**Using Association Analysis to Optimize the HTML Editor**

# This document provides information about the Association Analysis and how this can be applied to the html editor.

## What is Association Analysis:

According to the authors Pang-Ning et al. In the book Introduction to Data Mining, association analysis can be used to revealed strong relationships within large amount of data of transaction sets. These relationships are called association rules. To measure how strong an Association rule is, we can use support and confidence. The author defines support as what “determines often a rule is applicable to a given dataset” and confidence as what “determines how frequently items in Y appear in transactions that contain X”. The formal definitions are:

Support, s(X🡪Y) = σ(X∪Y)/N

Confidence, c(X🡪Y) = σ(X∪Y)/σ(X)

An association rule mining problem is when a transaction set contains rules that are bellow a threshold held for both support and confidence.

The Apriori algorithm is used to reduce the computation of choosing the most frequency item sets (item sets that meet the support threshold). The Apriori principal states that if an item set is frequent, then all of its subsets must also be frequent.

## How can this be applied to the html editor:

1. Group of element:
   1. We can look at the hierarchy of each element and use the Apriori algorithm. We start by first looking at each group of element and calculate the support for each element of the group and eliminate according to a threshold. Finally, find association rules by calculating the confidence of each frequent item set.
   2. This could be done by using a recursive function that creates a path of the hierarchy. The hierarchy can be computed only when it is need it to reduce the number of calculations the editor will do.
      1. Ex. Have minsupport & minconf = 0.75 and we have elements:



* + 1. Look at each element hierachy
       1. h1/: support = 1/6
       2. p/: support = 2/6
       3. div/ul/li, li, li/: support = 2/6
       4. p/span/: support = 1/6
       5. dif/img, img, img/: support = 1/6

We can see that 3 is used most often as a body’s children.

* + 1. Look inside each body’s children tag:
       1. div:
          1. ul/li, li, li/: support =1/2
          2. img, img, img/: support=1/2
       2. p:
          1. span/: support =1/1

Here, we can see the most frequent items inside each child tag of the body.

* + 1. Then again inside each children/children tag:
       1. div - ul:
          1. li, li, li: support=3/3
       2. div - img: support=3/3

Here, we can see that the tag li is most frequent inside the tag ul and the tag img is most frequent inside the tag div.

1. Attributes:
   1. We can first look at each attribute type and value individually and also apply the Apriori algorithm on each.
   2. The problem with attributes is that we have attribute type and attribute value, we will have to decide if we want to include the attribute type in our calculations.
2. Element attribute combination:
   1. Same idea, first look at each element with all of its attribute individually and apply the Apriori algorithm.

Reference:

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. Introduction to Data Mining,

(First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.