Deanship of Graduate Studies

Al-Quds University

Analysis Study of Classification Techniques for Web Services

Duaa Waleed Ata Faroun

M.Sc. Thesis

Jerusalem –Palestine

1439/2017

Analysis Study of Classification Techniques for Web Services

Prepared by:

Duaa Waleed Ata Faroun

B.SC from Al-Quds University, Palestine

Supervisor: Dr. Rashid Jayousi

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in computer science/ Department of Computer Science / Faculty of Graduate Studies/ Al-Quds University.

1439/2017

Al-Quds University
Deanship of Graduate Studies
Computer Science Department

Thesis Approval

Analysis Study of Classification Techniques for Web Services

Prepared by: Duaa Waleed Ata Faroun
Registration No.: 21011268

Supervisor: Dr. Rashid Jayousi

Master thesis submitted and accepted .Date: 7/10/2017

The names and signature of the examining committee members are as follows:

1- Head of Committee: Dr. Rashid Jayousi Signature: .....

2- Internal Examiner : Dr. Nidal Kafri Signature: .....

3- External Examiner : Dr. Derar Eleyan Signature: .....

Jerusalem - Palestine

1439/2017

Dedication

To my parents, for their love, infinite support and encouragement,

To my beloved husband, without his caring, and support it would not have been possible,

To my brothers, sisters, friends and colleagues,

To all of you I say big "thanks".

Duaa Waleed Ata Faroun

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: ……..……

Duaa Waleed Ata Faroun

Date: 7/10/2017

Acknowledgment

First praise is to Almighty Allah, Lord of all creatures, the Most Gracious, Most Merciful, for his graces and blessings throughout all my life. Without Him, everything is nothing.

My sincere thanks for my supervisor Dr. Rashid Jayousi, for his sincere efforts, interest and time he has kindly spent to guide my research.

I am very grateful to all professors at Al-Quds University-Computer Science department, for the time they have spent to teach me.

Finally, I would like to thank my husband. His support, encouragement, quiet patience and unwavering love were undeniable.

Abstract

The internet is an important part of our life and as the number of web services is increasing it is becoming more difficult to find and compose web services. Data mining can help in this area as it can provide a model for clustering and classifying web services. In this research we conduct a model for clustering and classification of web services and present comparative study of classification techniques for web services. Web services are clustered based on their Web Service Description Language (WSDL).In this research two types of clustering were used for the purpose of comparison; Semantic Clustering and Non-semantic clustering. After getting the clusters we have performed five classification techniques (Neural network, Decision tree, Naïve Bays, SVM and KNN) for both previous clusters and compare the result.

In this research we have implemented clustering and classification model on a set of web services (22) which were first preprocessed by a set of preprocessing algorithms such as parsing, tokenizing, filtering stop word and others, then the similarity between web services was calculated using two similarity techniques: Cosine similarity and semantic similarity. The resulted similarity matrix presents the input for the clustering algorithm K-Medoids that used to build a model for clustering web services.

Following that, five classification techniques were implemented for the clustered data to compare their accuracy.

The results showed that the accuracy of semantic based classification was better than Non-semantic classification for all classification algorithms.

For semantic based classification Neural Network had the best accuracy with 95.7% then Naïve Bayes with accuracy 92.9% followed by SVM with accuracy 87.5%. Decision Tree and KNN had the lowest accuracy of 85.7% and 81% respectively.

Where for classification that not based on semantic similarity Neural Network also gave the best result with accuracy 89.5%. SVM and Naive Bayes had the same accuracy 85.7%. Where KNN and Decision tree had the lowest accuracy 71.4%.

دراسة تحليلة لطرق تصنيف خدمات الانترنت

اعداد الطالبة: دعاء وليد عطا فرعون

اشراف: د. رشيد جيوسي

الملخص

أصبح الانترنت جزءاً مهماً من حياتنا وأصبحت صفحات الانترنت عنصراً أساسياً لمختلف المؤسسات وبشكل فردي أيضاً. ومع هذا التطور السريع والتقدم في تصميم صفحات الانترنت ظهرت خدمات الانترنت وهي عبارة عن نوع من تطبيقات الويب التي تقدم خدمة الكترونية بين تطبيق واخر او نظام وأخر بحيث يرسل التطبيق طلب لخدمة الانترنت التي تقوم بدورها بارجاع النتيجة للتطبيق.

مع ازدياد عدد خدمات الانترنت وتطورها ظهرت الحاجة الى ترتيب هذه الخدمات أو فرزها بطريقة تسهل البحث عنها وتحديد الملائمة منها، عملية التنقيب في البيانات لاستخراج المعرفة منها لها دور مهم في هذا المجال بحيث تسهل تقسيم خدمات الانترنت الى مجموعات وتسهيل استرجاعها.

لذا فاننا قمنا من خلال هذا البحث باستخدام تقنيات التنقيب عن البيانات لبناء نموذج لتقسيم البيانات الى مجموعات وتصنيفها حسب هذه المجموعات، كما قمنا بعمل دراسة على مجموعة من تقنيات تصنيف خدمات الانترنت لتحديد كفاءة كل من هذه التقنيات.

ولبناء النموذج المقترح لتقسيم خدمات الانترنت الى مجموعات تم الاعتماد على الملف الوصفي لكل خدمة انترنت وهو عبارة عن ملف مكتوب باللغة الترميزية الوصفية لخدمات الانترنت (WSDL) بحيث يحتوي على وصف لهذه الخدمة مثل اسمها, البروتوكولات المستخدمة, نوع البيانات

و الرسائل المتبادلة. بحيث قمنا في هذا البحث باستخدام تلك الملفات لاستخراج البيانات الاساسية منها ثم قمنا بتطبيق عملية تقسيم لهذه الملفات التي تمثل خدمات انترنت الى مجموعات، ولقد تم عمل هذا التقسيم بطريقتين بهدف المقارنة؛ الطريقة الاولى هي الطريقة التقليدية باستخدام خوارزمية التقسيم (K-Medoids) حيث اعتمد التقسيم على مدى الترابط بين الملفات بالاعتماد على الكلمات الموجودة وتكرارها دون النظر الى معانيها. أما الطريقة الثانية فكانت أيضا باستخدام خوارزمية التقسيم (K-Medoids) ولكن بالاعتماد على معاني الكلمات في قياس مدى الترابط والتشابه بين ملفات خدمات الانترنت.

في هذه الدراسة تم استخدام 22 خدمة انترنت لتصنيفها، حيث تم معاجتها أولا باستخدام خوارزميات مختلفة للتخلص من الترميز الموجود فيها والتخلص من الكلمات غير الضرورية مثل أدوات الربط والكلمات المحجوزة ثم تم حساب التشابه والترابط بين الملفات كما سبق ذكره بطريقتين مختلفتين الطريقة الأولى هي بحساب التشابة باستخام (Cosine similarity) والطريقة الثانية باستخام التشابه المعنوي للكلمات (Semantic similarity)، ثم قمنا باستخدام مصفوفة التشابه الناتجة كمدخل لخوارزمية التقسيم (K-Medoids) التي تقوم ببناء نموذج التقسيم وتقسيم الخدمات الى مجموعات متشابهة.

بعد الانتهاء من عملية تقسيم ملفات خدمات الانترنت الى مجموعات, تم الاعتماد على هذه المجوعات لبناء نموذج تصنيفي وذلك باستخدام عدة تقنيات تصنيفية بهدف عمل دراسة عليها وايجاد الافضل، حيث تم استخدام التقنيات التالية: (K-Nearest Neighbor, Decision Tree, Naïve Bayes and Neural Network) .

أظهرت نتائج الدراسة أن التصنيف المعتمد على معاني الكلمات كان افضل من التصنيف العادي لجميع خوارزميات التصنيف حيث حقق (Neural Network) افضل كفاءة والدقة في التصنيف المعنوي لخدمات الانترنت والتصنيف العادي لخدمات الانترنت دون الاعتماد على معاني الكلمات كأداة لقياس الترابط والتشابه بينها.

Table of Contents

List of Figures

List of Tables

List of Appendices

Chapter One: Introduction

1.1 Introduction

With the increasing usage of internet and network based applications a need was raised to convert from the traditional software architecture to Service Oriented Architecture (SOA). A web service is a self-contained self-describing application component that can be published and invoked through the web [1]. Everyday new web services are added to the web as well as more requests are expected for such services, therefore in response to such rapid changes improvement on the efficiency web service retrieval techniques should be improved. Many companies provide ways to facilitate the process of locating a web service through search engine [2], [3] but the main problem found in such techniques is that when providers want to register new web service it must specify its category before publishing and that look tedious manual way.

A web service clustering is to group similar web services into clusters based on similarity between them as this can improve the process of locating a web service.

Each web service has its Web Service Description Language (WSDL) stored in a file that is used in this research to cluster web services. Different Clustering can be used such as simple text mining technique, Semantic Technique, or Structural techniques.

Classification techniques can be used to classify a set of test web services and accuracy measurement is used to measure the quality of such classification.

In this research web service clustering and classification has been implemented. A set of web service WSDL documents has been preprocessed so we can extract the most important and frequent words in each WSDL documents. Clustering algorithm then had been used to cluster these documents into a number of clusters. These clusters were used as the base to build classification model. Several classification algorithms have been used in order to compare their accuracy.

The above mechanism was implemented twice first using cosine similarity measure in clustering and the second using semantic similarity measures in order to make use of meaning of the words not only its appearance.

The results showed that the accuracy of semantic based classification was better than Non-semantic classification for all classification algorithms.For semantic based classification Neural Network had the best accuracy with 95.7% then Naïve Bayes with accuracy 92.9% followed by SVM with accuracy 87.5%. Decision Tree and KNN had the lowest accuracy of 85.7% and 81% respectively.

Where for classification that not based on semantic similarity Neural Network also gave the best result with accuracy 89.5%. SVM and Naive Bayes had the same accuracy 85.7% .Where KNN and Decision tree had the lowest accuracy 71.4%.

1.2 Problem Definition

As the number of web services increasing there must be a way to organize similar web services together to improve finding a web service and also when adding new web service it can be classified in efficient manner.

Clustering web service is not a direct process because it depends on the web services description. WSDL contains large amount of data it could vary between 102 and 105 [] so using all element in WSDL would result in massive processing time. Alternatively using specific element from web service could effect on the accuracy of clustering. Also using good similarity measure between two WSDL file is important. For example, two WSDL could be similar based on their structure (input message and output message), size, data type, content …etc. Therefore it is essential to find an algorithm that maximizes the accuracy of classification while minimizing amount of data that need to be analyzed.

1.3 Motivation

Due to the large number of web services that is getting larger and larger it makes it difficult for searching web services to locate an appropriate web service effectively. Web mining can serve powerfully in this area by clustering the current web services and any new web service can be classified to its best clusters, but as there are different classification and clustering techniques an appropriate clustering and classification method is needed to be applied.

For that we propose a model for clustering web service semantically based on the semantic of the words contained in each WSDL then implement different classification techniques and compare the results to find the most accurate classification technique.

1.4 Objectives

The main objectives of this research are to:

1. Cluster web services based on semantic word description.
2. Classify web services with different classification techniques.
3. Build a model for clustering web service semantically and non-semantically, and use this model in the classification phase.
4. Compare the result of classification to find the most accurate classification technique.

1.5 Contribution

In this research web service clustering and classification models have been implemented. These models have been built with different classification algorithms for the reason of comparison. A set of web service WSDL documents has been preprocessed so the most important and frequent words in each WSDL documents can be extracted. Clustering algorithm then has been used to cluster these documents into a set of clusters. These clusters are the base to build classification model. Several classification algorithms have been used in order to compare accuracy of the algorithms.

The above mechanism was Implemented twice first using cosine similarity measure in clustering and second using semantic similarity measures in order to make use of meaning of the words not only its appearance.

In this research, the proposed classification model does not depend on clustered data but it depends on the output of the clustering model with a good accuracy.

This thesis is organized as following: The literature and related work will be discussed in chapter two. Chapter three presents a background of our research includes WSDL definition, clustering and classification definitions and algorithms.

The research methodology followed in chapter four. Experimental results are presented and discussed in chapter five. Finally, the conclusion and future work were discussed in chapter six.

Chapter Two: Literature Review

2.1 Literature Review

Recently Clustering and classification of web services became an important research area as it helps in search and discovery of web services. Many researches have been conducted and many approaches have been proposed.

The simplest approach of classification is manual classification where UDDI are used [4]; when new web service want to be published it must be registered in the UDDI registry, this approach is complex and difficult. However many automatic approaches have been developed. These approaches can be divided into two groups: text mining classification and semantic annotation approach for classification.

Classification based on semantic annotation requires that all web services must have semantic annotation that semantically describes this web services. Semantic Annotation of Web Service Description Language (SAWSDL) defines a mechanism to associate semantic annotations with Web services that are described using Web Service Description Language (WSDL) [5]. There are semi-automatic annotation methods such as MWSAF (METEOR-S Web Service Annotation Framework) [6], ASSAM (Automated Semantic Service Annotation with Machine Learning) [7] and recently appeared IRIDESCENT tool for web service annotation [8]. These methods add semantic annotation to web services and then classify them based on similarity measures. The

disadvantage of these semi-automatic methods is that it requires a manual effort to add the annotations.

