible Markup Language (XML)*, 2003. http://www.w3.org/XML.

[27] Zeleznikow, J. and Hunter, D. *Building Intelligent Legal Information Systems*. Kluwer Law and Taxation Publishers, Deventer, the Netherlands, 1994.