Appendix I

1. Correlation of values used in metrics

1.1.Pearsons Correlation

This value (Σxy) represents the frequency of the term initially chosen by the user. The value for (nΣy2) represents the set of metadata at the knowledge level generated by the recommendation system, the frequency of each metadata is used for the calculation.

$$R = \frac{\Sigma xy}{n\Sigma y^2}$$

To implement this expression, the Java programming language was used, but I will give a simple example of how it works, this expression can be validated if compared to the original Pearson correlation calculation, being adapted for this purpose, a technique performed by other researchers cited in the metrics section of the article.

Let's assume that the algorithm selects four terms that represent the metadata of the recommendation system, categorizing them according to their frequency, these values are used as a reference in the xValues, yValues matrix.

For example, we have the terms selected by the algorithm with their frequencies that use the search term as the basis of your research:

double ft1 <- 1st term, frequency 40.

double ft2 <- 2nd term, frequency 30.

double ft3 <- 3rd term, frequency 25.

double ft4 <- 4th term, frequency 10.

The search term is also analyzed by the algorithm and if this term presents a relevant frequency, it may or may not be in the categorized vector, however, sometimes the low frequency may be considered that there is not enough aggregated collective knowledge at the time of analysis, it is a dynamic system and these values can change over time.

Analyzing the search term it is found, supposedly for the example, that it has the following classification

double ft <- Search term, frequency of 30.

Now we can use these values for the correlation calculation:

double[] xValues = (ft,ft,ft,ft)

double[] yValues = (ft1,ft2,ft3,ft4)

The xValues array represents the frequency of the search term that will be compared with each term suggested by the algorithm in yValues. In order to calculate this correlation, we will call a method and assign the calculated value to a variable R:

double R = (calculateR(xValues, yValues))

```
public double calculateR(double[] xValues, double[] yValues) {
        int n = xValues.length;
        double sumX = 0;
        double sumY = 0;
        double sumXY = 0;
        double sumX2 = 0;
        double sumY2 = 0;

        for (int i = 0; i < n; i++) {
           sumX += xValues[i];
           sumY += yValues[i];
           sumXY += xValues[i] * yValues[i];
           sumX2 += Math.pow(xValues[i], 2);
           sumY2 += Math.pow(yValues[i], 2);
        }
        double numerator = sumXY;
        double denominator = n * sumY2;
        return numerator / denominator;
    }
```