Therefore, it seems that the best choice is to use data mining in the classification and clustering of web service. Where variety of methods have been developed and proposed by the research community that will be discussed later in this chapter.

The first phase in web service clustering is extracting elements from web services. Different framework chooses different elements to extract from web services WSDL file. Element of web services such as ports, messages, URl's, semantic and so on. The best framework is the framework that uses semantic and QoS but as mentioned above it is difficult to add these elements to WSDL.

The works of [9, 10] are based on extracting port type, operation, message, names and comments from documentation. Then they adopted Naïve Bayes, SVM and Hyper Pipes to implement web service classification. But the number of elements is too much, preparatory work is a burden. And it is not true that the number of elements for classification is more, the accuracy is higher.

The next phase after extracting elements from WSDL file is measuring similarity between them. Many options are available for calculating similarity the simplest one is Euclidean distance.

In [11, 12] the authors combined text data mining techniques and Tree-Traversing algorithm to cluster Web services based on the WSDL using the service name. The similarity between Web services is measured using the Normalized-Google distance. Normalized-Google distance is the technique used by Google search engine to measure semantic similarity so that when the user types in keywords, it returns Web pages that are related to those keywords.

One of the most important things in Web service mining is calculating the similarity between words extracted from the WSDL and the best way is semantic similarity that depends on lexical English database such as WordNet. The WordNet is a lexical database of English words such as nouns, verbs, adjectives and adverbs [13] that is extremely popular and is easy to use.

WordNet also provided different algorithms for measuring similarity between words . For example information content measures (Lin, Rensik and Jiang) are based on the shared information between two concepts. These measures are shown in chapter Three; Table 3.3. In [14], the framework uses WordNet as the similarity measure for WSDL where web services are clustered based on the semantic similarity score in co-relation with the functional semantic information of service specifications using WordNet 2.1.

With the proposed similarity measure methods, the final stage of web services clustering is the clustering algorithm of the Web services. Many algorithms have been suggested and used. In [11, 12, and 15] Tree clustering algorithms are used to cluster Web services.
In [16] the framework use a well-known clustering algorithm called K-mean clustering. In first step clustering based on the similarity among names and textual description of services using incremental K-Mean clustering algorithm while in the second step semantic of structural features of WSDL documents are used for clustering using Bisecting K-Mean. Also in [17] K-mean algorithm has been used for spectrum clustering of the web service execution network which is constructed from logs.

K-mean is pretty simple algorithm where the framework starts with initial K points as the center of clusters, each Web service will be assigned to the nearest center point. After that, the framework updates the center of each cluster. The steps are repeated until all center points remain unchanged. In [16] the K-mean clustering algorithm is run twice to improve accuracy. After the first run, the framework re-calculates the similarity based on the information in each cluster. Then the framework runs a second K-mean clustering to produce the final result.

In [18] the proposed method is differs from the traditional method. It runs a multi layer clustering method. For each characteristic of WSDL such as type, message, input, and composition patterns, the proposed method runs K-mean clustering on those characteristic individual. For example, the framework in [16] will cluster Web services solely based on its type similarity. On next phase it will cluster Web services solely based on its message similarity.

After clustering on individual criteria is complete, it uses match-based clustering algorithm to group Web services using previous result from individual clustering.

For classification many approaches have been implemented but they depended on defined classes of web services. For example In [19] Crasso implemented text mining for classification of web services using three classification algorithms Rocchio, KNN and Naïve Bayes .Where in [20] and [21] the authors adopted SVM to perform automatic classification. Table 2.1 below shows a summary of literature review.

Table 2.1: Literature review summary

Study	**Study purpose**
Discovering E-Services Using UDDI in SELF-SERV[22]	Clustering web service manually using UDDI register
Web Service Classification Based on Automatic Semantic Annotation and Ensemble Learning [23]	Classification based on Semantic annotation
Semi-automaticWeb Service Classification Using Machine Learning [24]	Clustering of web services by extracting specific element of WSDL file.
Wei Liu and Wilson Wong, "Web service clustering using text mining techniques" [11]	Using text data mining techniques and Tree-Traversing algorithm to cluster Web services

P.R. Reddy and A. Damodaram, "Web services discovery based on semantic similarity Clustering" [14]	Using WordNet to measure similarity between web services.
Mao Li and Yi Yang, "Efficient clustering index for semantic web service based on User preference" [15]	Using Tree algorithm for clustering of web services
Qianhui Liang, Peipei Li, P.C.K. Hung, and XindongWu, "Clustering web services for automatic categorization" [16]	Using K-Mean algorithm for clustering
M.Crasso, A.Zunino, and M.Campo,"AWSC: An approach to web service classification based on machine learning techniques" [19]	Using different classification algorithm for classification of web services.

Chapter Three: Background

This chapter presents background about WSDL, Clustering and classification techniques. it is organized as following: 3.1 presents WSDL definition, history and structure. Clustering algorithms are showed and compared in 3.2. In 3.3 Distance measures that used in clustering are conducted. Finally section 3.4 presents Classification methods.

3.1 Web Service Description language (WSDL)

According to W3C [25], a Web service is an application over Web that is designed to handle the machine-to-machine interaction using a set of characteristics. In 2002 the W3C Web Services Architecture Working Group defined a Web Services Architecture as "a Web service has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP (Simple Object Access Protocol) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards".

So web service architecture consists of three basic characteristics:

1. Interface Described in **WSDL** (Web Service Definition Language)
2. Registered in **UDDI** (Universal Description Discovery and Integration)
3. Interacted via **SOAP** (Simple Object Access Protocol)

The standard architecture that can be driven from those characteristics is presented in Figure 3.1.

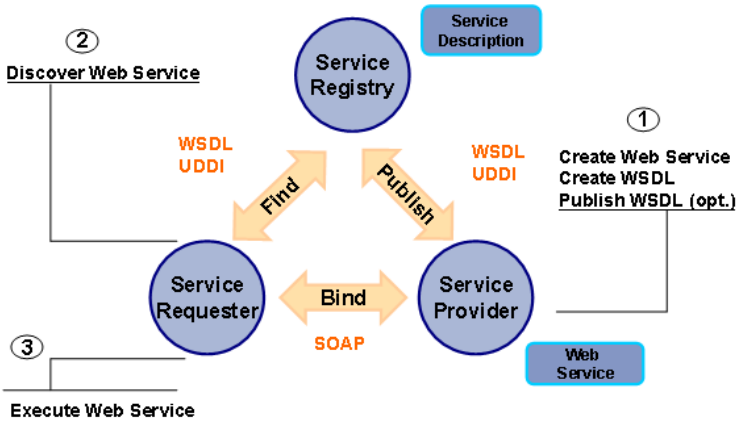


Figure 3.1: Service Oriented Architecture

A WSDL document describes a web service. It specifies the location of the service, and the methods of the service, using these major elements:

1. <types> → defines the data types used in the web service.
2. <message> → defines the data being communicated.
3. <port type> → defines the set of operation that can be performed
4. <binding> → defines the protocol and data format for each port type.

Figure 3.2 shows an Example for a web service component and structure. This web service is named Hello Service and it's available at [26].

```
<definitions name="HelloService"
   targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<message name="SayHelloRequest">
   <part name="firstName" type="xsd:string"/>
</message>

<message name="SayHelloResponse">
   <part name="greeting" type="xsd:string"/>
</message>
```

Messages

```
<portType name="Hello_PortType">
   <operation name="sayHello">
      <input message="tns:SayHelloRequest"/>
      <output message="tns:SayHelloResponse"/>
   </operation>
</portType>
```

Ports

```
<binding name="Hello_Binding" type="tns:Hello_PortType">
   <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
   <operation name="sayHello">
      <soap:operation soapAction="sayHello"/>
      <input>
         <soap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:examples:helloservice"
            use="encoded"/>
      </input>

      <output>
         <soap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:examples:helloservice"
```

Binding

```
               use="encoded"/>
         </output>
      </operation>
   </binding>

   <service name="Hello_Service">
      <documentation>WSDL File for HelloService</documentation>
      <port binding="tns:Hello_Binding" name="Hello_Port">
         <soap:address
            location="http://www.examples.com/SayHello/" />
      </port>
   </service>
</definitions>
```

Service name

Figure 3.2: Web Service Example

The Types used in this web service are built-in data types and they are defined in XML Schema. It contains two messages as shown in the message part the first message is named "SayHelloReequest" that pasess the parameter firstName and the second is "SayHelloResponse" which return a greeting value. This operation which named "sayHello" is shown in the port part, where the input message is SayHelloRequest and output message is "SayHelloRespose". Finally in the Binding Part there is a direction to use SOAP and HTTP transport protocol.

3.2 Clustering

Clustering is dividing data into groups of similar objects. It is based on building a model for clustering the data, data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text

mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others [27].
Clustering serves in many research area specially statistic studies, machine learning and pattern recognition. However recently many clustering algorithms are available each algorithm is suitable to meet the requirement of specific problem.

Clustering algorithms can be divided into two main groups: hierarchical and partitioning methods based on Farley and Raftery (1998) suggestions. Han and Kamber (2001) suggest additional three main categories: density-based methods, model-based clustering and grid based methods [28]. So we can categorize clustering algorithms into the following groups, Table 3.1 shows a summary of clustering algorithms groups.

1. Partitioning Methods

Given a number of desired cluster partitioning methods assume an initial situation where each object belongs to some cluster, then relocate objects by moving them between clusters until reaching the desired situation. The following subsections present two common types of partitioning algorithms [28].

a- **Error Minimization Algorithm**

It aims to minimize a certain criterion of measuring the distance between cluster members and cluster representative. An Example of this criterion is the Square Sum of Error (SSE) which measures the square of the total distances between each object and its representative.

The most common partitioning algorithm that uses error criterion is K-Mean. K-Mean Cluster a set of objects through a set of iteration starting by partitioning the data randomly into a set of clusters (set initially).

b- Graph Theoretic Clustering

Make clustering according to graph. These algorithms present objects as a set of points in space, so distance between all pairs of objects is available. By connecting each object to all its neighbors a complete graph is constructed.

Zahn's clustering algorithm [30] is an example of graph theoretic algorithms it constructs a minimal spanning tree MST for a set of given objects Then it identifies inconsistence edge based on weight edge which is the distance between pairs of objects so inconsistences edge is the edge whose weight is significantly larger than the average of nearby edge weights.

2. Hierarchical Methods

These methods recursively cluster the data in either a top-down or bottom-up model to get a hierarchal decomposition. It can be divided into two groups based on how the hierarchical decomposition is made [29].

A- Agglomerative Hierarchical Clustering : Also called bottom-up, each object initially presents a cluster, it successively merge the objects or clusters closes to one other according to a distance metric until all groups are merged into one cluster or termination condition is set.

B- Division Hierarchical Clustering : Also called top-down, all objects initially belong to one cluster, in each iteration a cluster is split into smaller sub-clusters until each object is presenting one cluster or termination condition is set.

The result of these clustering techniques is a dendogram representing the nested grouping of objects and clusters.

3. Density-based Methods

Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers' points that lie alone in low-density regions (whose nearest neighbors are too far away). So the algorithm will continue growing given cluster until the density in the neighborhood exceeds a given threshold. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature [29].

4. Model-based Methods

These methods represent each group as a concept or class which has specific characteristic [28]. Decision tree and Neural Network are the most well-known model based algorithms. A brief description of these algorithms is explained later in classification section 3.4.

5. Grid-based Methods

These methods partition the space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage of the approach is its fast processing time [31].

Table 3.1: Clustering Algorithms

Algorithm	Examples	Definition	Characteristics
Partitioning	• K-Mean • K-Mediod • Zahn's clustering algorithm	Relocate objects by moving them between clusters until reaching the desired situation.	• Simple • Fast for low dimensional data • It can find pure sub clusters if large number of clusters is specified
Hierarchal	CLINK DIANA	Recursively cluster the data in either a top-down or bottom-up model to get a hierarchal decomposition.	• Good for data sets containing non-isotropic clusters. • provide multiple partitioning level • Inability to scale well
Density based	• DBSCAN	Define a cluster as a maximal set of density connected points and discovers clusters of arbitrary shape.	• Cannot handle varying densities • Sensitive to parameters
Grid based	• STING	Group points with many nearby neighbors.	• Useful for clustering very large data sets.
Model based	• Neural Network • Decision tree	Represent each group as a concept or class which has specific characteristic.	• The main advantage is that it can suggest the number of clusters and an appropriate model

3.3 Distance Measures

In order to cluster a set of data we need to measure the similarity or dissimilarity between two objects. There are two main methods to measure similarity: distance measures, similarity measures.

