Tree Edit Distance in a Document Comparison

Martin Milička

Brno University of Technology

LANGUAGE THEORY with APPLICATIONS 2011

Content

Motivation

In some cases, the textual based comparison is not good enough for a document comparison because there is missing a visual influence. It brings a human perception. In HTML, we are talking about structure based similarity.

- Document comparison
  - textual approach (text)
  - visual approach (structure, colour, sizes, etc.)
- Tree
  - is a well studied combinatorial structure in computer science
  - is a finite connected acyclic graph with distinguish root node
- Tree comparison
  - occurs in several areas (biology, structured text databases, image analysis, compiler optimalization)

Tree Edit Distance (TED)

Definition

The algorithm searches the sequence of edit operations turning tree T_1 into tree T_2. Tree edit distance is a sequence with the minimum cost. Evaluates the structural differences between DOM trees.

Cost function: defines the cost of every edit operation

Edit operations: insertion, deletion and relabeling

Specific tree notation:

- *Order x Unorder tree (connection to a time complexity)*
- *Labeled x Unlabeled tree*

Basic Operations

The operations are defined on pairs of nodes.

Relabeling

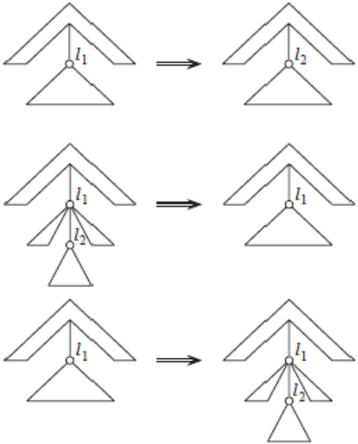
- changes the label of the node label l_1 to l_2

Deleting

- non-root node l_2 with parent l_1.
- making the children of l_2 to become the children of l_1

Inserting

- the complement of delete



Document Model

- Elements of web document are defined in DOM
- DOM has a tree structure
- DOM is an *ordered* tree
- DOM is a *labeled* tree - each node has a name

Problem: DOM trees are too complex for a tree structure comparison

Solution: abstraction + compression

Translation

Visual (class) tag	HTML tags
grp	table, ul, html, body, tbody, div, p
row	tr, li, h1, h2, hr
col	td
text	otherwise

$$\Sigma_{\mathbb{V}} = \{grp, row, col, text\}$$

$$trn :: \tau(\mathcal{T}ext \cup \mathcal{T}ag) \to \tau(\Sigma_{\mathbb{V}})$$

$$trn(f(t_1, \ldots, t_n)) = \begin{cases} \alpha(f) & n = 0 \\ \alpha(f)(trn(t_1), \ldots, trn(t_n)) & otherwise \end{cases}$$

where $\alpha :: (\mathcal{T}ext \cup \mathcal{T}ag) \to \Sigma_{\mathbb{V}}$

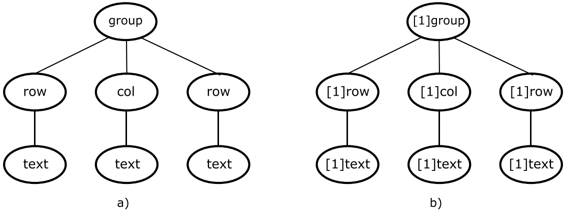
$\tau\,(\Sigma_{\mathbb{V}})$ term of algebra $\Sigma_{\mathbb{V}}$

$page \in \tau(\mathcal{T}ext \cup \mathcal{T}ag)$

Document Compression

$\tau([\mathbb{N}]\Sigma_{\mathbb{V}})$ is a marked term where $\mathbb{N}$ is a number of occurrence

For example: *[2]row([1]text)*



Compression types:

- horizontal
- vertical

Horizontal Compression

Let $t = [r_1]f(t_1, ..., t_n)$, $s = [r_2]f(v_1, ..., v_n) \in \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$ where $t \equiv_{\Sigma_{\mathbb{V}}} s$

$$join :: \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \times \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \rightarrow \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$$

$$join(t, s) = \widehat{join}(t, s, 1, 1, 1)$$

The auxiliary function $\widehat{join}$ is defined as:

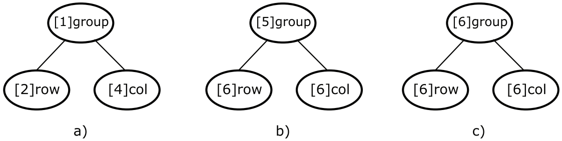
$$\widehat{join} :: \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \times \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$$

$$\widehat{join}(t, s, k_1, k_2, p) = \begin{cases} [m]f & n = 0 \\ [m]\, f(\widehat{join}(t_1, v_1, r_1, r_2, m), ..., \\ \qquad \widehat{join}(t_n, v_n, r_1, r_2, m)) & n > 0 \end{cases}$$

where $m = \lceil (r_1 * k_1 + r_2 * k_2)/p \rceil$

Horizontal Compression

Example:



The number of *rows* is computed as $m = \lceil (1 * 2 + 5 * 6)/6 \rceil$.

