FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Computer Science	Spring semester, 2018 Open
Author: Anousheh Shenavari Shirazi	..Anousheh Shenavari Shirazi... (signature of author)
Programme coordinator: Nina Egeland Supervisor(s): Professor. Chunming Rong Cristina Viorica Heghedus	
Title of master's thesis: Machine Learning methods to detect improper and irrelevant citations	
Credits: 30	
Keywords: Machine Learning algorithm, Citations relation, Data mining, Classification, Decision Tree, Naive Bayes algorithm	Number of pages: 38…… +supplemental material/other: 53…… Stavanger,…15/06/2018……. date/year

UNIVERSITY OF STAVANGER

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

MASTER THESIS IN COMPUTER SCIENCE

Machine Learning methods to detect improper and irrelevant citations

Author:
Aousheh SHENAVARI SHIRAZI

Supervisors:
Prof. Chunming RONG
Cristina VIORICA HEGHEDUS

June 2018

Abstract

The focus of this study is on the relation between papers and their citations using Machine Learning algorithms to detect improper and irrelevant citations. The model takes the paper's citations and classifies them into two classes, "Related" and "Barely related" citations. Here we considered two Machine Learning algorithms, "Decision tree algorithm" and "Naive Bayes algorithm" along with introducing the statistical algorithm called "Prior statistical algorithm" to classify the relation.

During the design process of the classification models, the required data for implementing have been collected from a large-scale and reliable data source. Converting techniques have been used to transform data to the structured format.

The evaluation results show that the Prior statistical model has limitation since it applied on dataset considering only one feature, however from the two machine learning algorithms that we employed, Naive Bayes outperform decision tree since it was extremely fast and did not require a very large training set to obtain a good learning model, however Decision Tree was easier to implement and understand.

Preface

This work is carried out as a master thesis in computer science at the University of Stavanger during the Spring semester of 2018. The idea of this study is one of the interesting topics in Machine Learning technologies and classifications which is suggested by the supervisor of the project. The objective is to build a classification model that can be applied to express the relation between the papers and their citations appropriately.

Stavanger, Spring 2018

Acknowledgements

First of all, I would like to express my sincere gratitude to my thesis advisor and supervisor; Professor Chunming Rong for his great and insightful comments, patience, motivation, immense knowledge and continuous support through to the whole process of this master thesis. Besides my first supervisor, I would like to thank my second supervisor, Cristina Viorica Heghedus, for her attention, valuable comments and encouragement, but also having a weekly meeting with some fellow master and PhD students which encouraged me to widen my study from various perspectives

Last but not the least, I would like to thank my family, especially my husband, for continuous encouragement and supporting me spiritually throughout researching and writing this thesis.

Table of Contents

List of Tables

List of Figures

1 Introduction

This chapter includes three sections. The first section introduces the problem statement and general overview of the thesis's topic. Research background and motivation considered in Section 1.2 and in Section 1.3 the thesis outline will be represented.

1.1 Thesis Overview

There are various research papers published in conferences and journals which can be easily accessed through the internet. In a dataset of scientific articles, each document can be considered to hold both the content of the document itself and its citations to the other documents.

Authors are mainly using the relevant information and important previous work as references to having a high-quality research paper, whether they are published on paper, presented in a lecture or broadcast online. There are different search engines were developed such as Google Scholar or online digital library websites such as Scopus, IEEE Xplore, Elsevier, etc. to retrieve relevant academic papers.

The STM Report, whose the leading global trade association for academic and professional publishers reveals there were about 28,100 active scholarly peer-reviewed English-language journals in 2014, collectively publishing approaching 2.5 million articles a year, the figure **Fig. 1.1** shows the growth in the number of active, peer-reviewed journals recorded in Ulrich's directory between 2002 and 2012; over this period the number grew by about 2.5% a year [13].

A large number of scientific papers around the world make their citation relation a very interesting and highly important curriculum.

The scientific documents have a different form of structure than the non-scientific document, for instance a research paper at least contains attributes like authors, abstract, con-

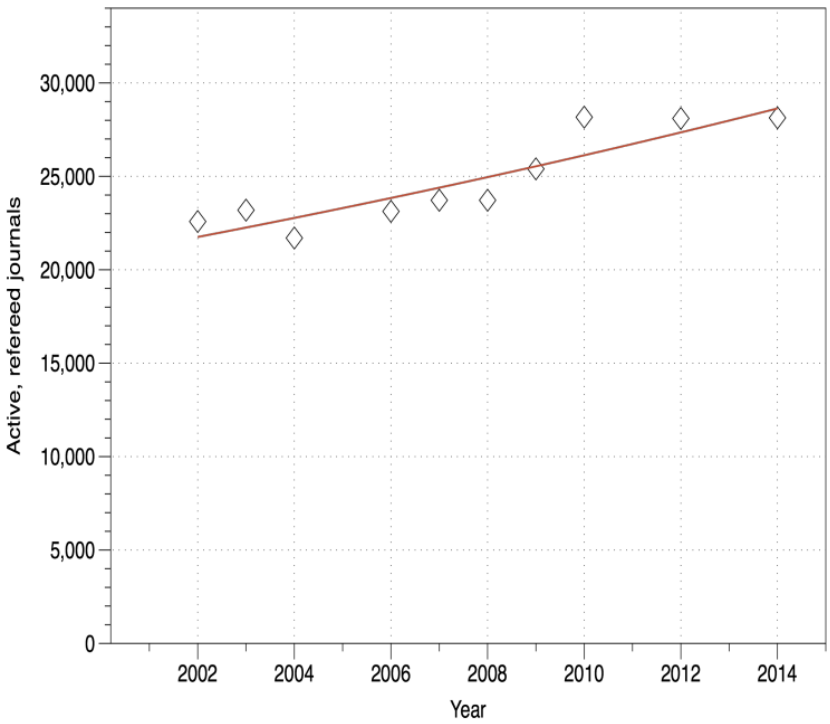

Figure 1.1: Growth in the number of active, peer-reviewed journals recorded in Ulrich's directory, 20022012 [13]

text, and references, however, some attributes such as specific information regarding the citations like their frequency, how many times they have been called by the author during the paper, and their position, which section of the document contains specific citation, are not included in research papers explicitly.