3.3.1 Distance Measures:

Distance measure is used to determine how similar two objects are by calculating the distance between them. Many distance measures are available [28] Table 3.2 presents a brief description:

Table 3.2: Distance Measures

Distance Measure	Description
1. Minkowski	Distance Measure for Numerical Attribute
2. Distance Measure for Binary Attribute	The distance is calculated based on contingency table
3. Distance Measure for Nominal Attribute	Two ways : • Simple matching • Create binary attribute from nominal attribute and measure binary distance
4. Distance Metrics for Ordinal Attribute	Used when attributes are ordinal (the sequence of value is meaningful)
5. Distance Metrics for Mixed –type Attribute	Combine and distance measure

3.3.2 Similarity Function:

Similarity function measures similarity by comparing two vectors t_a and t_b [32] There are many similarity functions such as:

1. **Cosine Measure**

It is one of the most popular similarity measures applied to text document. It measures the angel between two vectors t_a and t_b as :

$$sim\,(t_a, t_b) = \frac{t_a \,.\, t_b}{|t_a| \,.\, |t_b|}$$

Where t_a and t_b are m-dimensional vectors over the term set T= {t_1, t_2… t_n}. Each document is presented as a dimensional vector with the weight of the terms it contained. The cosine similarity is non- negative and bounded between [0 -1] because the weight of each term in a document is non-negative. So if two documents are identical their vectors will have the same orientation with similarity cosine of 1. On the other hand if two documents are different their vectors will be perpendicular so the cosine similarity is 0.

2. **Person Correlation Measure**

Pearson's correlation coefficient is another measure of the extent to which two vectors are related. There are different forms of the Pearson correlation coefficient formula.

This metric measures how highly correlated are two documents and is measured from -1 to +1. A Pearson Correlation Coefficient of 1 indicates that the documents are perfectly correlated but in this case, a score of -1 means that the documents are not correlated.

3. **Extended Jaccard Measure**

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient

is computed in such a way that the number of shared terms divided by the number of all unique terms presented in both documents. It measures similarity between two text documents by comparing the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but are not the shared terms. The formal definition is:

$$\boldsymbol{sim}\,(\boldsymbol{t_a}, \boldsymbol{t_b}) = \frac{\boldsymbol{t_a}\,.\,\boldsymbol{t_b}}{|\boldsymbol{t_a}|^2 + |\boldsymbol{t_b}|^2 - \boldsymbol{t_a}.\boldsymbol{t_b}}$$

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the $t_a = t_b$ and 0 when t_a and t_b are disjoint, where 1 means the two documents are the same and 0 means they are completely different.

4. Dice Coefficient Measure

Dice coefficient measure is defined as two times the number of terms which are common in the compared strings and divided by the total number of terms presented in both strings. It can be expressed as:

$$\boldsymbol{sim}\,(\boldsymbol{t_a}, \boldsymbol{t_b}) = \frac{\boldsymbol{2t_a}\,.\,\boldsymbol{t_b}}{|\boldsymbol{t_a}|^2 + |\boldsymbol{t_b}|^2}$$

3.3.3 Semantic Similarity Function: :

Semantic similarity measure is a central issue in artificial intelligence, psychology and cognitive science for many years. It has been widely used in natural language processing, information retrieval, word sense disambiguation, text segmentation, question answering, recommender system, information extraction and so on [33].

Recently many semantic similarity measures have been developed, these measures aims to measure the semantic distance between two concepts based on defined ontology which is a large database of concept and their relation. There are much ontology available it can be classified as general ontology and domain ontology that represents words of specific domain such as medical ontology e.g. UMLS, *SNOMED and MeSH.*

However in this research WordNet ontology has been used because it is a general ontology which is a general ontology that attracted a great concern in recent year and measures based on it shows powerful result [34].

Several methods of determining semantic similarity have been purposed; In general it can be divided into four groups:

- Path based measures: measure the distance between two concepts based on the hierarchy structure of the ontology (is-a, part-of). The length of the path linking the two concepts is computed to measure how similar they are.
- Information content (IC) measures: measure the distance between two concepts based on the information content.
- Feature based measures: In this measure each term is described by a set of terms specifies its feature. So the distance between two concepts is measured as a function of their features.
- Hybrid measures combine the structural measures and some of other measures techniques.

Differences between these methods are shown below in Table 3.3 [33].

Definition of related concept in the above measures for two concepts c_i and c_j are：

(1) len (c_i ,c_j): the length of the shortest path from synset c_i to synset c_j in WordNet.

(2) lso (c_i ,c_j): the lowest common subsumer of c_1 and c_2

(3)depth(c_i): the length of the path to synset ci from the global root entity, and depth(root)=1.

(4) deep_max: the max depth(c_i) of the taxonomy.

(5) hypo(c): the number of hyponyms for a given concept c.

(6) node_max: the maximum number of concepts that exist in the taxonomy.

(7) sim (c_i ,c_j): semantic similarity between concept c_i and concept c_j .

For two compared concepts c_i and c_j in taxonomy, the length of the shortest path from concept c_i to concept c_j can be determined from one of three cases.

- Case1: c_i and c_j are the same concept, thus c_i , c_j and $lso(c_i, c_j)$ are the same node. We assign the semantic length between c_i and c_j to 0, ie. $len(c_i, c_j)=0$.
- Case2: c_i and c_j are not the same node, but ci is the parent of c_j . thus $lso(c_i, c_j)$ is c_i . We assign the semantic length between ci and c_j to 1, ie. $len(c_i, c_j)=1$..
- Case3: Neither c_i and c_j are the same concept nor c_i is the parent of c_j , we count the actual path length between ci and c_j , therefore 1<= 2*deep_max.

Table 3.3: Comparison of Different Semantic Similarity

Category	Principle	Measure	Feature	Advantage	Disadvantage
Path based	Function of path length linking the concepts and the position of the concepts in the taxonomy	Shortest Path	count of edges between concepts	Simple	two pairs with equal lengths of shortest path will have the same similarity
		Wu & Palmer's Measure	path length to subsumer, scaled by subsumer path to root	Simple	two pairs with the same lowest common subsumer($lso(c_1,c_2)$) and equal lengths of shortest path will have the same similarity
		Leakcock & Chodorow's	count of edges between and log	Simple	two pairs with equal lengths will be similar

		Li's	non-linear function of the shortest path and depth of lso	Simple	two pairs with the same lso and equal lengths of shortest path will have the same similarity
IC based	The more common information two concepts share, the more similar the concepts are.	Rensik	IC of lso	Simple	two pairs with the same lso will have the same similarity
		Lin	IC of lso and the compared concept	take the IC of compared concepts into considerate	two pairs with the same summation of $IC(c_1)$ and $IC(c_2)$ will have the same similarity
		Jiang		take the IC of compared concepts into considerate	two pairs with the same summation of $IC(c_1)$ and $IC(c_2)$ will have the same similarity

Feature based	Concepts with more common features and less non-common features are more similar	Tversky	compare concepts' feature, such as their definitions or glosses	take concept's feature into considerate	Computational complexity. It can't works well when there is not a complete features set.
Hybrid based	combine multiple information sources	Zhou	combines IC and shortest path	well distinguished different concepts pairs	Parameter to be settled, turning is required. If the parameter can't be turned well it may bring deviation.

As shown in Table 3.3 each semantic similarity measure has its characteristics. Path based measures considered simple because it takes in consideration only the position of concepts and the path length linking the concepts. But the density of pairs (how much information the share) is difficult to reflected. Information content similarity measures take in consideration the common information that two concepts are shared, it is effective but the distance between concepts cant reflected. Feature based measures take the assumption that two concepts with more common features and less non-common features are more similar. The disadvantage of this measure is that it needs a complete set of features between two concepts. Finally Hybrid based measures combine multiple information sources (distance and shared information) so it can distinguish different concepts pairs.

3.4 Classification

Also called Supervised learning is the process of predicting the class of a set of test data based on a classification model that have been built from a set of training records that labeled with specific class [35]. Classification task begins with a historical clustered data set to build the classification model and then assign any new object to specific class. It has many applications in many fields such as customer segmentation, business modeling, marketing, credit analysis, biomedical, medical diagnosis… etc. For example it can be used in business to categorize bank loan applications as either safe or risky

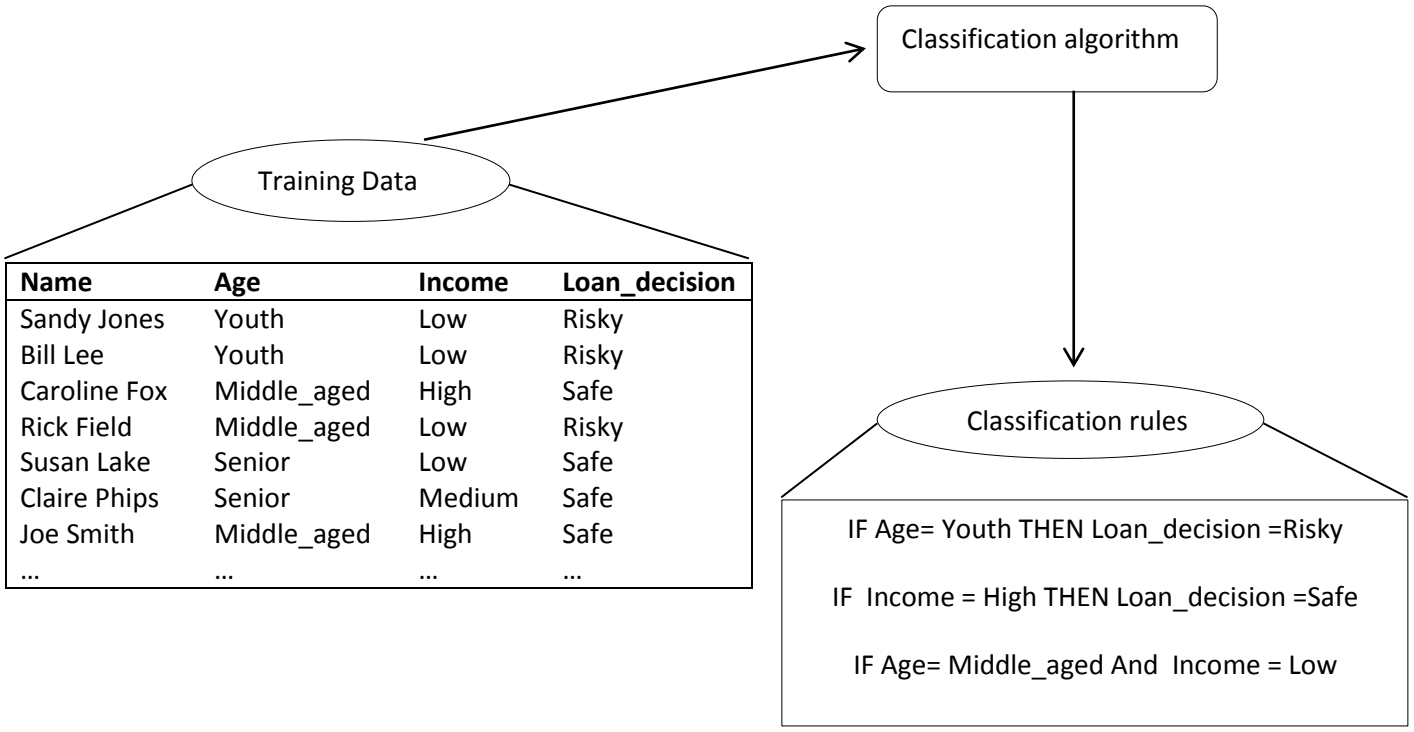
Classification process is a two-phase process that consists of learning phase; where a model is constructed from the training instances and classification phase where the model is used to assign a class labels for a given data instances.

The first phase (Learning phase) starts with a training data set where each instance of data should have a well-known class. Then data preparation and preprocessing is done where preprocessing functions can be used to prepare data for classification that include removing unnecessary words, preparing the data in a format suitable for classification, converting data from one form to another and selecting most important features that will represent the data perfectly. The result of this phase is a classification model that will be used in the classification phase.

In the next phase (classification phase) the classification model is used to classify a test set. Then the accuracy of the classifier is estimated by calculating the percentage of the test instances that have been classified correctly by the classifier. If the accuracy is acceptable the model can be used to classify any new data.

