Tree Edit Distance in a Document Comparison

Martin Milička, *imilicka@fit.vutbr.cz*

October 12, 2011

Abstract

World Wide Web is a huge repository of information. It is still growing. Some information can be redundant in this repository.The similarity of web pages can be focused on visual features or text.

The document similarity based on the text content is concerned to textual information. The visual structure is not considered. It is case of licences, standards, etc. The similarity is measured on the text matching of documents.

On the other hand, the document comparison based on visual features uses document structures, colours, sizes, etc. This comparison is used for the comparison based on a human perception. Web pages are mixtures of semantic content, page structure, and layout. However, we will be concerned on the page structure comparison. The source code of a HTML document has a tree structure. In our work, we use Tree Edit Distance algorithm.

Tree Edit Distance is an algorithm that computes the minimal editing costs of transforming one document's structure into the other. In fact, the cost is a sequence of editing operations. The basic editing operation on a tree T is a *deletion*, an *insertion* and a *substitution*. Each editing operation has a defined cost. There can be defined a specific operations that can improve the original tree edit distance algorithm. For instance, the algorithm can work with a subtree move which allows moving a subtree under a new node in one step.

This algorithm can be defined for the ordered or unordered tree and this has a connection to the time complexity.

Due to a time complexity, the algorithm has many modifications. However, each modification is concerned to a specific problem and contains some constraints.

If the constraint is a number of leaves we can reduce it by a node compression. This can be used in a web page context where we can define the compression rules on the document object model (DOM). The important thing is that the compression has no influence on the document visual structure.

In our work, we present two different approaches of the tree comparison. The first one is a bottom-up mapping and the other one is an up-down mapping.

We can meet the solution where trees are compared like strings also. Each tree is represented as pre-order sequence of nodes (labels). Afterwards, sequences are compared like strings.