Horizontal compression

$$hrz(t) = \begin{cases} t & n = 0 \\ hrz(f(t_1, \ldots, t_{i-1}, s, t_{j+1}, \ldots, t_n)) & ((1 \leq i \leq j \leq n) \text{ and} \\ \qquad \text{where } s = join(t_i, \ldots, t_j) & \quad (t_i \equiv_{\Sigma_{\mathbb{V}}} t_{i+1} \ldots t_{j-1} \equiv_{\Sigma_{\mathbb{V}}} t_j)) \\ f(hrz(t_1), \ldots, hrz(t_n)) & \text{otherwise} \end{cases}$$

Vertical Compression

The safe vertical conditions (SVC):

$r = 1$	(number of repetition)
$n = 1$	(number of children)
$\neg(f \equiv group \wedge root(t_1) \not\equiv group)$	(preserve the page structure)
$root(t_1) \not\equiv text$	(preserve the information in page)

Let $t = [r]f([m]g(t_1, \ldots, t_n)) \in \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$ and if the rules of Save vertical compression are fulfilled then the *shrinking* of t is defined as:

$$shr :: \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \rightarrow \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$$

$$shr([r]f([m]g(t_1, \ldots, t_n))) = \begin{cases} [r]f(t_1, \ldots, t_n) & m = 1 \wedge g \not\equiv group \\ [m]\,g(t_1, \ldots, t_n) & otherwise \end{cases}$$

Vertical Compression

Vertical compression

$$vrt :: \tau([\mathbb{N}]\Sigma_{\mathbb{V}}) \rightarrow \tau([\mathbb{N}]\Sigma_{\mathbb{V}})$$

$$vrt(t) = \begin{cases} t & n{=}0 \\ vrt(shr(t)) & t \text{ obeys SVC} \\ [r]\, f(vrt(t_1), \ldots, vrt(t_n)) & \text{otherwise} \end{cases}$$

Tree Edit Distance in a Document Comparison

Let $nd_1, nd_2 \in [\mathbb{N}]\,\Sigma_{\mathbb{V}}$ be two marked trees. Then λ denotes a fresh symbol that represents the empty marked term, i.e., $[0]t$ for any t.

Each edit operation is presented as:

$$(nd_1 \to nd_2) \in ([\mathbb{N}]\,\Sigma_{\mathbb{V}} \times [\mathbb{N}]\,\Sigma_{\mathbb{V}}) \backslash (\lambda, \lambda)$$

where $(nd_1 \to nd_2)$ is relabeling if $nd_1 \not\equiv \lambda$ and $nd_2 \not\equiv \lambda$
is a deletion if $nd_2 \equiv \lambda$
is an insertion if $nd_1 \equiv \lambda$

Metric cost function:

$$\gamma :: ([\mathbb{N}]\,\Sigma_{\mathbb{V}} \times [\mathbb{N}]\,\Sigma_{\mathbb{V}}) \backslash (\lambda, \lambda) \to \mathbb{R}$$

$$\gamma(nd_1 \to nd_2) = \begin{cases} 0 & nd_1 \equiv_{\Sigma_{\mathbb{V}}} nd_2 & \\ r_2 & nd_1 \equiv_{\Sigma_{\mathbb{V}}} \lambda & (insertion) \\ r_1 & nd_2 \equiv_{\Sigma_{\mathbb{V}}} \lambda & (deletion) \\ max(r_1, r_2) & otherwise & (relabeling) \end{cases}$$

Tree Edit Distance in a Document Comparison

The cost of a sequence $S = s_1, ..., s_n$ of edit operations is given by

$$\gamma(S) = \sum_{i=1}^{n} \gamma(s_i)$$

The *edit distance* $\delta(t_1, t_2)$ between two trees t_1 and t_2 is defined:

$$\delta(t_1, t_2) = min\,\{\gamma(S)\}$$

Web pages comparison

$$cmp :: \tau\left([\mathbb{N}]\Sigma_{\mathbb{V}}\right) \times \tau\left([\mathbb{N}\Sigma_{\mathbb{V}}\right) \rightarrow [0..1]$$

$$cmp(t, s) = 1 - \frac{\delta(t_{zip}, s_{zip})}{|t_{zip}| + |s_{zip}|}$$

where $t, s \in \ \tau\left([\mathbb{N}]\Sigma_{\mathbb{V}}\right)$ are two pages,

t_{zip}, s_{zip} are irreducible visual represenatives of t and s

References

M. Alpuente, D. Romero.
A Visual Technique for Web Pages Comparison.
Theoretical Computer Science , 235:3–18, 2009.

P. Bille.
A survey on tree edit distance and related problems.
Theoretical Computer Science, 337(1-3):217–239, 2005.

G. Valiente.
An Efficient Bottom-Up Distance between Trees.
8th International Symposium of String Processing and Information Retrieval, 212–219, 2001

Thank you for your attention.

Questions?