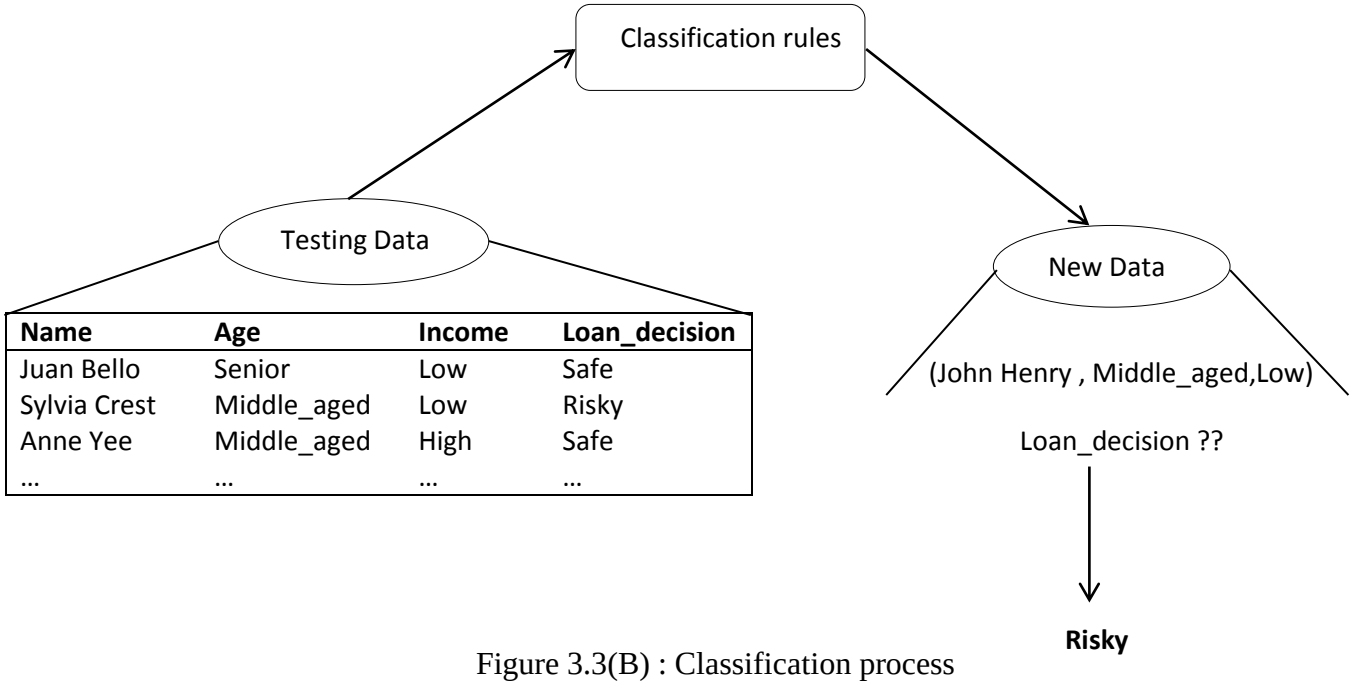
Figure 3.3 shows an example of the classification process for a set of data. The data set consists of four attributes (Name, age, Income and Loan_decision) where Loan_decision is the class attribute. Learning phase is shown in Figure3.3 (A) where the classification algorithm will use a set of training data set to build a classification model; here the classification model is a set of rules that identify loan application as being either safe or risk. In Figure 3.3 (B) the testing phase

where the test data set is used to measure the accuracy of the classification model (classification rules) that have been used to classify the test data set.



Name	Age	Income	Loan_decision
Sandy Jones	Youth	Low	Risky
Bill Lee	Youth	Low	Risky
Caroline Fox	Middle_aged	High	Safe
Rick Field	Middle_aged	Low	Risky
Susan Lake	Senior	Low	Safe
Claire Phips	Senior	Medium	Safe
Joe Smith	Middle_aged	High	Safe
…	…	…	…

Figure 3.3(A) : Classification process



Name	Age	Income	Loan_decision
Juan Bello	Senior	Low	Safe
Sylvia Crest	Middle_aged	Low	Risky
Anne Yee	Middle_aged	High	Safe
…	…	…	…

Figure 3.3(B) : Classification process

Many techniques are available to get training set and test set but the most used technique is dividing data set by using two-thirds for training set and the remaining third for testing set.

The most important part in the classification process is choosing of the classification algorithm. It considered critical step as the accuracy of classification model depends on the classification algorithm.

There are many classification algorithm have been developed these algorithm are described below.

3.4.1 Decision Tree:

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each root node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. An Example of decision tree [36] is presented in Figure 3.4 for the training set presented in Table 3.4.

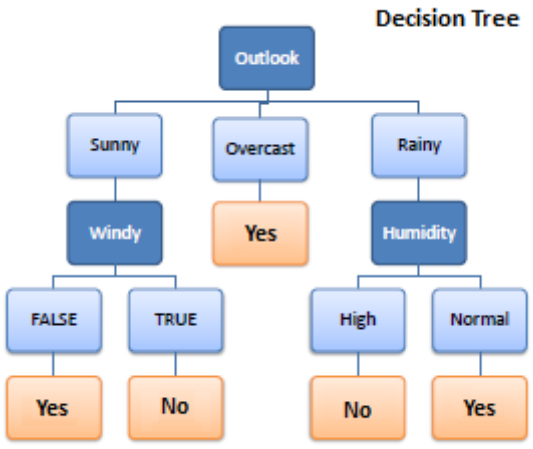


Figure 3.4: Decision Tree Example [36]

Table 3.4: Training Set of Decision Tree [36]

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

The data set in that example has four predictor attributes (Outlook, Temperature, Humidity and Windy) and one target attribute (Play Golf) which has two values: Yes and no.

To build a decision tree one of the predictor must be selected that will give the best gain. In that case Outlook attribute has been selected because it gives the best gain ration; this node will have three branches (Sunny, Overcast and Rain). For the branch "Overcast" all the instances on that subset have the same target attribute (play golf = Yes) so it will be a leaf node labeled by "Yes". For the branches "Sunny" and "Rainy" the algorithm will repeat the previous steps to choose the best new predictor from the remaining attributes (Except Outlook).

The pseudo code shown in Figure 3.5 is for Iterative Dichomeiser (ID3) algorithm which has been developed J. Ross Quinlan by during the late 1970s and early 1980s. It starts with original set "Examples". On each iteration it will calculate the information gain for every unused attribute to select the attribute with the highest information gain. Then it splits the set "Example" by that attribute .the algorithm will continue splitting the data considering only unused attribute.

The algorithm will stop splitting on two cases:

- If all elements of the subset are belonging to the same class then the node is denoted by that class and turned on into leaf node.
- If there is no unused attribute to select from but the subset contain mixed classes then it will turn on the node into leaf node labeled with the most common class in the subset.

The feature which divides the training set perfectly will be the root of the tree. So finding the root of the tree is an important step, many methods have been developed such as information gain [37] that used in ID3 algorithm described above, Gini index [38] that used in CART algorithm.

Decision tree classification algorithm considered one of the most used classification algorithm due to a set of reasons. First its simplicity in implementation and understanding. Second it performs well with large data in a short time, Third It can handle both numerical and categorical data. On the other hand the main problem is building an optimal Decision tree is an NP complete problem [39].

```
Decision_tree  (Examples, Target_Attribute, Attributes)
 Create a root node for the tree
   If all examples are positive, Return the single-node tree Root, with label = +.
   If all examples are negative, Return the single-node tree Root, with label = -.
   If number of predicting attributes is empty, then Return the single node tree Root,
   with label = most common value of the target attribute in the examples.
   Otherwise Begin
      A ← The Attribute that best classifies examples.
      Decision Tree attribute for Root = A.
      For each possible value, vi, of A,
         Add a new tree branch below Root, corresponding to the test A = vi.
         Let Examples(vi) be the subset of examples that have the value vi for A
         If Examples(vi) is empty
            Then below this new branch add a leaf node with label = most common
target value in the examples
         Else below this new branch add the subtree Decision_tree (Examples(vi),
Target_Attribute, Attributes – {A})
   End
   Return Root
```

Figure 3.5: Decision Tree Pseudo Code

3.4.2 Nearest Neighbor:

Also called lazy learning or instance based method as it delay the generalization process until classification is done. For that it needs less computation time during training phase

than eager-learning algorithm such as (Naïve base, Decision tree and Neural Network) but more computation time during classification process.

K-Nearest Neighbor algorithm classifies each object by finding it's K-nearest neighbor from the training set calculating the distance using one of the distance metric function, some of the most used distance metrics are shown in Table 3.5 [39]. However this will only work with numerical values. In cases such as text classification another metric, such as the overlap metric (or Hamming distance) can be used.

Table 3.5: Methods to Find the Distance between Instances

Minkowski	$D(x,y) = (\sum_{i=1}^{m} \lvert x_i - y_i \rvert^r)^{1/r}$
Manhattan	$D(x,y) = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Chebychef	$D(x,y) = \max_{i=1} \lvert x_i - y_i \rvert$
Euclidean	$D(x,y) = (\sum_{i=1}^{m} \lvert x_i - y_i \rvert^2)^{1/2}$
Canberra	$D(x,y) = \sum_{i=1}^{m} \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$

As shown in Table 3.5 for any two numeric instances $x_i=(x_{i1},x_{i2},\ldots,x_{ip})$ and $x_j=(x_{j1},x_{j2},\ldots,x_{jp})$. The distance between them can be measured using different measures. The general distance measure is Minkowski which can be expressed as:

$$\boldsymbol{D(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g)^{1/g}}$$

Minkowski can be considered as general distance; from it we can get other distances. The commonly used Euclidean can be achieved if g=2 which is the natural distance in geometric interpretation. If g=1 then Manhattan distance is achieved which is the sum of absolute paraxial distances. Finally if g= ∞ we get the greatest of the paraxial distances which is Chebychev distance.

Some other distances are derived from these distance to deal with not standardized data; weighted variable. A special weighted version of Manhattan distance is Canberra distance which divides the absolute differences between variables of two instances by the sum of the absolute variable prior to summing.

Figure 3.6 presents a general pseudo code for Nearest Neighbor algorithm. The Algorithm will use a set of training data X that has a previously known classes labeled by the variable Y to classify unknown class sample x. First it will compute the distance between the new unknown instance and all other instances in the training set d*(X_i, x)* according to a specific distance metric. Then it will use these distances to find the set of k-nearest instances. Finally it will return the resulting class of x which is the most frequent class label of the k-nearest instances.

```
Classify (X, Y, x )
// X: training data, Y: class labels of X, x: unknown sample.
For i=1 to m do
Compute distance(Xi, x)
End for
Compute set I containing indices for the k smallest distances d(Xi, x)
Return majority label for {Yi where i Є I}
```

Figure 3.6: A general pseudo code for Nearest Neighbor Algorithm

3.4.3 Support Vector Machine:

This method was introduced by Vapink and his colleagues to solve the problem of data classification and regression [40]. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression [41]. It presents input data set as two sets of vectors in an n-dimensional space, an SVM will create a separating hyper-plane in that space, that maximizes the *margin* between the two data set. The margin between the data set is calculated by constructing two parallel hyper-planes one on each side of the separating hyper-plane, which are "pushed up against" the two data sets [40]. A good separation is achieved by the hyper-plane that has the largest distance to the neighboring data points of both classes that will give more generalization for all possible data. These hyper-planes are found by using the support-vectors and margins, that is calculated using math ticks that includes using lagrangian formula and Karush-Kuhn-tucker (KKT) condition.

The simplest case is when we have two separable classes (Calss1 and Class2) with two attributes A1, A2 (two dimension data 2-D) For a set of data D which is presented as (x_1,y_1), (x_2,y_2), …, (x_d, y_d). Where x_i is a training instance with a class y_i. This case is shown in Figure 3.7.

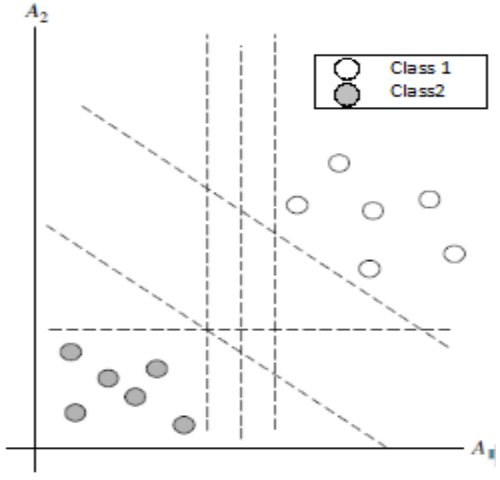


Figure 3.7: Two separable classes data set

As shown in Figure 3.7 there are infinite numbers of possible separating lines (called hyper planes for generalization in a multi-dimensional data). SVM will search for the two hyper planes that have the maximum margin. Figure 3.8 presents two possible cases for two hyperplanes that separate the data set D into two classes .But as shown in that figure the margin in the two cases is different , it is in (B) more that in (A), So SVM will choose the hyper-planes in (B) Because it has the optimal hyper-planes with the largest margin between [31].

