*A Mini-project Report*

# Using Semantic Similarity-based Clustering to Facilitate Web Service Discovery

*carried out as part of the course* ***Web Services (IT466)***

*Submitted By,*

**Apoorva Chandra S** (11IT09)

**Dilip Mallya** (11IT19)

**JNVK Chaitanya** (11IT32)

**R Kaushik** (11IT65)

*in partial fulfillment for the award of the degree*

*of*

**Bachelor of Technology**

*In*

**Information Technology**

*At*


**Department of Information Technology**

**National Institute of Technology Karnataka, Surathkal**

***November 2014***

# Certificate

This is to certify that the project entitled **"Using Semantic Similarity-based Clustering to Facilitate Web Service Discovery "** is a bonafide work carried out as part of the course **Web Services (IT466)**, under my guidance by,

1. Apoorva Chandra S (11IT09)
2. Dilip Mallya (11IT19)
3. JNVK Chaitanya (11IT32)
4. R Kaushik (11IT65),

students of VII Sem B.Tech (IT) at the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, during the academic semester **July - November 2014**, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, at NITK Surathkal.

Place:

Date: Signature of the Instructor

# Declaration

We hereby declare that the project entitled **"Using Semantic Similarity-based Clustering to Facilitate Web Service Discovery "** submitted as part of the partial course requirements for the course **Web Services (IT466)** for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal during the **July - November 2014** semester has been carried out by us.

We declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

Further, we declare that we will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Instructor.

**Signature of the Students:**

1. Apoorva Chandra S (11IT09)

2. Dilip Mallya (11IT19)

3. JNVK Chaitanya (11IT32)

4. R Kaushik (11IT65)

Place:

Date:

**Abstract**

The objectives set during this project was to build a web service search tool that will accept a query from the user and return the web services relevant to the query entered by the user from a dataset of web services present. Using the concept of semantic similarity the dataset of WSDL documents are clustered in an aglomerative, hierarchical fashion. Thus the discovery of the service most relevant to the query is simplified.

**Keywords:** Web Services, Service Discovery, WSDL, Semantic Similarity, Clustering, Aglomerative

# Contents

# List of Figures

# 1 Introduction

Web services provide vital access to software functions through the web i.e an HTTP connection.Basically a service is created and published for use and is used by a Consumer for their application or for other purposes. Web Service Discovery is the process of finding a Web Service that can be used to perform a specified task.

Publishing a Web service involves creating a software product (or *artifact*) and making it available for Consumers to use. Web Service Publishers also provide the endpoint for the web service with an interface description using the Web Services Description Language (WSDL) so that a consumer can use the service.

These WSDL documents are XML-based, and are interface-definition files that be used for describing the functionality offered by a web service. These can be used to automatically find a service that matches a Consumer's need as these files are XML-based, i.e. are machine-readable. A WSDL file contains a description of how the service can be called, what parameters it expects, and what data structures it returns. It is to a Web Service, what a method signature is in a Programming Language.

# 2 Literature Survey

To overcome problems in discovering web service using semantics, Reddy et al propose a mechanism of clustering WSDL files which provide interface information (like input, output, port, messages and binding information). First similarity between web services, is calculated, using the information in WSDL files, then tag similarity is calculated. A new concept of Relevance Feedback (RF) is introduced to use the response of the users on the search results to improve the precision and recall capabilities of the system. Feedback is important to make corrections, changes and improvements in processing, also providing valuable information to the user to improve their queries by providing suggestions in the form of alternative queries.

## 2.1 Problem Statement

To build a web service finder that will accept a query from the user and return the web services relevant to the query entered by the user from a dataset of web services present The list of relevant services will be returned from the clusters formed of the services present in the dataset.

## 2.2 Objectives

1. To find the name of a service, Message names and types, port type operations types used by a service by using regular expression matching
2. To compute the message names and types,types used by the serice and the port operations similarity using the formula suggested by Reddy et al (Refer 2.1)
3. To find the overall similarity between two web services by adding the above mentioned similarities
4. To form clusters of services based on the similarities betweeen the services by setting a threshold
5. To perform hierarchical clustering on the already formed clusters to give a higher level of clarity regarding the services
6. To build a user friendly front end for a service discovery tool.