This extra information related to the citation help to recognize how strongly or weakly the paper related to its citations. Since the paper's citations can be any particular resources like papers as well, many methods and algorithms have been developed in understanding and constructing paper's relation. Some basic approaches are bibliographic coupling, co-citation analysis, cited by and reference list [5].

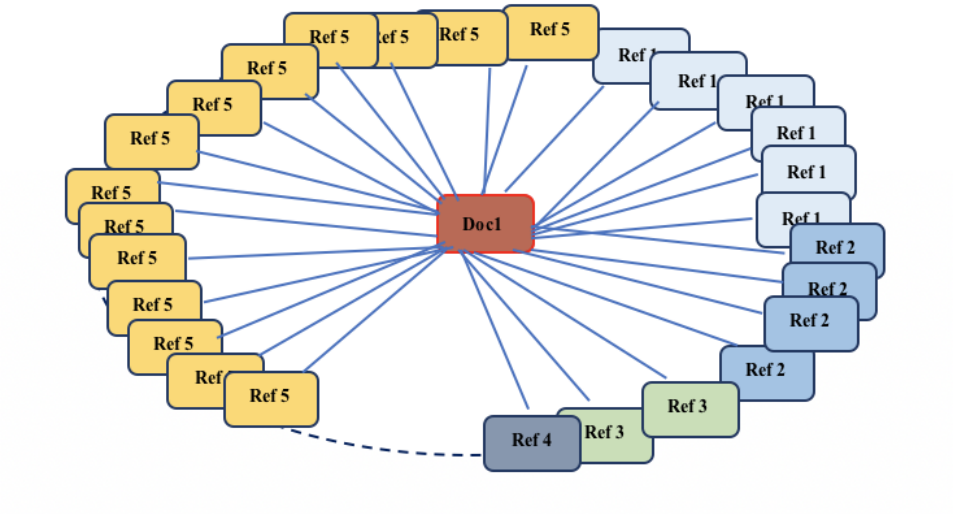

Figure 1.2: Document and citations relation

To be clearer regarding the paper and its citations, one example illustrated in **Fig. 1.2**. Any academic document like "Doc1" uses some number of resources and as the figure shows reference "Ref5" has been cited more than the others and that may conclude that citation "Ref5" is more related to the topic than the other references. To express the citation's relation first, we need to use data mining techniques to extract the citations and then for defining the relation, we can observe which section in the paper they have been used and how many times they have been cited. The more they have been used in the articles the more relevant they are to the topic.

Authors mostly use the more relevant citations in the part of paper's structure where the main core of topic argument located rather than the introduction section in the beginning or related work sections.

In data analysis, we can find relations among data objects and in this thesis, we will emphasize two approaches to perform data analysis by applying classification methods, the first algorithm targets the statistical measurements between citations' frequency feature and then performs classification to figure out how closely each reference related to the paper. The frequency of calling a citation and its position in the research depends on its

novelty contribution in a research.

Second classification method will be Machine Learning techniques considering Decision Tree algorithm and Naive Bayes classification to identify a model that best fits the relationship between the attribute set and class label of the input data. In general, the model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records.

The figure **Fig. 1.3** illustrates the citation dataset graph where the nodes correspond to documents and links refer to the relation in a way that there is a link from node ”Doc1” to node ”Doc2”, if the paper corresponding to node ”Doc1” cites the paper corresponding to node ”Doc2” or vice versa.

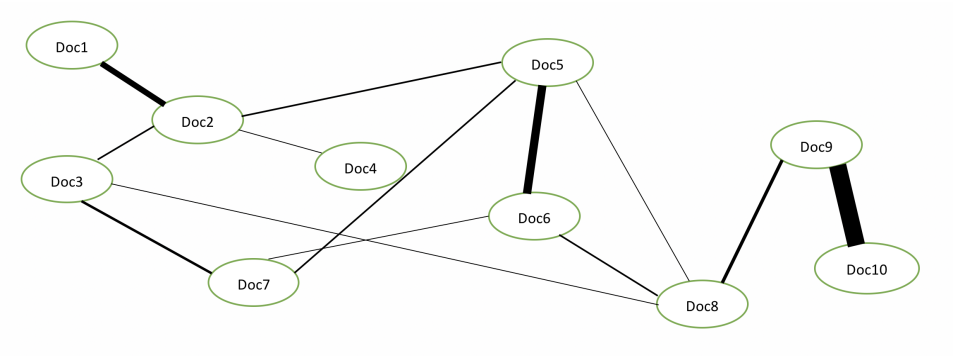

Figure 1.3: Citation dataset graph

As it is explicit from the graph some of the links are thicker than the others, for instance, the link between nodes ”Doc9” and ”Doc10”, and this means that the relation between these two documents are stronger in a way that they might be more relevant, in this relation the citation one, has been used more often in the main discussion of the document whose cited it. One approach to obtaining how strongly two documents are related is to give weight for each part of the paper and for any time calling the reference the weight value related to that part using the reference will be considered. Accordingly, the link between nodes ”Doc9” and ”Doc10” has a higher weight which can be concluded that they are more related.

This project aims to describe the relationship between an academic paper and its citations in other words how relevant citations get used during an academic paper. Three algorithms will be applied, the first algorithm focuses on the statistical measurements between citations' frequency and by obtaining comparisons we will able to realize how closely each reference related to the paper. The frequency of calling a citation and its position in the research depends on its novelty contribution in a research, for that reason, in the second approach a Decision Tree induction will be applied by using both features, “frequency” and “position”, it is a non-parametric approach for building classification models and does not require any prior assumptions regarding the type of probability distribution. The third algorithm is Naive Bayes classifications to classify a test record, it computes the posterior probability for each class, in both Machine Learning algorithms we considered continuous attribute which consequently has been used to estimate the class-conditional probability

for Naive Bayes classification, furthermore, we assumed Gaussian normal distribution for learning the model.