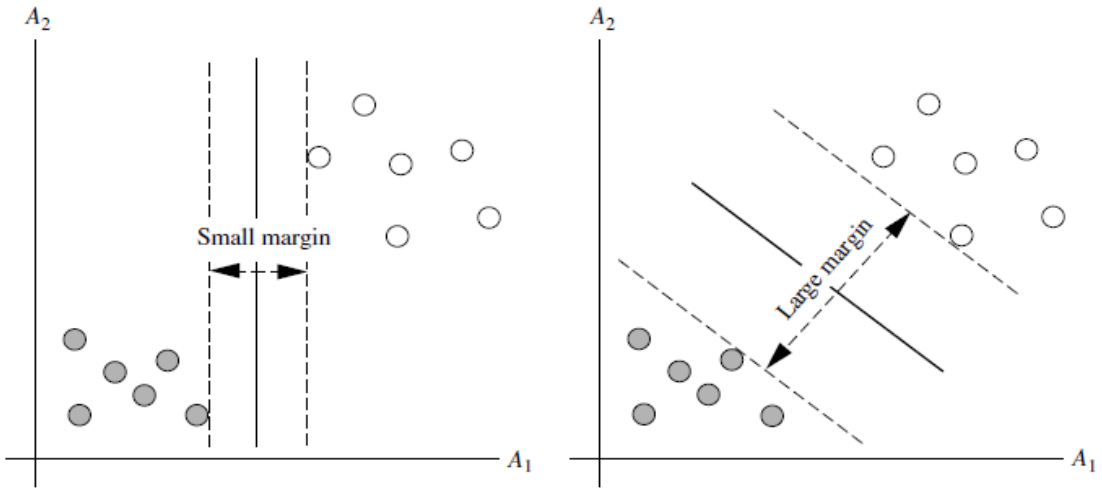


Figure 3.8: Maximum Margin [31]

3.4.4 Neural Network:

Neural networks represent a brain metaphor for information processing. These models are biologically inspired rather than an exact replica of how the brain actually functions. Neural network consists of many processing units called neurons. Neural computing refers to a pattern recognition methodology for machine learning. The resulting model from neural computing is often called an artificial neural network (ANN) or a neural network. [42]

Neural Network is a set of connected input and output units where each connection in the network has its weight. Figure 3.9 shows an example of neural network [31].

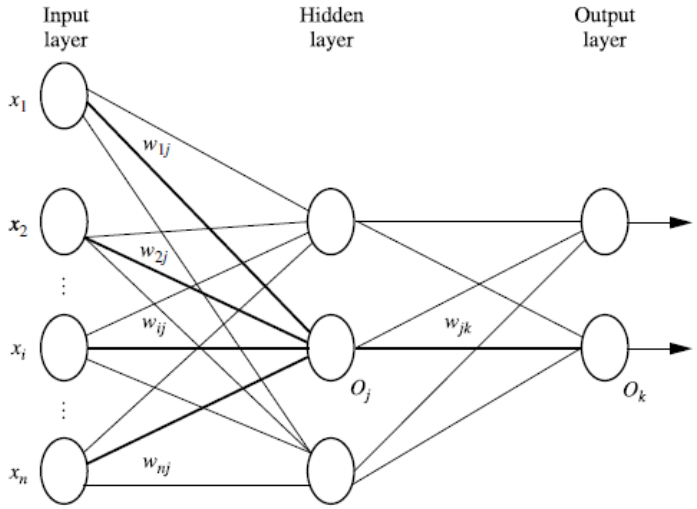


Figure 3.9: Neural Network [31]

Each layer consists of a set of units. Input values are passed through the input units in the input layer and they are weighted, these weights are send to the units in the hidden layer, there can be more than one hidden layer. The weighted output of the hidden layer is fed to the final units that make the output layer which presents the network prediction for given data.

For a network that consists of one hidden layer it called two-way neural network. And a network that has two hidden layers will be called three way neural network and so on.

A neural network that has no one of the weights cycles back to a previous layer is called feed forward. On the other hand there are recurrent neural networks that have back connections to the previous layers.

The most popular Neural Network algorithm is backpropagation algorithm which gains repute in 1980s. Backpropagation algorithm performs learning on a multilayer feed-forward neural network that consists of input layer, hidden layer and output layer.

3.4.5 Statistical Learning Algorithm:

Statistical techniques have an explicit probability model that shows the probability of each object to the class it belong to.

Bayesian Network is the most known and used statistical learning algorithm. Naïve Bayes classifier is a statistical classification that based on the assumption of independence among predictors (features of each instance or object); simply the presence of one feature in one class is independent on any other features [39].

Bayes' theorem is named after Thomas Bayes, a nonconformist English clergyman who did early work in probability and decision theory during the 18th century. Let X be an data instance with unknown class, let H be a hypothesis that X is belonging to a specific class N, the algorithm will compute the probability p(H|X) the probability that data instance X belong to class N, this value is called posterior probability which is shown in equation (1).

$$\boldsymbol{P(H|X)} = \frac{\boldsymbol{P(X|H)P(H)}}{\boldsymbol{P(X)}} \qquad (1)$$

- $P(H|X)$ is the posterior probability of *class* N given X.
- $P(H)$ is the prior probability of *class*.
- $P(X|H)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(X)$ is the prior probability of *predictor*.

Naive Bayes classifier uses the above theorem. For example for a data set D consists of a set of instances and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X =(x_1, x_2, x_3,…,x_n). If there are m classes, C_1, C_2, … , C_m. Given a instance**,** X**,** the classifier will predict that X belongs to the class having the

highest posterior probability, conditioned on X. That is, the naive Bayesian classifier predicts if X instance belongs to the class C_i if and only if:

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i.$$

Thus the maximum posteriori *hypothesis* based on Bayes' theorem is:

$$\boldsymbol{P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}}$$

3.5 Evaluation

Evaluation is an important step of data mining classification process in order to measure and qualify the classification model. Three main performance evaluation criteria are used: learning curves, confusion matrix and Receiver Operating Curves (ROC).

The confusion matrix presents the number of correct and incorrect prediction made by the model compared with the real values [43]. For two classes it can be presented as shown in Figure 3.10

		Current Classes	
Predicted Classes		**True Class**	**False Class**
	True Class	True Positive	False Positive
	False Class	False Negative	True Negative

Figure 3.10: Confusion Matrix for two classes

In that matrix there are four terms:

- True positive: are the positive instances that are correctly classified by the classifier.
- True Negative: are the negative instances that are correctly classified by the classifier.
- False Positive: are the positive instances that are incorrectly classified by the classifier.
- False Negative: are the negative instances that are incorrectly classified by the classifier.
- True positive and true negative tell when the classifier is getting things right while false positive and false negative tell us when the classifier is getting thing wrong.

From this matrix many measures can be derived such as:

- True Positive Rate: the fraction of positive cases predicted as positive.
- False Positive Rate: the fraction of negative cases predicted as positive.
- True Negative Rate: the fraction of negative cases predicted as negative.
- False Negative Rate: the fraction of positive cases predicted as negative.
- Precision: is the number of correctly predicted value relative to the total number of predictive value. It can be considered as a measure of exactness that what percentage of instances labeled as positive are actually positive. It can be defined as:

$$\boldsymbol{precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

- Recall (equals the true positive rate) the proportion of cases classified as positive in relation to all the positive cases. It can be considered as a measure of completeness that what percentages of positive instances are labeled positive. It is called also sensitivity, it can be defined as:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

- Specificity: it is the true negative rate ; the proportion of negative instances that are correctly identified, it is defined as :
-

$$\boldsymbol{Specificity = \frac{TN}{N}}$$

- F-Measure: (or F-Score) Called the harmonic mean, it combines the precision and sensitivity in equation 2.

$$\boldsymbol{F - Measure = \frac{2 \times TPR \times precision}{TPR + Precision}} \quad (2)$$

These measures and others are presented in Figure 3.11.[43]

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)
	Predicted condition negative	**False negative** (Type II error)	**True negative**
		True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$

Figure 3.11: Confusion Matrix and other Measures

ROC Curves also is a way to evaluate classification model. It is FPR vs. TPR curve with FPR present the x axis and TPR is the y axis for the different possible cut points of a diagnostic test. In that curve the perfect classification point is (0,1) while the point (1,0) mean that all cases classified in correctly. An ROC curve Example is shown below in Figure 3.12 [44].

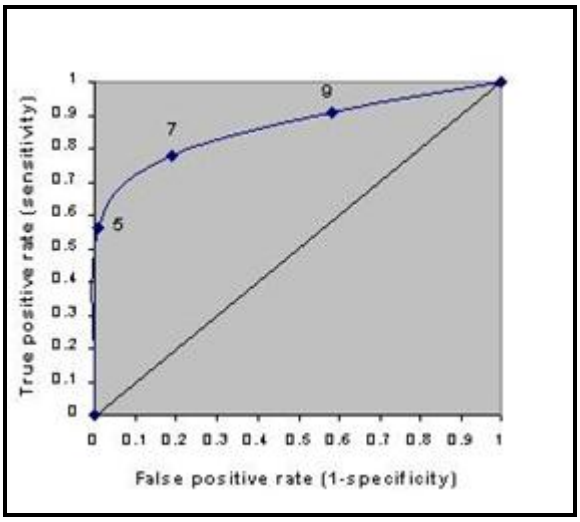


Figure 3.12: ROC Curve Example

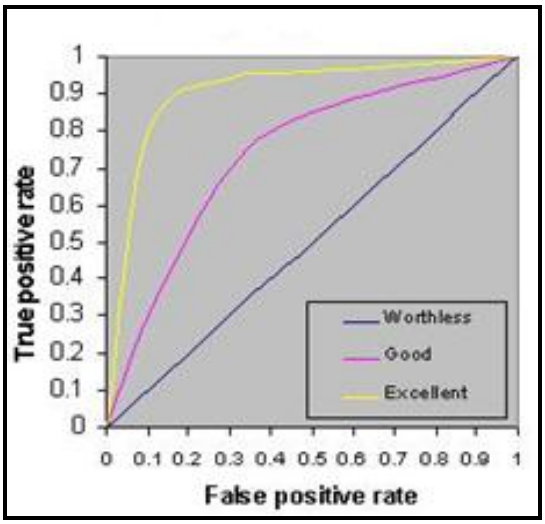


Figure 3.13:Comparing ROC Curves

An ROC curve demonstrates several things:

1. It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
2. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
3. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test. Figure 3.13 shows three ROC curves representing excellent, good, and worthless tests plotted on the same graph.
4. The area under the curve is a measure of text accuracy.

Chapter Four: Methodology

4.1 Introduction

For achievement of the objectives of this research this chapter is divided into six subsections; Data Collection present the collected data, overall approach, WSDL preprocessing, WSDL word extraction and similarity calculation, clustering, classification.

4.2 Collected Data

We utilize the fourth version of OWL-S service retrieval test collection (OWLS-TC4) [45] as the experimental data source, providing 1076 Web Services and 1083 Semantic Web Services from nine different domains (Communication, economy, education, food, geography, medical, simulation, travel and weapon). But Only 22 web services have been selected randomly using a java program that selects four domains and then select a number of web services from each domain proportional to the overall all number of web services in each domain from that test collection. These web services are used in both semantic clustering and non-semantic clustering.

4.3 Overall approach

4.3.1 Clustering Approach:

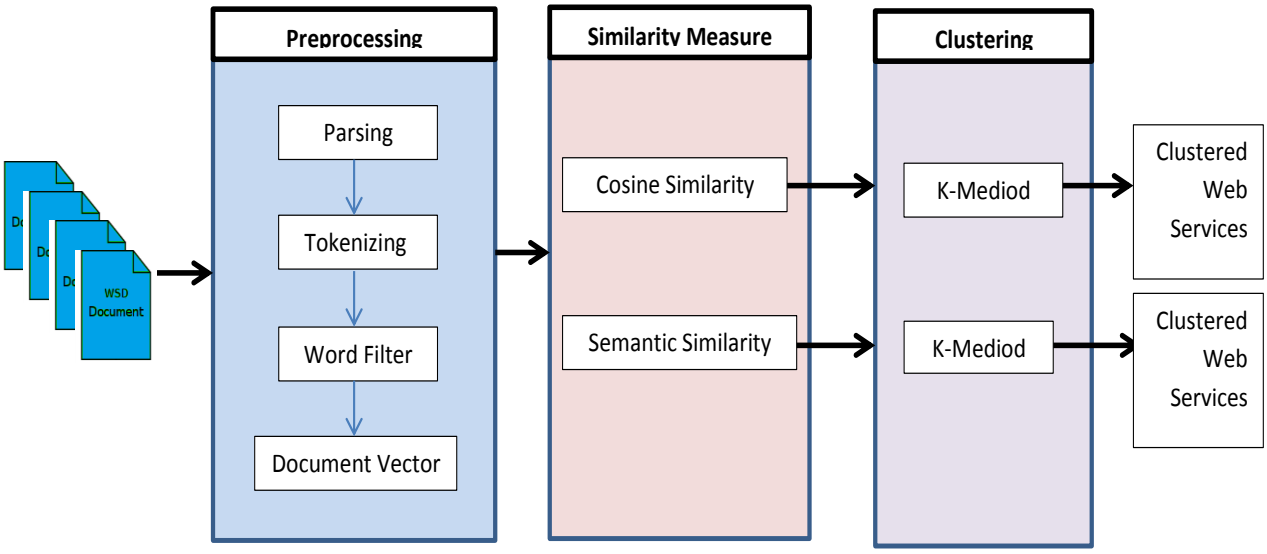


Figure 4.1: Clustering approach

This section will explain the process of web service clustering, which can be seen in Figure 4.1. Web service classification is based on WSDL document. As WSDL is the most widely accepted and used web service description language.

