Thesis Draft

Rubei Riccardo

April 16, 2018

Contents

1 Introduction

1.1 Overview

A general introduction of Crossminer, my work and how it contributes.

2 The Similarity Problem

2.1 Overview

How to solve the problem of similarity between documents, subsection about SVD, LSI and other techniques.

3 MUDABlue

3.1 Overview

The first procedure analysed was MUDABlue, unfortunately none implentation was available on the web, so i reimplemented it from scratch. The MUDABlue method is an automatc categorizaton method or a large collecton of software systems. MUDABlue method does not only categorize sooware systemsd but also determines categories rom the sooware systems collecton automatcally. MUDABlue has three major aspects: 1) it relies on no other information than the source code, 2) it determines category sets automatically, and 3) it allows a software system to be a member of multiple categories. Since we were interested only in the evaluation of the similarity we discarded the phases related to clusterization and categorization.

3.2 The Approach

The MUDABlue approach can be briefly summarized in 7 steps, as the following image depicts:

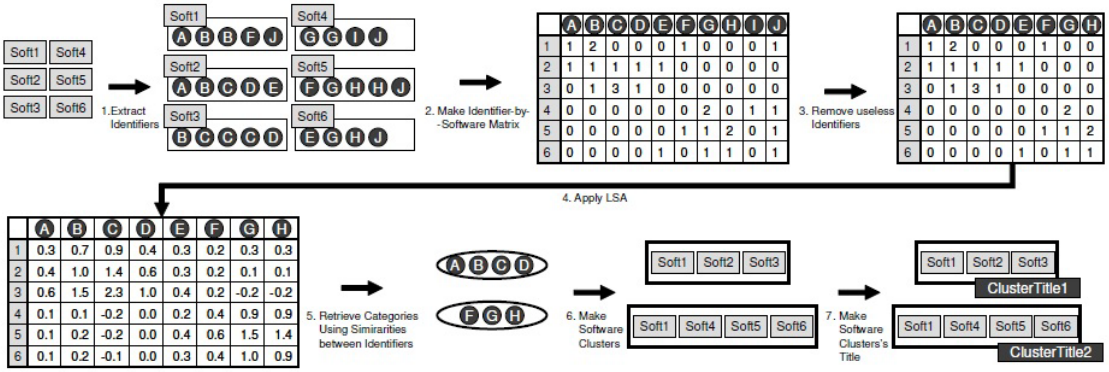

Figure 1: MUDABlue phases.

3.2.1 Exctract Identifiers

With identifier we are talking about relevant strings that can allow to characterize a document. In this phase each repository is scanned in order to find the target files, and for each of them the identifiers are exctracted, avoiding adding useless items such as comments. The dataset was a 41C projects gathered from SourceForge.

3.2.2 Create identifier-by-software matrix

As stated before, the main item to work with is the term-document matrix, in this case we count how many times each term appears in each file for all the projects. The result is matrix **m x n** with m terms and n projects.

3.2.3 Remove useless identifiers

From the matrix we remove all the useless terms, that is all the terms that apperas in just one repository, considered a specific terms, and all the terms that appears in more than 50% of the repositories, considered as general terms.

3.2.4 Apply the LSA

Once the matrix is ready con be worked, the SVD procedure is applied and then the LSI. As explained before [NOTE] the SVD procedure decompose the original matrix in 3 other matrices. When we multiply back these matrices we use a rank reducted version of the S matrix in order to generete the final one. The authors didn't provide us any details about their final rank value, so we tested many values and eventually selected one.

3.2.5 Apply the Cosine Similarity

By using the cosine similarity method, we compare each repository vector with all the others and eventually getting an **n x n** matrix, in which is expressed the similarity of all the repository couple, with a value [0.0-1.0].

3.2.6 Categorization

The point 6 and 7 are not covered because not related to our work.

4 Clan

4.1 Overview

A description of Clan approach.

5 Results

6 Conclusion