1.2 Research background and motivation

In this section, we review some related works on citation relations and we highlight the main purpose of this study at the end.

Mostly, the research paper is conducted to determine problems or to find answers to uncertainties and the first step in writing a paper is to find articles of interest for researchers, consequently, author needs to find related documents to have the quality assurance and cite them during the context as citations.

Scientific article recommendations have been assisting at this matter. One novel article recommendation, called Citation-based scientific Article Recommendation (CAR) [7] proposed a method which combines the information of researchers' historical preferences and citation relations between articles.

In general, each document in a dataset of academic articles represents both the context of the document itself and its citations to the other documents. One model considers both document citation relations and the content of the document itself via a probabilistic generative structure [16] which exploits a two-level topic model that includes the citation information about "father" topics and text information for subtopics.

Research papers are frequently get cited in the related documents where recommendation techniques will be conducted. The recommendation models are mainly based on similarity among documents to do recommendation of related ones. A new recommendation technique proposed [15] based on reference graph of documents, combined with traditional recommendation techniques and this method can recommend research paper more accurate and effective.

The next step after exploring the paper's relation, the results can build the relation network. This relation network can be visually represented as a graph where a node illustrates a paper and link shows the relation to its references. This relation network is used in developing a research map application [9].

Another way to figure out the relations among documents will be applying frequent item mining for instance, n-gram, stemming, stop word removal and etc. One work has been [10] proposed as an order accumulative citation matrix, which is formulated from the citation information in the publications. This method presents a new validity measure which represents the validity of discovered relations regarding the proposed evaluation criteria. Natural Language Processing typically has been used to discover the similar documents as well, one system [1] has been introduced to retrieve publications from Google Scholar and visualize them as a 2D graph using the citation relation, where the nodes represent the documents while the links represent the citation or reference relation between them, the similarity score between each pair of papers measures based on both the number of paths and the length of each path. More paths and shorter lengths hold higher similarity score.

Citation count ranking has been a useful measure regarding the relevancy of documents and evaluates the quality, [8] one novel that focuses on the relevant articles conducted this

method to define the relevancy. This method proposed the scoring of citing article, only when the cited article is appropriate.

Our concern in this paper will be mainly on the relation between every paper and its citations by classifying the references and we are interested to see if any classified citation has no relation to the paper cited that citation.

1.3 Thesis outline

The rest of this thesis is structured as follows. In Chapter 2, we review the dataset preparation process and all the related concepts including, data mining process, the primary dataset and experimental dataset and challenges.

In Chapter 3, we describe the development process of the Machine Learning classifications by first, considering classification introduction and feature extraction, following by the methods have been applied in our case study, such as prior algorithm, Decision Tree algorithm and Naive Bayes algorithm.

Chapter 4 consists of the evaluation of outputs regarding the implemented of previous Chapter methods. Finally, in Chapter 6, we summarize our achievements and conclusions, outline the limitations and make improvement suggestions for future studies.

2

Data collection and challenges

In the following Sections, we will be considering data mining introduction and general concepts of data preparation process at first in Section 2.1 and then we will be presenting the Primary dataset which we started to work within Section 2.2 and lastly in Section 2.3 the Experimental dataset which has been used in the classification models. We will highlighted the challenges that we encounter, in the related section along with the solution strategy.

The first issue was working with the dataset that did not have the proper text for documents, it will be explained more in the first dataset section. Since papers are mostly in PDF format I needed to extract the text format from PDF format. The first option to use was PyPDF2 package which worked with Python3, however did not have a way to extract images, charts, or other media from PDF documents, but it could extract text and returns it as a Python string, however it failed to return the whole content of papers. We tried PDFMINER library as well, but still failed as before.

2.1 Data Mining process

Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a future observation [11].

Data mining tasks are typically divided into two main categories, predictive tasks, where the main task is to predict the target value according to the other attribute measures, and descriptive tasks, which obtain the relation or correlation between attributes in dataset. Our concentration will be in descriptive data mining tasks to explore any relations regarding the paper and its citations.

The life cycle of a data mining project shows in **Fig. 2.1** [14], before applying the data mining techniques and implementations we need to define the problem, as we did earlier in the Introduction Chapter and decide what kind of data we will need to be collected for ”Modeling” step in the cycle, thereforeee ”Data understanding” step has a very important role for starting the process. In this project, we had to collect the new data since the quality of the Prior dataset was not appropriate as it will be explained in next Section.

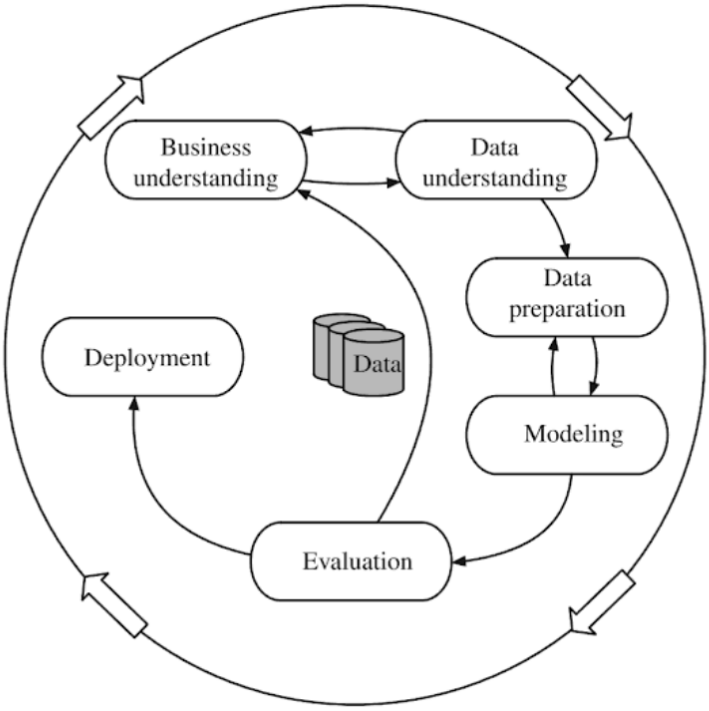

Figure 2.1: Life cycle of a data mining process [14]

In ”Data preparation” process, the raw data will be converted into useful information for Machine Learning techniques to generate the model and is suitable for further data analysis, which includes some transformation such as preprocessing steps.