4.3.1.1 Preprocessing:

The process of clustering web service must be passed through several steps. First of all preprocessing functions have been implemented on the WSDL files to parse, remove unnecessary word, tokenize, etc.

All WSDL have been parsed so all tags have been removed then the parsed document will be tokenized so that all words can be transformed into regular English words. For example the word StandardMessageFault is a compound word so the frame work will break all compound words into individual words using tokenizing algorithm that will use

capital letter as breaking point to break the compound word. Also the framework will filter out any words that are xml content words such as String and type. Also words are filtered by stop list [46]. Stop list contains a set of common English connection words, propositions, articles, etc.

Finally the framework will extract most frequent words in each document to get a document vector for each document. Not only most frequent words have to be extracted but also most meaning words in each document. For that the framework will use Term Frequency Inverse Document frequency text mining algorithm TF/IDF. It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. TF/IDF value increases as the number of times a word appears in a document.

- **Term Frequency (TF):** the number of times a term occurs in a document and it reflects term weight. It can be expressed as tf(*t*,*d*); the number of times that term *t* occurs in document *d*.

- **Inverse Document Frequency (IDF):** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$idf(t,D) = log \frac{N}{|\{d \epsilon D : t \epsilon d\}|}$$

- N: total number of documents in the corpus N= |D|
- $|\{d \epsilon D : t \epsilon d\}$: number of documents where the term t appears

- **TF/IDF:**

$$tfidf(t,d,D)=tf(t,d).\ idf(t,D)$$

The framework will extract words with top five TF/IDF score that are not related to WSDL format and structure from each WSDL therefore correctly identifying important words for each WSDL.

We can summarize the preprocessing step in a set of preprocessing functions:

- Parsing: to remove XML tags and only get the text.
- Tokenizing: this include tokenizing words and remove connected words such as
- Remove unnecessary words: this include removing stop words and connecters.
- Extracting words: this is based on the most frequent words in the document to extract the most frequent five words.

4.3.1.2 Similarity Measure :

The Similarity between these files is then measured; two approaches have been applied to measure similarity; Cosine similarity and Semantic WordNet similarity. Cosine similarity is a simple technique that will measure similarity between document vectors based on angular distance. Where WordNet algorithm will measure similarity based on semantics .the framework will use Lin measure which is an information content measure that measures the distance between two concepts based on the information content.

4.3.1.3 Applying Clustering Algorithm:

After that the framework will implement Clustering model using K-Medoids clustering technique. The K-Medoids algorithm has been chosen because there is a need for an algorithm that can accept semantic similarity matrix (it will be given to the algorithm).

4.3.2 Classification Approach :

For both Semantically clustered data and non-semantically clustered data classification techniques will be applied. First the data must be partitioned into two sets: Training set and

testing set. Then five classification techniques will be applied (Decision Tree, KNN, Neural Network and Bays Classifier). Finally, accuracy is measured and compared. Classification approach is shown in Figure 4.2.

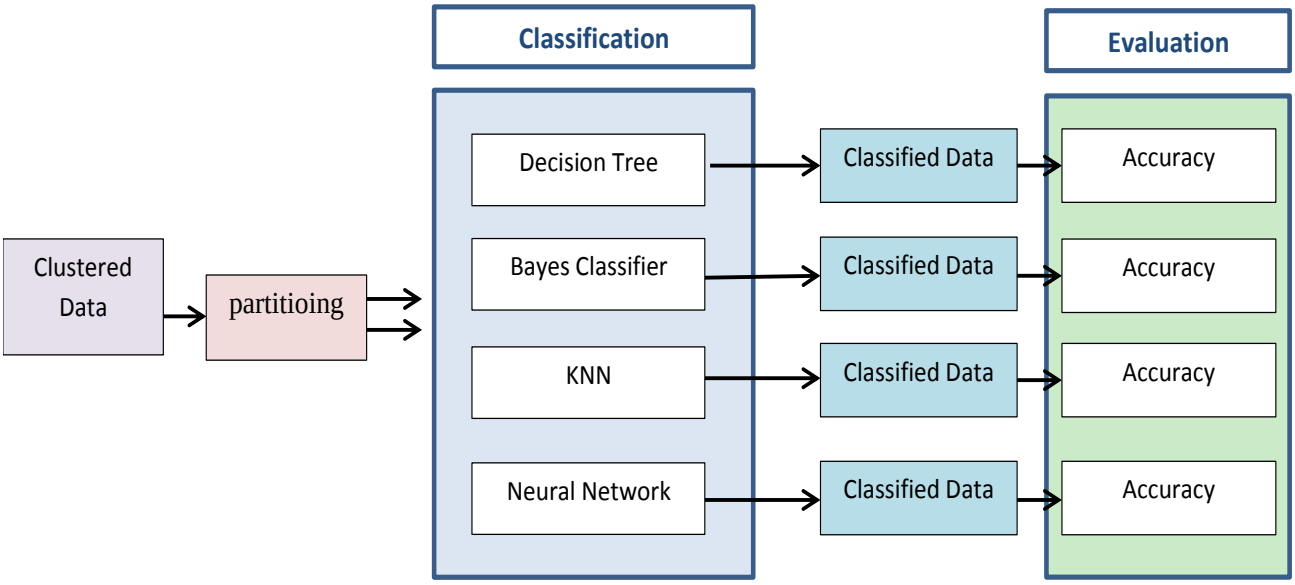


Figure 4.2: Classification Approach

4.3.2.1 Classification:

Classification algorithm will be applied for a test set of the clustered data. The clustered data is divided into two parts (70% of data) Training set that will be used in the training phase of the classification to build the classification model and (30% of data) Test set these are used in the classification phase to evaluate the accuracy of the classification model.

Five classification techniques have been used: KNN, Naïve Bayes, Neural Network, SVM and Decision Tree.

After implementing these techniques for both clustered data the results have been compared based on the accuracy of each algorithm.

4.3.2.2 Evaluation :

Classification approach evaluation has been measured by measuring accuracy of each of the classification algorithm. Also ROC curves have been used to represent confusion matrix and evaluate each classification algorithm.

Chapter Five: Experiments and Results

5.1 Introduction

In order to solve the problem of clustering web services and at the same time improve the clustering results in this research two Experiments have been implemented: Simple text clustering and Semantic Clustering have been done.

For simple text clustering initially a set of web services was randomly chosen from web service collection OWLS-TC4 which includes nine domains: communication, economy, education, food, geography, medical, simulation and weapon. Only 22 web services have been selected from four domains (medical, education, communication and geography) using java program that select a number of web services from each domain proportional to the number of web services in that domain. All WSDL files are parsed using java parser, tokenized and unnecessary words are removed; as a result a Bag of words is created. The result was a set of documents each contain a set of words.

Next KNIME Data mining tool have been used to build the model. First TFIDF algorithm has been used to produce the most five important words in each document Tf-idf value of part of words is seen in Table 5.1. Then cosine similarity measure is used to measure similarity between documents, the resulted similarity matrix is used as an input for K-Medoids algorithm that clusters the WSDL files into four clusters, Figure 4.1 in section 4.3.3 illustrates the model implementation.

Table 5.1: Tf-Idf value for part of words

Word	Web service	Tf-Idf
Hospital	w0	0.230205
Hospital	w1	0.230205
Country	w2	0.340432
Medical	w3	0.089932
Comedy	w4	0.359727
Film	w4	0.30694

The same process is performed for semantic clustering but here after extracting most important words in each document WordNet similarity algorithm is used to produce similarity matrix, the algorithm will read the entire input file and find similarity between then based on the semantic of the words stored on them. The resulted similarity matrix is symmetric with dimension [d,d] where d is the number of documents which is [22,22] where it was implemented to use more WSDL files in the experiment but as the number of web services increases the similarity matrix will be larger and that require more computation time and power.

Also this matrix has been exported to K-Medoids algorithm; for that it had been chosen because it can accept any similarity matrix.

After clustering, classification techniques are used, where the clustered set is partitioned into training set and test set (70% training set and 30% testing set). Classification model is built also on KNIME data mining tool using different classification algorithms in order to compare the results of these algorithms.

KNN, Neural Network, Bayes classifier, SVM and Decision tree are the classification algorithms that have been used for both set of clustered data. The result is shown below in the result section.

5.2 Results

After implementing different classification algorithms on clustered data two measurements are used to measure the performance of classification algorithms; the accuracy of classification and confusion matrix.

5.2.1 Accuracy

Accuracy of classification is the proportion of total number of correct prediction. Table 5.2 shows the accuracy results for each classification algorithm, as shown in the table for classification that based on semantic similarity the results was better for all classification algorithm. Neural Network had the best accuracy with 95.7% then Naïve Bayes with accuracy 92.9% followed by SVM with accuracy 87.5%. Decision Tree and KNN had the lowest accuracy of 85.7% and 81% respectively.

Where for classification that not based on semantic similarity Neural Network also gave the best result with accuracy 89.5%. SVM and Naive Bayes had the same accuracy 85.7% .Where KNN and Decision tree had the lowest accuracy 71.4%.

As shown in Table 5.2 semantic classification had better accuracy results than non-semantic classification for all classification algorithms.

Table 5.2: Accuracy of Classification Algorithms

Algorithm	**Semantic**	**Non semantic**
	Accuracy %	**Accuracy%**
Decision Tree	85.7	71.4
Bayes Classifier	92.9	85.7
KNN Classifier	81	71.4
SVM	87.5	85.7
Neural Network	95.7	89.5

5.2.2 Confusion Matrix:

As mentioned above in section 3.5 confusion matrix basically presents the number of correct and incorrect prediction made by each classification algorithm compared with the real values and many other measurement have been calculated from it.

In this research this matrix has been calculated for each classification algorithm in both experiments (semantic and non-semantic) as an evaluation measurement.

A- Non semantic classification

Table 5.3 shows confusion matrix for Naïve bays classifier for non-semantic set and the result show that for the first cluster "class1": 2 were True positive, 0 were false positive, 4 were True Negative and 1 was False Negative. That means for class1 there is 2 cases predicted to be in this class and it is really belong to that class, 0 cases aren't predicted to be in this class but they are actually belong to this class, 4 cases predicted not to be in that class and they are actually not belong to this class, finally one case predicted not to be in that class but it is actually belong to it. From these values we calculated several measures; recall is 0.66 , precision is 1 , specificity is 1 and finally f–measure is 0.8 .The same is for the second cluster "class2": 2 were True positive, 0 were false positive, 5 were True Negative and 0 was False Negative so on for other classes.

Table 5.4 shows confusion matrix for Neural Network classifier for non-semantic set and the result show that for class1: 2 were True positive, 0 were false positive, 5 were True Negative and 1 was False Negative and so on for other classes. Table 5.5 shows confusion matrix for Decision Tree classifier for non-semantic set. Table 5.6 shows confusion matrix for SVM classifier for non-semantic set. Table 5.7 shows confusion matrix for KNN classifier for non-semantic set. (ND is no defined value because of dividing by zero).

Naïve Bayes

Table 5.3: Naïve Bayes Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	Specificity	F-measure
Class 1	2	0	4	1	0.66	1.0	1.0	0.8
Class 2	2	0	5	0	1.0	1.0	1.0	1.0
Class 3	1	1	5	0	1.0	0.5	0.83	0.66
Class 4	1	0	6	0	1.0	1.0	1.0	1.0

Neural Network

Table 5.4: Neural Network Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	specificity	F-measure
Class 1	2	0	5	1	0.67	1	1	0.8
Class 2	2	0	6	0	1	1	1	1
Class 3	1	1	6	0	1	0.5	0.8	0.66
Class 4	2	0	6	0	1	1	1	1

Decision Tree

Table 5.5: Decision Tree Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	specificity	F-measure
Class 1	4	2	1	0	1.0	0.66	0.33	0.8
Class 2	1	0	5	1	0.5	1.0	1.0	0.66
Class 3	0	0	6	1	0.0	ND	1.0	ND
Class 4	0	0	7	0	ND	ND	1.0	ND

SVM

Table 5.6: SVM Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	specificity	F-measure
Class 1	0	0	7	0	ND	ND	1.0	ND
Class 2	0	0	6	1	0.0	ND	1.0	ND
Class 3	2	0	5	0	1.0	1.0	1.0	1.0
Class 4	4	1	2	0	1.0	0.8	0.66	0.88

KNN

Table 5.7: KNN Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	specificity	F-measure
Class 1	0	0	7	0	ND	ND	1.0	ND
Class 2	2	1	4	0	1.0	0.66	0.8	0.8
Class 3	1	0	6	0	1.0	1.0	1.0	1.0
Class 4	3	0	3	1	0.75	1.0	1.0	0.857

B- Semantic Classification

Table 5.8 shows confusion matrix for Naïve bays classifier for semantic set and the result shows that for class1: 1 were True positive, 0 were false positive, 6 were True Negative and 1 was False Negative and so on for other classes. Table 5.9 shows confusion matrix for Neural Network classifier for semantic set and the result shows that for class1: 2 were True positive, 0 were false positive, 6 were True Negative and 0 was False Negative and so on for other classes. Table 5.10 shows confusion matrix for Decision Tree classifier for semantic set, Table 5.11 shows confusion matrix for SVM classifier for semantic set and Table 5.12 shows confusion matrix for KNN classifier for semantic set.