# 3 Methodology

## 3.1 System Architecture

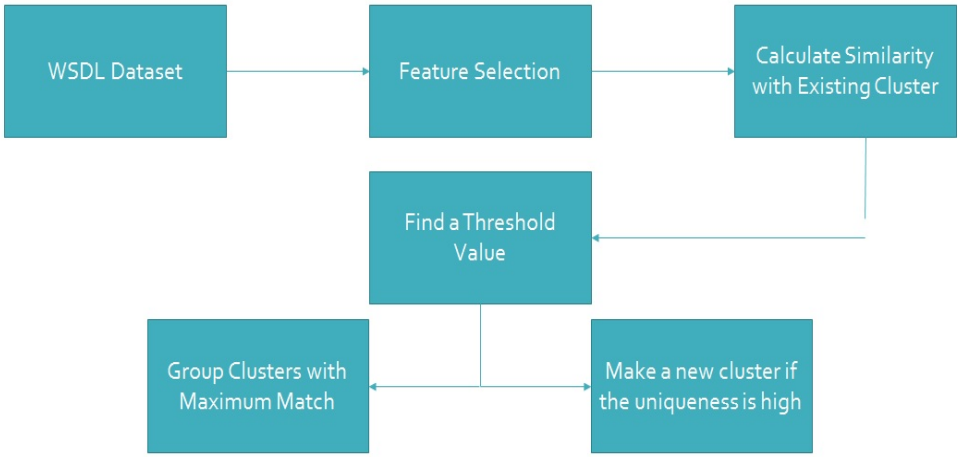


Figure 3.1: *Workflow Proposed*

We have a dataset which has a collection of web services. Then for each service present in the dataset the following are extracted from the service using regular expression matching :

1. Name of the service
2. Message names and types
3. Port operations
4. Types used by the service

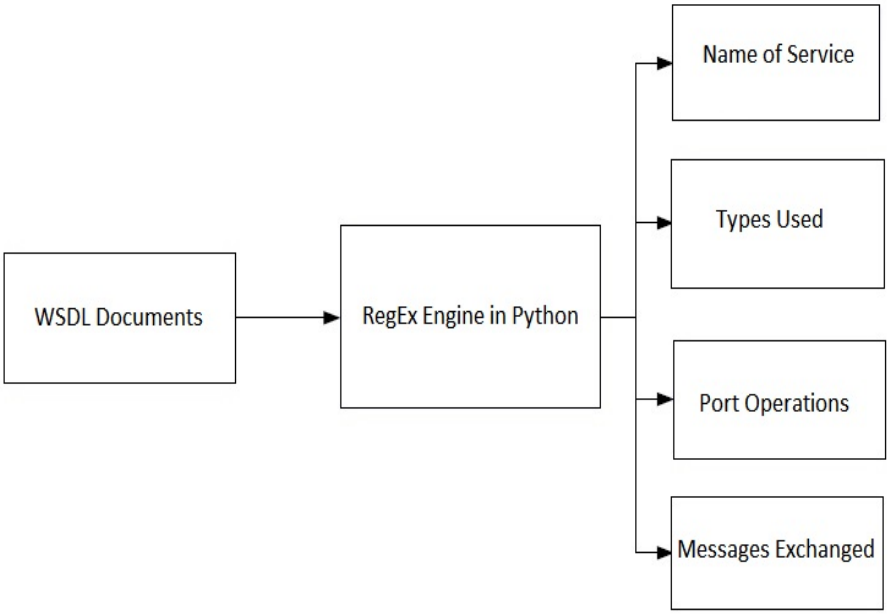


Figure 3.2: *Finding Features*

Once all these features are extracted (Refer Figure 3.2) then a feature vector for each of the above mentioned features is built for each service. Then every new service is compared with the clsuter heads of the existing services for similarity. The message name and type, port operations and the types used similarity is computed using the formula below.

$$sim_{type}(ws_i, ws_j) = \frac{2 * Match(Type_{wsi}, Type_{wsj})}{|Type_{wsi}| + |Type_{wsj}|}$$

The overall similarity is a weighted sum of all the induvidual similarities mentioned above. The value of the weights was established empirically.

Once the similarities with all cluster heads is obtained (Refer Figure 3.3) then the maximum similarity from the list of similarities is compared with the threshold value(which is 0.9) in our case. If the similarity is greater than 0.9 then the service is clubbed in one of the existing clusters else a new cluster is formed.