Data preprocessing involves removing information that not required, such as fusing data from sources, cleaning data, tokenizing, Remove duplication, common and stop words, stemming and selecting features that are relevant, in our case we can remove the records which do not contain the references.

Performing data postprocessing ensures that only useful results are incorporating into the outline decision. This procedure usually includes pruning routines, filtering, combination, or knowledge integration. All these procedures provide a filter for unreliable knowledge derived by an inductive algorithm.

2.2 Preprocessing the dataset

Data Preparation is an important part of data mining process. Data preprocessing is a technique where the raw data, which is not feasible, converts into a clean dataset, **Fig. 2.1** illustrates steps in the preparation flow. As we will explain the dataset, considering new one and challenges in the next two section thoroughly, we realized the best format according to the need of the model implemented in Python programming language in our experimental dataset was XML or HTML formats.

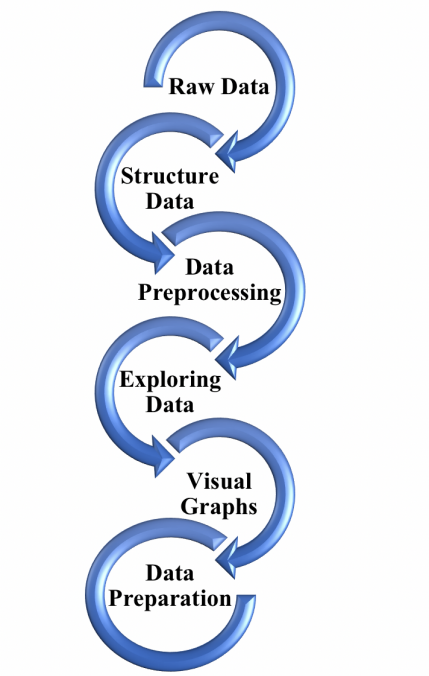

Figure 2.2: Preprocessing Overview

In preprocessing step, we spotted the papers whose using the compact format for the number of citations, for instance, if author cites three references for a section like, ”[3,4,9]” or ”[1014]”, where they have been contributing to the same part commonly, the model could only track the first citation, in our example 3 or 10, and the rest of the citations, 4, 9 and 14 in our example, was eliminated, therefore in preprocessing performance we ignored those combine format as all those references hold same frequency value.

The primary focus in cleaning the data was on handling noisy data such as the words like, ”figure”, ”TABLE”, ”table”, ”fig.” and ”FIGURE”, were detected and consequently removed, even the presence of tag label such as, ” <region >”, was not appropriate to achieve the proper result and had to be deleted as well.

2.3 Primary dataset

For the beginning, we have considered a citation network dataset by the size of 374,6 MB. It was contained 629,814 papers and 632,752 citations. Each record or paper was associated with some attributes, such as ”ID”, ”title”, ”authors”, ”venue”, ”year”, ”references' ID” and ”abstract”. Every paper or record in the file has been separated by the blank line. To better understanding, the table below as **Table. 2.1** illustrates few papers with their attributes from this dataset.

ID	Titles	Authors	Venue	Year	references' ID	Abstract
1	A+ Certification Core Hardware (Text and Lab Manual)	Charles J. Brooks		2003		
24	On product covering in 3-tier supply chain models: natural complete problems for W[3] and W[4]	Jianer Chen,Fenghui Zhang	Theoretical Computer Science	2006	251778 436906 623227 287885	The field of supply chain management has been growing at a rapid pace in recent years ...
35	An Integrative Modelling Approach for Simulation and Analysis of Adaptive Agents	Tibor Bosse,Catholijn M. Jonker,Jan Treur	Proceedings of the 39th annual Symposium on Simulation	2006	247215 618899	To simulate adaptive agents with abilities matching ...

Table 2.1: Sample of three records with features of Prior dataset

The metadata format of the file was in the text format and special characters have corresponded to each feature by default as it is shown in the **Table. 2.2**. For the preprocessing step we have removed the records that did not include any references and after exploring the dataset we considered that a pair of two references' ID and paper's ID features are important for solving our classification problem.

Characters in the text file	Description
#*	Paper Title
#@	Authors
#t	Year
#index 00	index id of this paper
#%	Id references of the paper
#c	publication venue
#!	Abstract

Table 2.2: Characters corresponds to each feature in Primary dataset

Although, The relation between papers and their citation cannot be classified in this particular dataset as short sample part of the dataset shown in **Fig. 2.4**, subsequently for the following reasons we decided to collect new dataset which will be represented as "Experimental dataset and challenges" in the next Section;

- The first important issue was the actual content of the papers which was not provided in the dataset, we needed to access the context for detecting the features related to the citations and applying the model.
- Another issue regarding this dataset was the citation's title, the only feature with whole text was "Abstract" and the feature related to the citations which was "references' ID" had no expression. A sample of records from dataset and feature extracted from one paper has illustrated in **Fig. 2.3** and **Fig. 2.4** respectively, and the lines start with "#%" belongs to reference's ID, where does not include any title nor any explanation regarding those IDs.