Naïve Bayes

Table 5.8: Naïve Bayes Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	Specificity	F-measure
Class 1	0	6	1	0.5	1	1	0.67	1
Class 2	1	4	0	1	0.75	0.8	0.85	3
Class 3	0	7	0	1	1	1	1	1
Class 4	0	7	0	1	1	1	1	1

Neural Network

Table 5.9: Neural Network Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	Specificity	F-measure
Class 1	2	0	6	1	0.67	1	1	0.8
Class 2	2	0	6	0	1	1	1	1
Class 3	2	0	6	0	1	1	1	1
Class 4	0	1	5	1	0	0	0.83	ND

Decision Tree

Table 5.10: Decision Tree Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	Specificity	F-measure
Class 1	1	0	6	1	0.5	1	1	0.67
Class 2	2	0	6	0	1	1	1	1
Class 3	2	0	5	0	1	1	1	1
Class4	1	1	6	0	1	0.5	0.86	0.67

SVM

Table 5.11: SVM Confusion Matrix

Class	TP	FP	TN	FN	Recall	Precision	Specificity	F-measure
Class 1	0	0	5	2	0.0	ND	1.0	ND
Class 2	1	0	6	0	1.0	1.0	1.0	1.0
Class 3	2	0	5	0	1.0	1.0	1.0	1.0
Class 4	2	2	3	0	1.0	0.5	0.6	0.666

KNN

Table 5.12: KNN Confusion Matrix

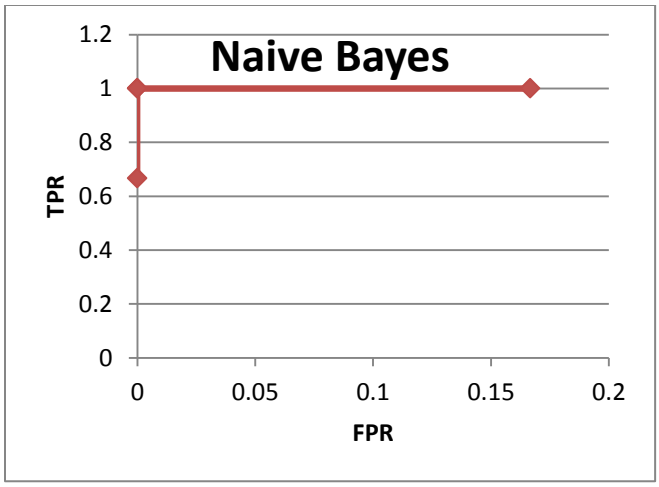
Class	TP	FP	TN	FN	Recall	Precision	specificity	F-measure
Class 1	2	0	4	1	0.67	1	1	0.8
Class 2	2	1	4	0	1	0.67	0.8	0.8
Class 3	1	0	6	0	1	1	1	1
Class 4	1	0	6	0	1	1	1	1

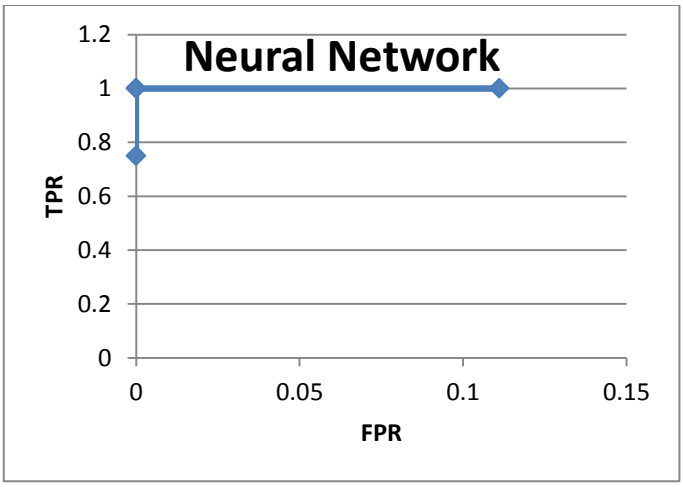
5.2.3 ROC Curves:

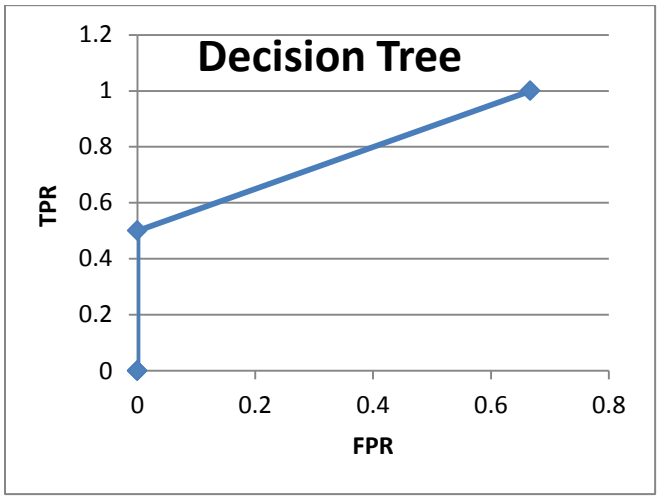
ROC Curves also is a way to evaluate classification model. It is False Positive Rate vs. True Positive Rate curves with FPR present the x axis and TPR is the y axis. In that curve the perfect classification point is (0,1) while the point (1,0) mean that all cases classified incorrectly. The accuracy of a classification model can be measured from the area under the curve.

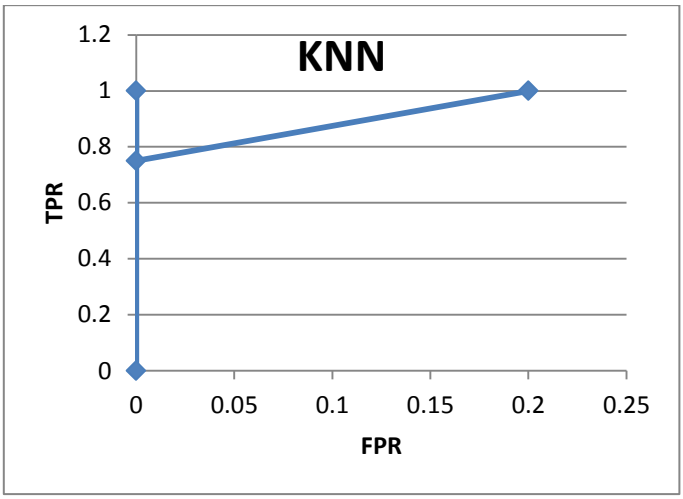
As mentioned above in section 3.5 the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test on the other hand the closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

ROC curves for no semantic classification are shown in Figure 5.1. Neural Network, Naïve Bayes, KNN and SVM have approximately the same classification accuracy where Decision tree has the smallest area under the curve; which mean the lowest accuracy. Figure 5.2 shows that for semantic classification, Neural Network, Naïve Bayes and SVM have similar performance. Decision tree and KNN have lower performance.









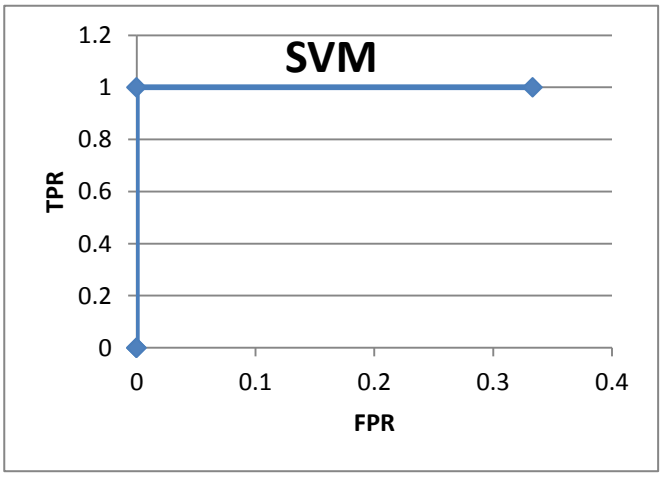
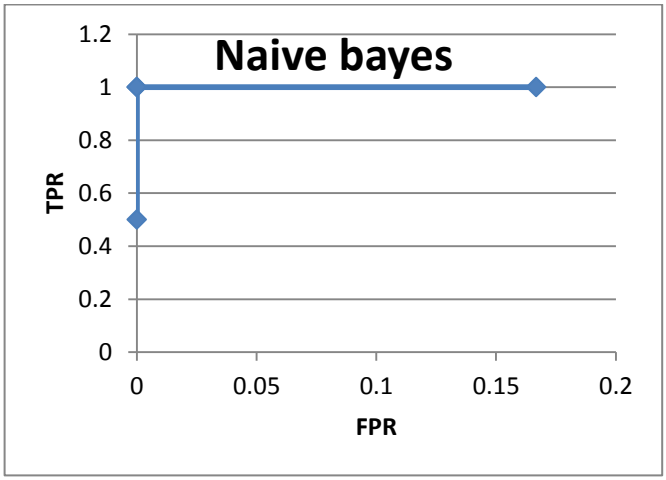
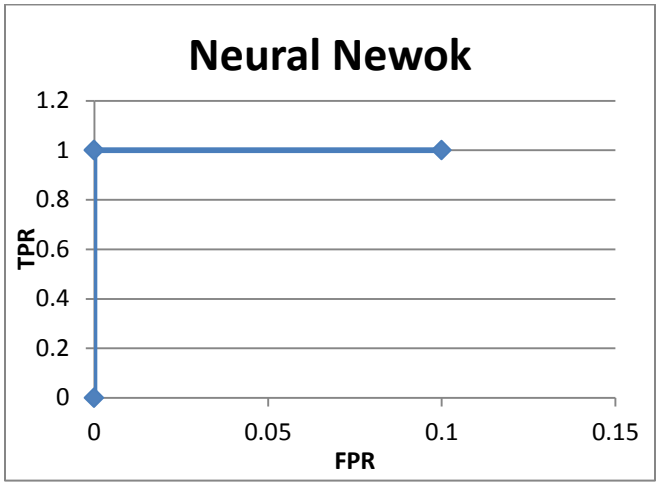
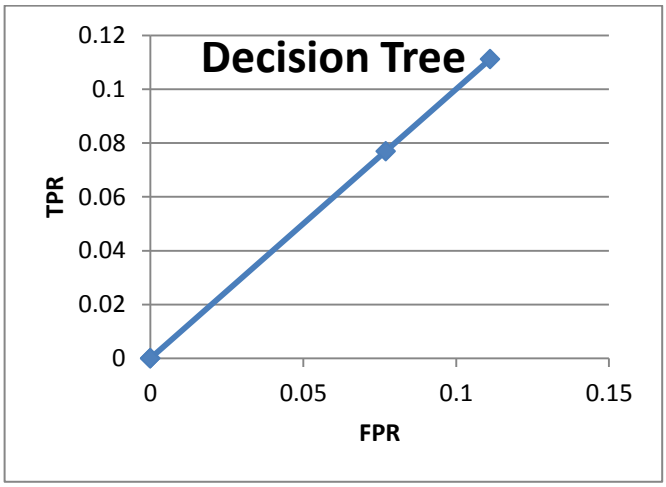
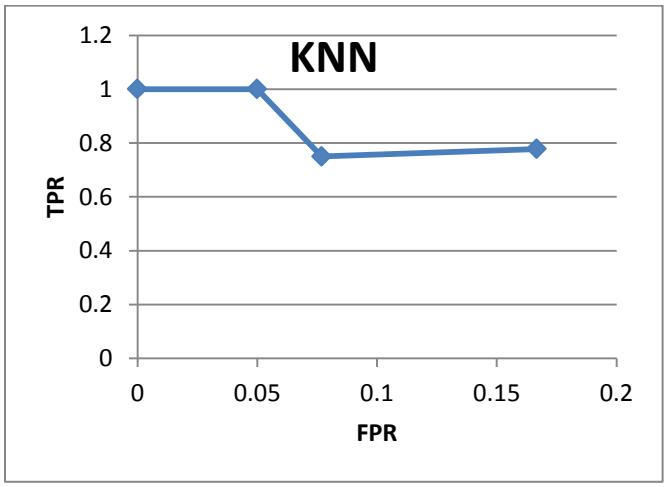


Figure 5.1: ROC Curves for Non-Semantic Classification









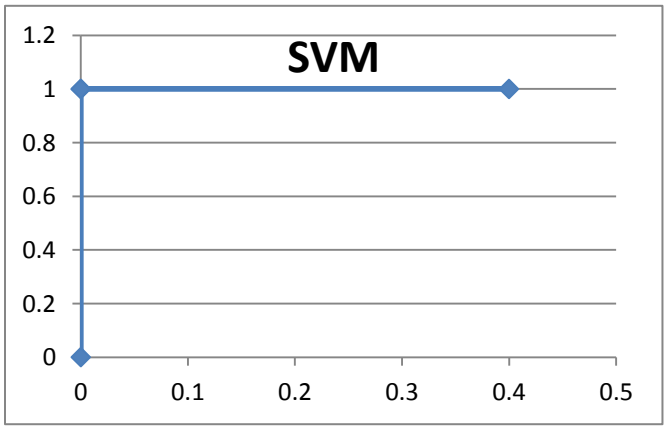


Figure 5.2: ROC Curves for Semantic Classification

Chapter Six: Conclusion

6.1 Conclusion

In this research clustering and classification models have been built and implemented on a set of web services expressed as their WSDL file. The model can be divided into two phases the first phase is clustering of WSDL which depended on the output of preprocessing algorithms including parsing, filtering, removing unnecessary words and tokenizing which process each WSDL file to get the most important words in it. Clustering of web services have been done using K-Medoids clustering algorithm while two techniques for measuring similarity have been used cosine similarity and semantic similarity.