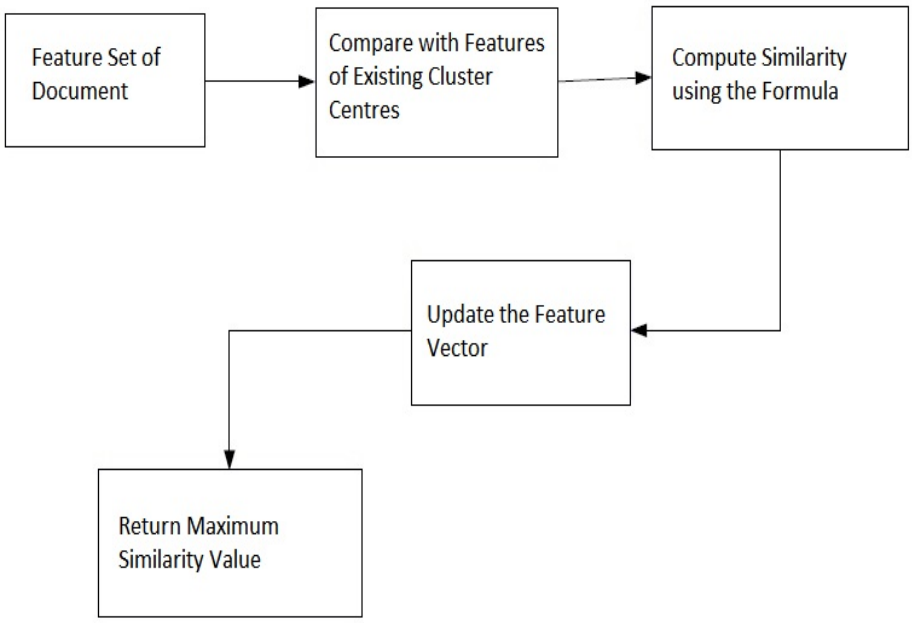


Figure 3.3: *Finding Similarities with Existing Clusters*

When all the documents are clustered the clusters are refined. Here all the cluster with just one service in them are merged and these services are ousted into the cluster with which they resemble the maximum without a threshold. (Refer Figure 3.4)

After this refining process, a step of hierarchical clustering is carried out. This clustering is done to improve service discovery time.

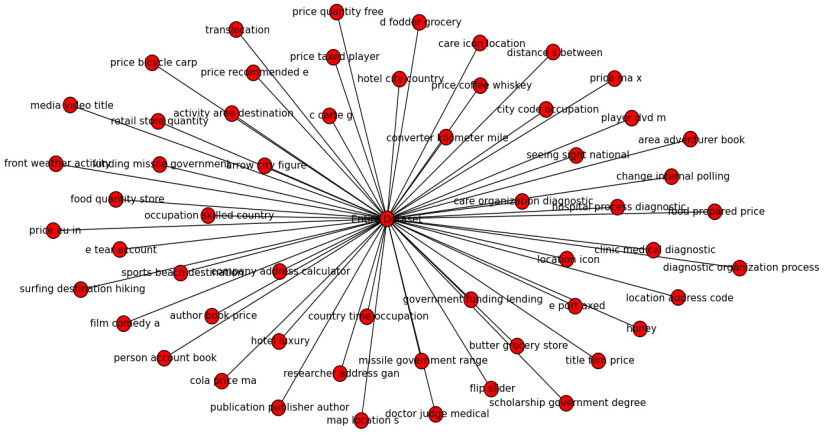

Figure 3.4: *Clusters Generated with Optimised Weighting*

# 4 Implementation

## 4.1 Process

This project involves the clustering of various wsdl documents based on the data types, message names , message types, port types and names used by the services. So as a part of preprocessing the above mentioned properties for each wsdl document had to be extracted. To extract the above mentioned features a regular expression engine was created which would extract operation names, port names, names of the wsdl document and the mesaage names and types. Regular expression for the pattern o fthe occurence of theses attributes of each file were formed and the features were extracted. Thesefeatures of each file were stored in an associative array so that they could be extracted without delay whenever required without processing in a sequential fashion. In the paper that was chosen for implementation the formula for calculating similarity between datatypes used in the wsdl, the message names and types and the operation names in mentioned below.

$$sim_{type}(ws_i, ws_j) = \frac{2 * Match(Type_{wsi}, Type_{wsj})}{|Type_{wsi}| + |Type_{wsj}|}$$

Using the above mentioned formula the similarity for data types , message names and types and port types was calculated for each wsdl file in the dataset. After that the net similarity between two wsdl documents was computed as a weighted average of the above mentioned similarities. The rationalebehind the weighted average scheme will be explained in the innovation part of the report.