For the mentioned issues we decided to consider new dataset of papers as we get the chance to access the proper text of papers for feature extraction.

```
#*HandbookofImageandVideoProcessing(Communications,NetworkingandMultimedia)
#*StarWarsBattlefrontII(PrimaOfficialGameGuide)
#*Protectedsystemsingeneralsystemstheory
#*UsingXml:AHow-to-do-itManualforLibrarians
#*AppleComputerResourcesinSpecialEducationandRehabilitation
#*IntroductiontoComputationalScience:ModelingandSimulationfortheSciences
#*PCsinEasySteps(InEasySteps)
#*TheCompleteGuidetoFlex2withActionScript3.0
#*AdvancedVisualBasic.NET:ProgrammingWebandDesktopApplicationsinADO.NETandASP.NET
#*PerformanceEvaluationofPacketProcessingArchitecturesUsingMulticlassQueuingNetworks
#*Authenticatingmobilephoneusersusingkeystrokeanalysis
```

Figure 2.3: Sample list of papers in the primary dataset

2.4 Experimental dataset and challenges

The new dataset includes 14 papers in a collection from the smart company conference and the size is 14,8 MB, as the papers are in the PDF format, we required to convert the

```
#*Multimedia Directory 1997
#@
#t1997
#c
#index22

#*ASIS&T Thesaurus of Information Science, Technology, And Librarianship (Asist Monograph Series)
#@Alice Redmond-neal,Marjorie M. K. Hlava
#t2005
#c
#index23

#*On product covering in 3-tier supply chain models: natural complete problems for W[3] and W[4]
#@Jianer Chen,Fenghui Zhang
#t2006
#cTheoretical Computer Science
#index24
#%251778
#%436906
#%623227
#%287885
#!The field of supply chain management has been growing at a rapid pace in recent years, both as a research area and as a
practical discipline. In this paper, we study the computational complexity of product covering problems in 3-tier supply
chain models, and present natural complete problems for the classes W[3] and W[4] in parameterized complexity theory. This
seems the first group of natural complete problems for higher levels in the parameterized intractability hierarchy (i.e., the
W-hierarchy), and the first precise complexity characterizations of certain optimization problems in the research of supply
chain management. Our results also derive strong computational lower bounds and inapproximability for these optimization
problems.
```

Figure 2.4: Sample of three papers with their attributes from the primary dataset

PDF to TEXT format to make the dataset suitable for models.

We have considered different converters such as “XPDF” and “PDFTOTEXT”, however they break the text and ruin the format consequently it was impossible to locate all of the citations correctly. Another option was using toolkits like, ”Foxit” PDF toolkit, but registration was needed and the tools were not accessible for free. thereforee I have considered Python modules such as ”PDFMINER”, although it did not seem to perform well for Python 3. Instead we have installed ”PyPDF2” module in order to be able to read the PDFs in Python3, yet the problem raised for extracting images, charts and tables. The fact is the module can only extract text and return it as a Python string in an unstructured format.

To overcome the earlier issues, We have determined to use the HTML format instead of TEXT which is more well-structured for data mining methods. Accordingly, I have considered the PDF converters available online, although these converters have two problems, first, it is timeconsuming process, as it takes a while for only one PDF file with 359 KB size and second is the format, the text is full of different unreadable kinds of characters.

Then, we explore one of the open source PDF to HTML converter, ”pdf2htmlEX”, which for downloading we needed docker container. After installation, we have realized different tag labels in the text make it unable to extract special feature such as “reference ID”.

At the end to the best option to overcome the challenges of conversion to achieve suitable format of the dataset for feeding the models was ”PDFX” v1.9. system, which is a fullyautomated PDF to XML converter for scientific articles, it is associated with easy ways to use and access as well.

The system takes a full-text PDF article as input and outputs an XML document. The key aspect of the presented approach is that the rule set relies on relative parameters derived from font and layout specifics of each article, rather than on a template-matching paradigm. The system thus obviates the need for domain- or layout-specific tuning or prior training, exploiting only typographical conventions inherent in scientific literature

[3]. The transforming process took 50 minutes to convert 14 papers into XML format with the size of 14,8 MB and the converted file in XML format will be saved in the same path directory as the source of PDF files.

At this point of our project, we gained the proper format of dataset and we are able to start data analysis task and developing the classification models.

2.4.1 Data exploration

Data exploration is very beneficial performance, in selecting the proper preprocessing methods and in data analysis techniques.

As we discussed it so far, A list of papers from the smart company conference has been selected and transformed in XML format, whose much like HTML format, The difference between HTML and XML formats expressed as they were designed for different target, like XML was designed to carry the data without concerning how data looks, on the other hand HTML was designed to display data.

XML stands for Extensible Markup Language, Each XML document has both a logical and a physical structure. Physically, the document is composed of units called entities. An entity may refer to other entities to cause their inclusion in the document. A document begins in a ”root” or document entity. Logically, the document is composed of declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup and the logical and physical structures must nest properly [2].

The content of each transformed file contains different tag labels, whose are the structure's elements surrounded by angle brackets. These XML tags normally come in pairs like <region>and </region>and different tag labels in the XML file shows different part of the paper such as <article-title>, <abstract>, <body>, <xref>and so on. A part of the file has been illustrated as bellow in **Fig. 2.5**.

We realized by detecting two tag labels like, <xref>and <article-title>, we will be able to extract features, as citation's frequency feature for each particular paper.The other tag labels such as <region>, <abstract>and <body>will be selected for the second feature ”citation's position” where emphasizes which section of the paper the particular citation located.

```
  <abstract class="DoCO:Abstract" id="6">—Ambient Noise Seismic Imaging (ANSI) is a recently developed
      geophysical methodology to image the shallow subsurface structures of earth using ambient/environment
      noise as the source. Integrating ANSI computing within distributed sensor networks will enable real-
      time continuous monitoring of subsurface dynamics for sustainability studies. However, the research
      challenges associated with this innovative approach are significant. Traditional data collection using
      sensor networks imposes practical difficulty for real-time applications, because of the sheer amount
      of data and large-dense sensor arrays versus limited network bandwidth. This paper is the first to
      investigate how to utilize the computing capabilities of sensor nodes to perform the computation of
      ANSI under resource constraints. We explored two distributed approaches (aggregation and consensus)
      for computing ambient noise eikonal tomography and obtaining phase velocity maps. We performed
      experiments using CORE emulator to obtain phase velocity maps on real data from USArray Transportable
      Array. Results demonstrate that our approaches can illuminate phase velocities under network
      constraints. We also show that the proposed aggregation and consensus algorithms not only balance the
      computation load but also achieve low communication cost and high data loss tolerance. Index Terms —
      subsurface imaging, ambient noise, eikonal tomography, sensor networks, distributed computation.</
      abstract>
</front>
<body class="DoCO:BodyMatter">
  <region class="DoCO:TextChunk" id="34" page="1" column="1">I. I NTRODUCTION Subsurface Imaging is a
      technique widely used in geophysical exploration for investigating structures under the surface of
      earth. Understanding and addressing the subsurface sustainability has a significant impact on natural,
      social, and economic issues across the globe. Real-time imaging of shallow underground structures is
      essential to assess the sustainability and potential hazards of geological structures [ <xref ref-
      type="bibr" rid="R1" id="7" class="deo:Reference">1</xref>]. The recent landslide tragedy in
      Washington state 1 is yet another example showing that understanding the subsurface sustainability is
      crucial to public safety. To fully utilize the dense seismic array when there are few earthquakes or
      other obvious events, a new approach called Ambient Noise Seismic Imaging (ANSI) [2]—[<xref ref-
```