After clustering different classification techniques have been used inorder to compare their result by measuring accuracy of each technique and its confusion matrix.

Classification based on semantic clustering showed better accuracy than simple clustering and this is due to using semantic meaning of words in clustering. But in semantic classification as the number of extracted word increase the efficiency of that technique reduced since it will take very long time in measuring the similarity matrix between documents, that reason restricted our experiment on a small set of web services.

As result showed the best classification technique for both simple clustered WSDL and for Semantic Clustered WSDL was Neural Network.

For future there is a need for working more on the number of words extracted from each WSDL document especially for semantic clustering because as the number of words extracted increased the computation for similarity matrix will take very long time. Also structural similarity between web services can be merged with semantic similarity to improve the performance of clustering and classification.

References

[1] H. Wang, J. Z. Huang, Y. Qu, and J. Xie, “Web services: problems and future directions,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no. 3, pp. 309–320, 2004.

[2] O. Hatzi, G. Batistatos, M. Nikolaidou, and D. Anagnostopoulos, “A Specialized Search Engine for Web Service Discovery,” *2012 IEEE 19th International Conference on Web Services*, 2012.

[3] Seekda GmbH, “Seekda!Web Services Search Engine,” *Seekda* . [Online]. Available: http://webservices.seekda.com. [Accessed: 16-Oct-2017].

[4] “Introduction to WebLogic Web Services,” *Programming WebLogic Web Services.* [Online]. Available: http://download.oracle.com/docs/cd/E13222_01/wls/docs81/webserv/index.html. [Accessed: 16-Oct-2017].

[5] D. Bouchiha and M. Malki, “Semantic Annotation of Web Services,” *Proceedings ICWIT*, pp. 60–69, 2012.

[6] A. A. Patil, S. A. Oundhakar, A. P. Sheth, and K. Verma, “Meteor-s web service annotation framework,” *Proceedings of the 13th conference on World Wide Web - WWW 04*, 2004.

[7] A. Heß, E. Johnston, and N. Kushmerick, “ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services,” *The Semantic Web – ISWC 2004 Lecture Notes in Computer Science*, pp. 320–334, 2004.

[8] T. G. Stavropoulos, D. Vrakas, and I. Vlahavas, “Iridescent,” *Proceedings of the 3rd International Conference on Web Intelligence*, Mining and Semantics - WIMS 13, 2013.

[9] A. Heß and N. Kushmerick, “Learning to Attach Semantic Metadata to Web Services,” *Lecture Notes in Computer Science The Semantic Web - ISWC 2003*, pp. 258–273, 2003.

[10] A. Heß and N. Kushmerick, “Automatically attaching semantic metadata to Web Services,” *Lecture Notes in Computer Science The Semantic Web - ISWC 2003*, 2003.

[11] W. Liu and W. Wong, “Web service clustering using text mining techniques,” *International Journal of Agent-Oriented Software Engineering*, vol. 3, no. 1, p. 6, 2009.

[12] W. Wong, W. Liu, and M. Bennamoun, “Tree-Traversing Ant Algorithm for term clustering based on featureless similarities,” *Data Mining and Knowledge Discovery*, vol. 15, no. 3, pp. 349–381, Aug. 2007.

[13] P. University, “What is WordNet?,” *Princeton University*, 17-Mar-2015. [Online]. Available: http://wordnet.princeton.edu/. [Accessed: 16-Oct-2017].

[14] P. R. Reddy and A. Damodaram, “Web services discovery based on semantic similarity clustering,” *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pp. 1–7, 2012,.

[15] M. Li and Y. Yang, “Efficient clustering index for semantic Web service based on user preference,” *2012 International Conference on Computer Science and Information Processing (CSIP)*,pp. 291 –294, 2012.

[16] Q. Liang, P. Li, P. C. Hung, and X. Wu, “Clustering Web Services for Automatic Categorization,” *2009 IEEE International Conference on Services Computing*, pp.380 –387, 2009.

[17] X. Zhang, Y. Yin, M. Zhang, and B. Zhang, “Web Service Community Discovery Based on Spectrum Clustering,” *2009 International Conference on Computational Intelligence and Security*, pp 187 –191, 2009.

[18] S. Dasgupta, S. Bhat, and Y. Lee, "Taxonomic clustering of web service for efficient discovery," *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM 10*, pp1617–1620, 2010.

[19]M. Crasso, A. Zunino, and M. Campo, "AWSC: An approach to Web service classification based on machine learning techniques*," Inteligencia Artificial*, vol. 12, no. 37, pp. 25-36,Dec. 2008.

[20] M. Bruno, G. Canfora, M. D. Penta, and R. Scognamiglio, "An Approach to support Web Service Classification and Annotation," *2005 IEEE International Conference on e-Technology*, e-Commerce and e-Service, pp 138-143, 2005.

[21] H. Wang, Y. Shi, X. Zhou, Q. Zhou, S. Shao, and A. Bouguettaya, "Web Service Classification Using Support Vector Machine," *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, pp 1,3-6, 2010.

[22] Q. Z. Sheng, B. Benatallah, R. Stephan, E. Oi-Yan Mak and Y. Q. Zhu , "Discovering E-Services Using UDDI in SELF-SERV," *International conference on E-Business*, Beijing , China, May 2002.

[23] L. Yuan-Jie and C. Jian, "Web Service Classification Based on Automatic Semantic Annotation and Ensemble Learning," *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, 2012.

[24] J. Yang and X. Zhou, "Semi-automatic Web Service Classification Using Machine Learning," *International Journal of u- and e-Service, Science and Technology*, vol. 8, no. 4, pp. 339–348, 2015.

[25] *XML WSDL*. [Online]. Available: http://www.w3schools.com/xml/xml_wsdl.asp. [Accessed: 16-Oct-2017].

[26] P. Berkhin, "A Survey of Clustering Data Mining Techniques," *Grouping Multidimensional Data*, pp. 25–71, 2006.

[28] O. Maimon and L. Rokach,"Clustering Methods," in *Data Mining and knowledge Discovery Handbook*, 2'nd ed, Springer Science & Business Media, ch15, 2010.

[29] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Boston: Pearson Addison Wesley, ch 8, 2005.

[30] C. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 68–86, 1971.

[31]J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Amsterdam: Elsevier/Morgan Kaufmann, 2012.

[32] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley and Sons Inc, 2001. .

[33] L. Meng1, R. Huang and J. Gu, "A Review of Semantic Similarity Measures in WordNet," *International Journal of Hybrid Information Technology*, Vol. 6, no. 1, January, 2013.

[34] T. Slimani, "Description and Evaluation of Semantic Similarity Measures Approaches*," International Journal of Computer Applications*, vol. 80, no. 10, pp. 25–33, 2013.

[35] C. C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms," *Mining Text Data*, pp. 163–222, 2012.

[36] S. sayad, "Decision tree example," *An Introduction to Data Science*. [Online]. Available: http://www.saedsayad.com/decision_tree.htm. [Accessed: 16-Oct-2017].

[37] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in induction*. New York: Acad. Pr., 1966.

[38] D. H. Moore, "Classification and regression trees, by Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Brooks/Cole Publishing, Monterey, 1984,358 pages, 5," *Cytometry*, vol. 8, no. 5, pp. 534–535, 1987.

[39] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatics 31*, Vol 31, No 3, pp. 249-268, 2007.

[40] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the fifth annual workshop on Computational learning theory - COLT 92*, 1992.

[41] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Amsterdam: Elsevier/Morgan Kaufmann, pp70-181, 2012.

[42] E. Turban, R. Sharda, and D. Delen, *Decision support and business intelligence systems*. Harlow, Essex: Pearson, 2014.

[43] "Receiver operating characteristic," *Wikipedia*, 12-Oct-2017. [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic. [Accessed: 16-Oct-2017].

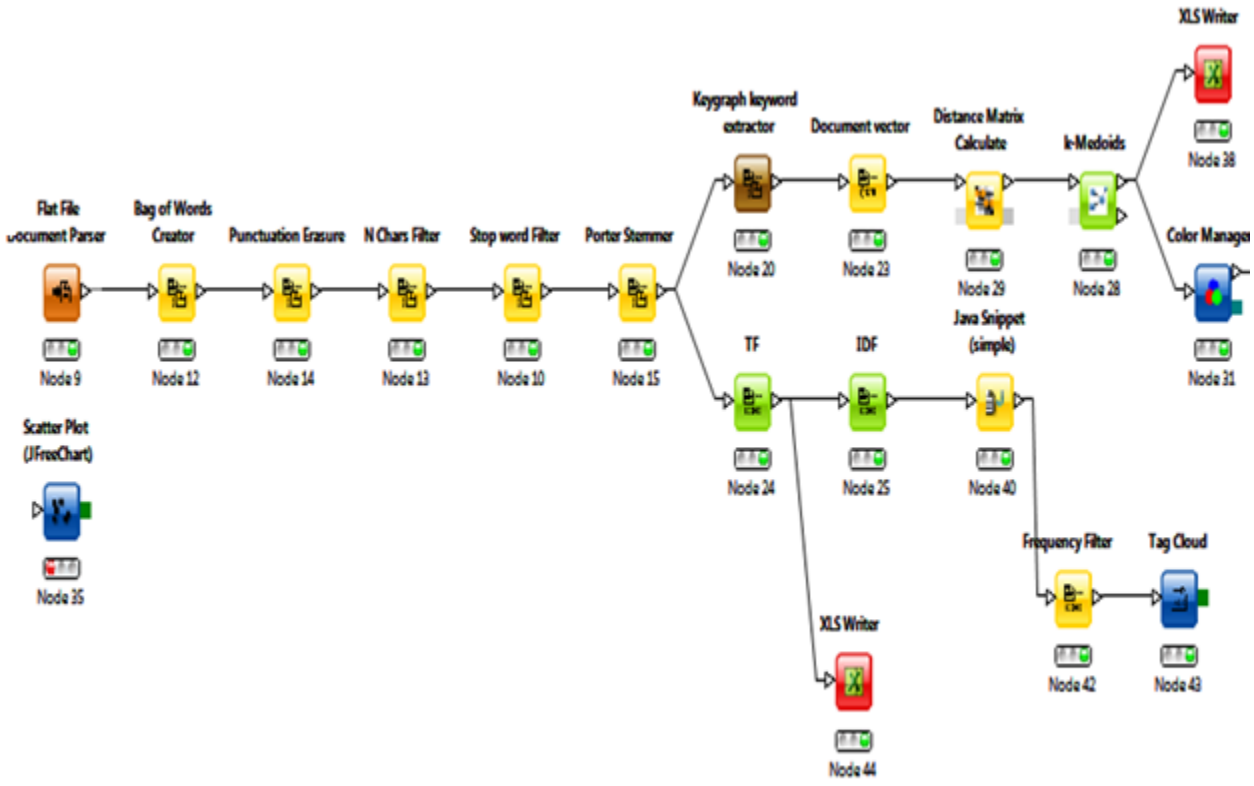
[44] "Confusion matrix," *Wikipedia*, 14-Sep-2017. [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix. [Accessed: 16-Oct-2017].

[46] C. Fox, "A stop list for general text," *ACM SIGIR Forum*, vol. 24, no. 1-2, pp. 19–21, Jan. 1989.

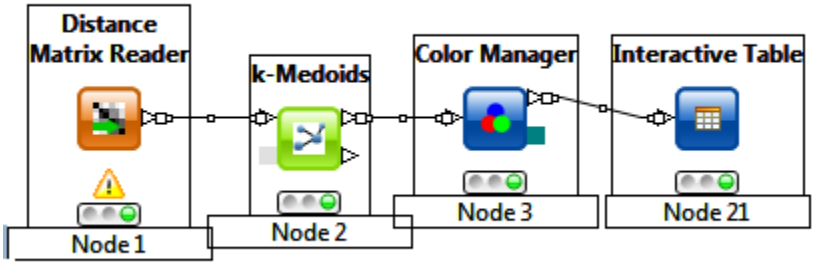
Appendices

Models :

1. Non-Semantic Clustering Model



2. Semantic Clustering Model



** Distance Matrix Reader will read semantic similariy matrix from a java code that will calculate it.

3. Classification Model