For the purpose of clustering the first wsdl file encountered was chosen to be the first cluster centre. The clustering process proceeds as follows : similarity between a wsdl file under consideration and each cluster center is found out and is compared against a fixed threshold. If above the threshold then the wsdl file is joined into the cluster with which it has a maximum similarity else it is formed as a new cluster.

The center of the clusters does not change as the cluster size increses so one cluster may have wsdls pertaining to more than one topic or one domain.After the clustering process all the clusters with just one wsdl file are inserted into clusters which they match the most, with the old cluster which they previously occupied being deleted.

The clusters are named according highest frequency operation names that are found

in the wsdls accomadated in that particular cluster. This is because most of the wsdl files in the dataset have names derived from the IP and port number where they have been hosted. These operation names are obtained by camel case separation of words from the operation names.

## 4.2 Tools Used

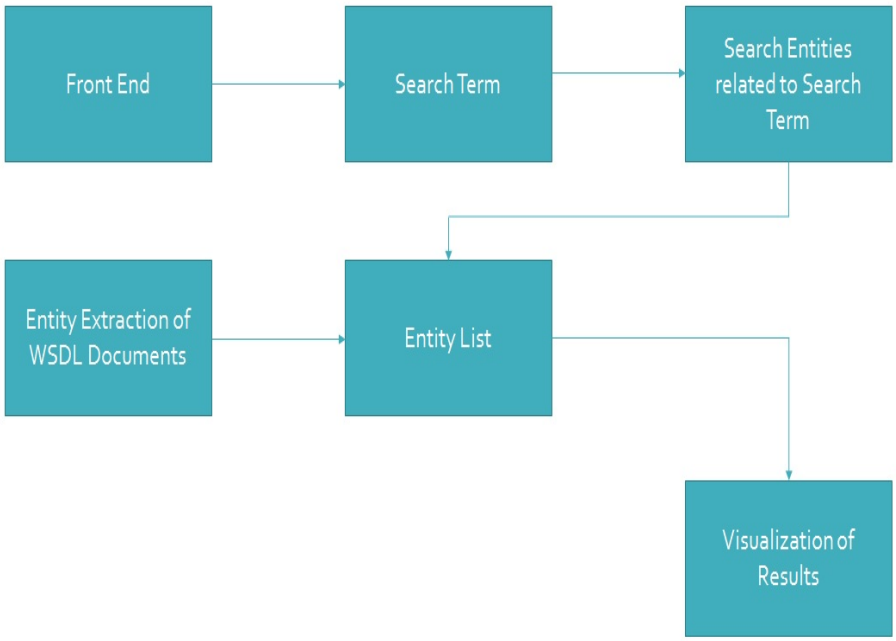


Figure 4.1: *User View of Process*

The source code for extracting the attributes, performing the clustering and naming the clusters is written in Python. The packages used along with simple python were the natural language processing toolkit, used for english dictionaries and computation of frequency distributions, a network package which helps in the easy creation of graphs and jsons.

The visualization is done with the help of a flask server coupled with a python form processing logic HTML5 and CSS3 The visulaization consists of the following parts :

1. A collapsible-force layout graph whihch gives an overall view of all clusters and the view of the largest cluster which matches with the serch query term entered by the user

2. A bar chart indicating the relative size of the clusters

3. A word cloud indicating the relevant tags related to that particular searrch query term entered by the user.

## 4.3 Innovation

Following are the innovations in the effort of implementing the chosen paper:

1. Instead of computing overall similarity as the sum of all similarities(message name,types used, port names) the similarity is computed as a weighted average with the largest weight assigned to the operation names and the smallest weight assigned to the data types that are used by the wsdls This is because message types give more information about the activities of the web services rather than the types used and the effort here was based on the information gain of each of the attributes

2. Alternative naming by using the frequent tags generated by seperating the camel cased operation names. This is because the naesof the wsddl files in some of the datasets are such that they are derived from the IP and PORT numbers where they were hosted. Genreally the name of a wsdl file is an indicator of the type of the service but in cases where the name of the file cannot be used the operation names can be used

# 5 Results and Analysis

The datasets that have been considered for clustering :

1. A dataset with 1076 documents - Number of clusters formed is : 56 , time taken to cluster the dataset - 6 minutes The naming of the clusters in this dataset was done using the name of the wsdl files itself.