Figure 2.5: A part of output result in XML format

3

Development of the Classification algorithms

In this chapter, we describe the development process of Classification with focus on two approaches as ”Statistical Approach” and ”Machine Learning Approach” mentioned in chapter 1, which will be considered in Section 3.1 and 3.2 accordingly; We obtain the statistical relations for citations and design a model and introduce the “Prior Algorithm” in Section 3.1.1 and its implementations where we classify the citations will be described in Sections 3.1.2. Then, in Section 3.2.1 and 3.2.2 the Machine Learning Algorithms as “Decision Tree Algorithm” and ”Naive Bayes Algorithm ” will be defined where we explain how the modules work. Lastly, implementations of last two Machine Learning algorithms for the citation’s relations is expounded in Section 3.2.3 and 3.2.4.

3.1 Classification Introduction

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The model is derived based on the analysis of a set of training data (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the label is unknown. [6].

The classification techniques such as Decision Tree classifiers, Support Vector Machines, Naive Bayes classifiers and K Nearest Neighbors apply for building the classification models, here we picked two methods of them along with introducing a new algorithm called Prior algorithm to solve the problem. Each technique applies a learning algorithm to determine a model. This model defines the relationship between the attribute set and class label of the dataset. The training dataset consists of records whose class labels are known, the figure **Fig. 3.1** shows the classification task as mapping input, attribute set,

into the output as class labels.

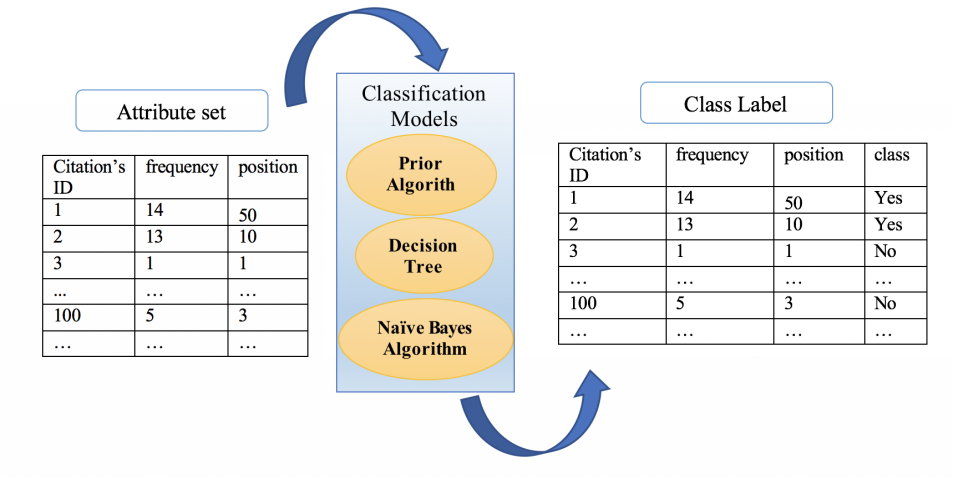

Figure 3.1: Classification task as assigning attribute set to the class labels

The table is known as ”Attribute set” in figure **Fig. 3.1** depicted some part of the training set whose will be used to generate the classification model and this model will be applied into the test set with unknown class labels.

3.2 Feature Extraction

It is frequently possible to create, from the original attributes, a new set of attributes that capture the important information in a dataset much more effectively. The creation of a new set of features from the original raw data is known as feature extraction [11].

When writing scientific papers, authors take in considerations the degree of usefulness of their references. Thus, they place the references in different sections, such as methodology, introduction and etc. based on their importance and relevancy. As a result, we decided to use the following features ”Frequency”, in the sense that how often the citation has been cited during the paper and ”Position”, which represent where in the paper author called citations.

We will only consider ”Frequency” feature in Statistical approach since we will do statistical computations regarding its values further in Prior Algorithm and it does not make scenes to do the same performance for ”Position” feature.

Implying to the position’s features, they will be assigned a higher weight if they are located in more important sections means the features are more relevant, while less important features are given a lower weight. These weights are initialized base on the domain knowledge about the paper and its concept dependency to the feature position as it is presented in the table **Table. 3.1**.

As we have the access to the converted XML format of papers, in the implementations we were looking for ”xref” tag label to pinpoint the references and to extract the position

feature we needed to eliminate all nodes' tags from XML file, since the tag labels did not allow us to identify the reference's position. Consequently, we have converted the XML file to the plain text file as it represented below in **Fig. 3.2**.

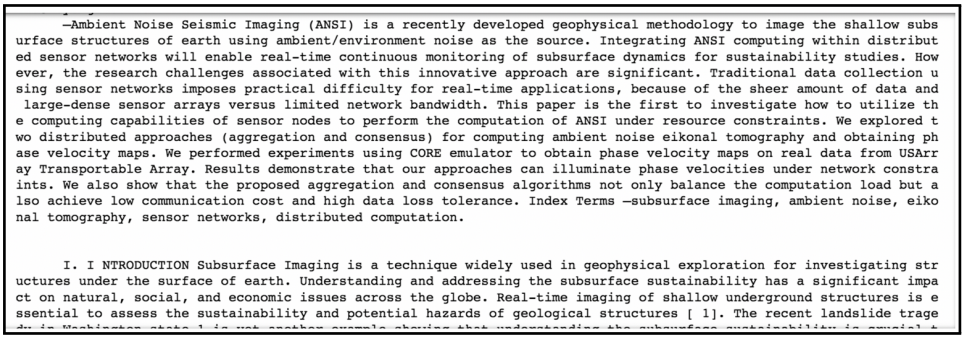

Figure 3.2: Sample of converted text format from XML format

The object values of the reference's frequency would be a set like $\{ x_1,... x_m\}$, where x value is the repetition for m references of each paper. In the transformed "TEXT" format we were observing the keywords in the context such as "Introduction", "Related work", "Abstract", "Conclusion" and the rest which considered as "body" and all of these different Sections held different values related to the position of citation as it shows below in **Table. 3.1**.

	Abstract	**Introduction(int)**	**Body(bod)**	**Related_work(rel)**
weight	0	1	5	1

Table 3.1: Table of initialized weight value for position feature

3.3 Statistical Approach

We obtain classification process with the help of statistical measurements. The "frequency" feature will be extracted for each citation of every paper, then we calculate statistical measurements such as Mean, Median and Variance values of the frequencies of all citations related to each paper. We will apply Prior Algorithm as will be explaining in the following Section to define citation's influence over the papers. **Fig. 3.3** illustrated for better understanding of the process.

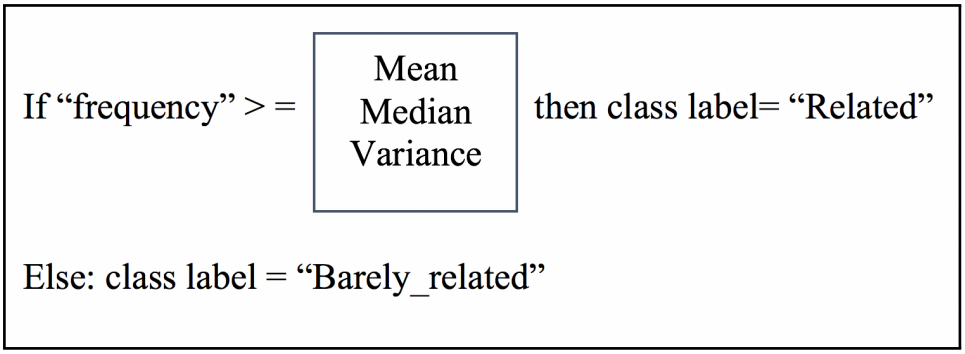

Figure 3.3: Some part of output result in XML format

To compute the Mean, Median and Variance values the following formulas assisted us. The equation (3.8) used to calculate the Mean value where N stands for the total number of citations and x is citations' frequency for each paper.

$$\bar{x} = \frac{\sum x}{N} \tag{3.1}$$

If the total number of paper's citations is an odd number, then the formula for Median value shows as a formula in (3.2) and if it is even number the equation will be (3.3). The formula for calculating the Variance value presented in (3.4).

$$Median = (\frac{n+1}{n})^{th} term \tag{3.2}$$

$$Median = \frac{(\frac{n}{2})^{th} term + (\frac{n}{2}+1)^{th} term}{2} \tag{3.3}$$

$$S^2 = \sqrt{\frac{\sum (x-\bar{x})^2}{N}} \tag{3.4}$$

3.3.1 Prior Algorithm

In this Section we will introduce Prior algorithm where the dataset is scanned to count the number of citations for each paper. The references will be extracted along with their frequency and a list containing references' ID, references' frequency, references' frequencies Mean value, article's title, references' frequencies Median and Variance values. The figure **Fig. 3.4** illustrates a part of the feature extraction. The first item in this list is a dictionary of key and value, representing the number of citations for a particular paper follow by their frequency, the second decimal number is a Mean value which has been calculated regarded

[(Counter({'1': 11, '2': 2, '3': 1, '4': 1, '5': 1, '6': 1, '7': 1, '8': 1, '9': 1, '10': 1}), 2.1, ['Transferring a n Interactive Display Service to the Virtual Reality'], 1.0, 9.877777777777773), (Counter({'9': 15, '7': 3, '16': 3, '14': 3, '8': 2, '6': 2, '13': 2, '15': 2, '1': 1, '2': 1, '3': 1, '10': 1, '11': 1, '4': 1, '5': 1, '12': 1}), 2.5, ['Multi-Client Searchable Encryption over Distributed Key-Value Stores'], 1.5, 11.733333333333333), (Counter({'23': 3, '3': 1, '9': 1, '15': 1, '16': 1, '17': 1, '18': 1, '19': 1, '20': 1, '21': 1, '22': 1, '24': 1, '25': 1, '26': 1}), 1.1428571428571428, ['Ubiquitous Tracking Using Motion and Location Sensor With Application to Smartphone'], 1.0, 0.28571428571428564), (Counter({'4': 4, '1': 3, '11': 3, '2': 2, '3': 2, '18': 2, '22': 2, '23': 2, '10': 1, '12': 1, '13': 1, '14': 1, '17': 1, '19': 1, '20': 1, '21': 1, '5': 1, '6': 1}), 1.6666666666666667, ['Hierarchical Demand Response for Colocation Data Centers'], 1.0, 0.8235294117647057), (Counter({'2': 2, '4': 2, '6': 2, '11': 2, '15': 1, '16': 1, '8': 1, '1': 1, '3': 1, '17': 1, '19': 1, '18': 1, '5': 1, '27': 1, '31': 1, '22': 1, '24': 1, '23': 1, '9': 1, '30': 1, '29': 1, '21': 1, '10': 1, '12': 1, '32': 1, '28': 1, '20': 1, '26': 1, '25': 1, '7': 1, '13': 1, '14': 1}), 1.125, ['SALM: Smartphone-based Identity Authentication Using Lip Motion Characteristics'], 1.0, 0.11290322580645161)]

Figure 3.4: The output list includes frequency of each reference, mean value of frequencies, paper's title, median and variance

to the frequencies, the third item is the paper's title, the fourth and fifth items are Median and Variance values respectively, computed regarding the citations' frequency.