2. A dataset with 12749 documents - Number of clusters formed is 340 , time taken for the clustering process is 3 hours and the naming of the clusters formed from the files of this dataset was done using the operation names.This was becuase most of the services in this huge dataset were named after the ip and the port addresses where the services were hosted

3. A dataset with 569 wsdl documents - Number of cluster formed out of this dataset was 40. Time taken to cluster was around3 minutes.The naming was done using the operation names.

The names provided to the clusters of the second dataset were not as good as the first dataset becasue many files in the second dataset do not have names relevant to the function they perform.

The advantages associated with the algorithm and innovations proposed is that it has no dataset limitations and can work on any dataset without any changes being made to the source code.

The associated disadvatages is that the overall method has a very large space complexity because of the large number of data structres used. With the increase in the size of the dataset the time complexity also increases by a large margin, so scalability is also an issue.

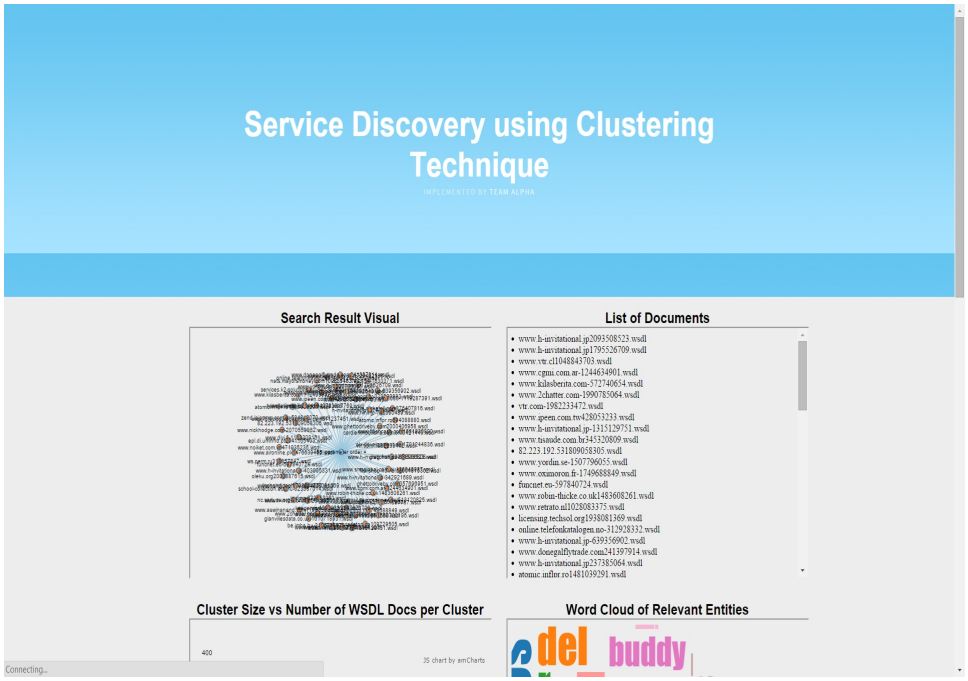


Figure 5.1: *Visualisations - Dashboard Created*

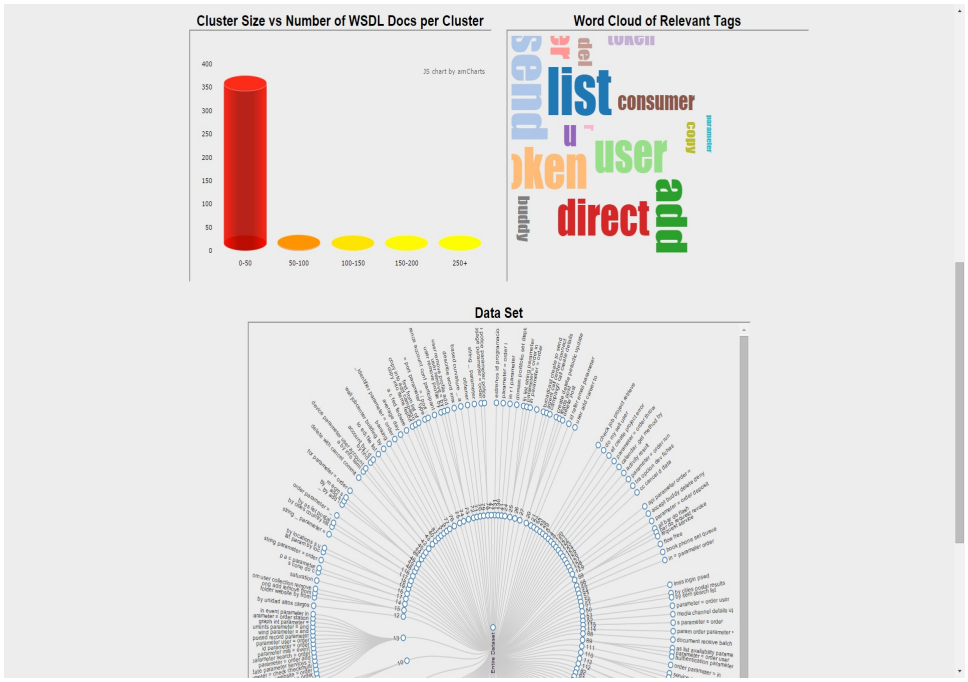


Figure 5.2: *Visualisations - Dashboard Contd.*

Apart from these, another visualization was developed to check the inter cluster distance and plot the clusters accordingly. This visualization was able to indicate the clusters which need to be broken down and reclustered again.

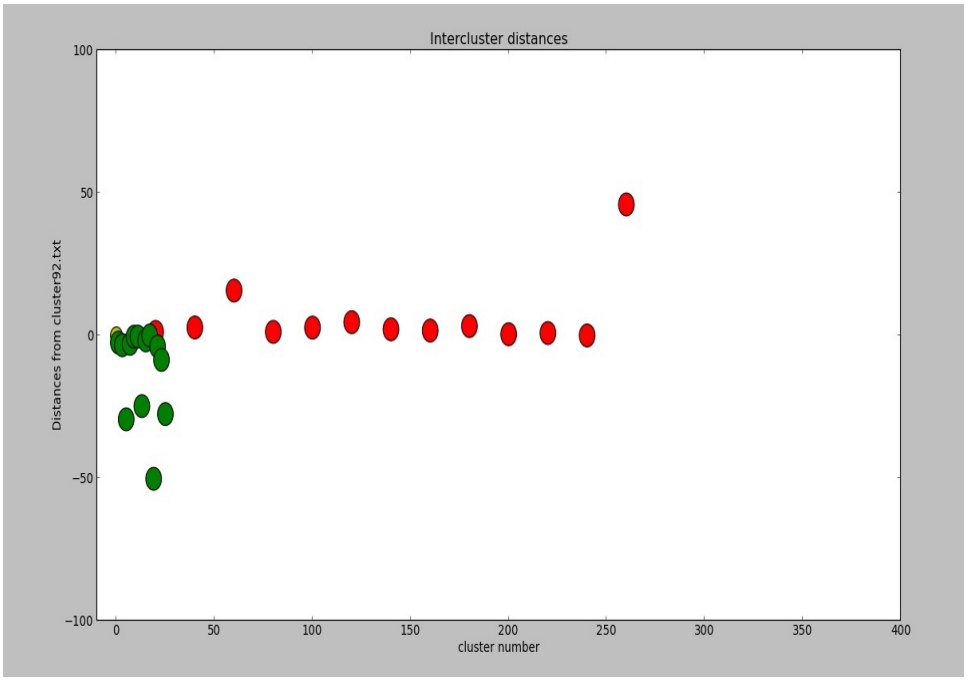


Figure 5.3: *Inter Cluster Distances Visualisations*

# 6 Conclusion

The general conclusions that we could draw out of clustering all these documents are :

1. For comparison of similarity between the clusters choosing service names was not a pratical approach as there could be dataset like the second dataset that was mentioned above.

2. Usage of a lexical similarity to compare message names, types and operation names may improve the clustering. But usage of lexical similarity after the clusterring process to name the first level hierarchical clusters and retrieval of the relsuts would definitely improve precision and recall

3. The coefficient of similarity computed between the cluster cannot be the sum of all similarities but a weighted sum of all similarities , weights which could be decided and computed using a genetic algorithm

As future work on the same topic that is intelligent clustering of wsdl files computing purity of clusters and feature selection from wsdl files using a genetic algorithm may also be included.

# 7 References


1. P Ravinder Reddy and Damodaram A, "Web Services Discovery based on Semantic Similarity Clustering ", *Sofware Engineering (CONSEG), 2012 CSI Sixth International Conference on.* IEEE, 2012

2. Angelos Hliaoutakis, Giannis Varelas, Epimeneidis Voutsakis, Euripides G.M. Petrakis and Evangelos Milios."Information Retrieval by Semantic Similarity", *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(3):5573, July/Sept. 2006