In Section 3.2 we considered the feature extraction, how we were able to detect the references and observing their frequency. After generating the list with elements mentioned above we do comparisons between the Mean value and the references' frequencies for each paper. If the frequency is greater or equal than the Mean value, then it will assign the reference ID to the "related" class along with the article's title otherwise it will assign to the "barely_related" class as illustrated in **Fig. 3.5**.

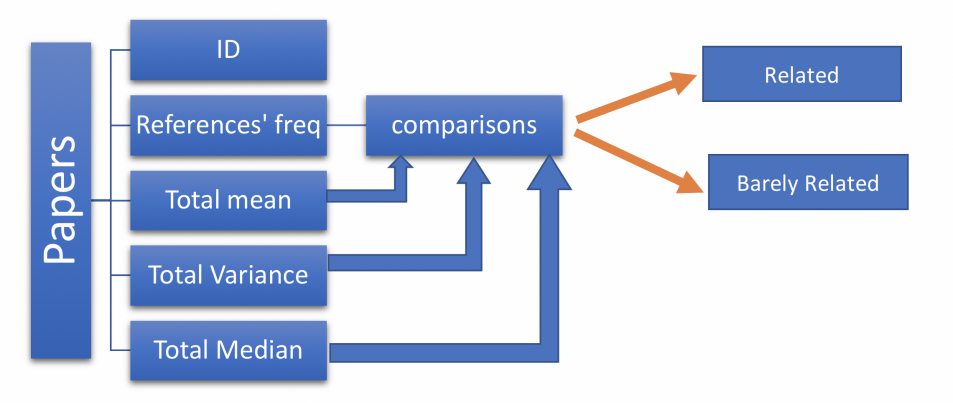

Figure 3.5: Citation graph

Although Mean value is sensitive to the presence of outliers, therefore we will apply the algorithm according to the Median and Variance values as well. If the distribution of values is skewed, then the Median provides a more robust estimate of the middle of a set of values also Mean is sensitive to the presence of outliers, for instance, the paper number 24 has ten citations and its first and second references have been called 14 and 3 times respectively, but the rest of citations called only once, therefore to overcome the traditional definition of Mean, we have proposed using the Median and Variance values.

3.4 Machine Learning Algorithm

Machine Learning is a fast-growing discipline whose investigates how computers can learn (or improve their performance) based on data. The main research area is for computer programs to automatically learn to recognize complex patterns and make the intelligent decision based on data [6].

The techniques typically use statistical concepts to learn the model from the dataset, as it identifies new patterns in data and enables data scientists to effectively identify useful information hidden in huge datasets. There are a dozen of Machine Learning Algorithms available and selecting the right algorithm is a key part of any Machine Learning project.

In general, there are three types of Machine Learning algorithms first, Supervised Learning, where the algorithm consists of a target variable which is to be predicted from a given set of predictor variables. This algorithm generates a function that map inputs to the desired outputs such as Regression, Decision Tree, Random Forest and etc.

Second Unsupervised Learning algorithm, where we do not have any target or outcome variable to predict, mainly used in clustering population and portioning a group of the dataset into different classes, for instance, Apriori algorithm, K-means and etc.

Third, Reinforcement Learning algorithm, whose the machine trains itself constantly using trial and error procedures in a way that machine learns and captures the best knowledge to make accurate particular decisions such as Markov Decision Process.

In this thesis project, we are aiming to build a Decision Tree and a Naive Bayes classifier by using the training data and later we will evaluate whichever performs better.

3.4.1 Decision Tree Algorithm

A skeleton decision of tree induction algorithm called TreeGrowth is shown in **Fig. 3.6**. The input to this algorithm consists of the training records E and the attribute set F. The algorithm works by recursively selecting the best attribute to split the data (Step 7) and expanding the leaf nodes of the tree (Steps 11 and 12) until the stopping criterion is met (Step 1) [11].

The question here is, whether the new reference is ”Related” or ”Barely-related” to the concept of the paper. One approach is to ask a series of questions about the attributes of the test record and after we receive an answer, a next question will be asked until we reach a conclusion about the class label of the record. When a decision tree has been built up, classifying a test record starts in a way that from the root node, we apply the test condition to the record and get the appropriate branch based on the result of the test records. This will lead us either to another internal node, which a new test condition is applied or to a leaf node.

For applying the Decision Tree algorithm, we are interested in two features, “Frequency” and “Position” features. After extracting them we apply Decision tree algorithm to do the classification. First, we initialize weight value as a definition for relevancy such that, the references in the introduction are for general information or inspiration matter and not that effective into the author’s methodology, for that reason, the references appear in the introduction get one weight value, however, the body section get five weight value, since the references in this part contributing more to the methodology and in the related

```
TreeGrowth (E, F)
 1: if stopping_cond(E,F) = true then
 2:    leaf = createNode().
 3:    leaf.label = Classify(E).
 4:    return leaf.
 5: else
 6:    root = createNode().
 7:    root.test_cond = find_best_split(E, F).
 8:    let V = {v|v is a possible outcome of root.test_cond }.
 9:    for each v ∈ V do
10:       E_v = {e | root.test_cond(e) = v and e ∈ E}.
11:       child = TreeGrowth(E_v, F).
12:       add child as descendent of root and label the edge (root → child) as v.
13:    end for
14: end if
15: return root.
```

Figure 3.6: A skeleton decision tree induction algorithm [11]

work section position gets one weight value as they represented in the table **Table. 3.1** in feature extraction section.

If references cited more than once in different positions, consequently having different weight values as a sample shows in table **Table. 3.2**, we proposed the formula to calculate the final score by comparing the "Frequency" of references and finding the majority of occurrence and consequently take the weight corresponds to that position times its frequency. The equation in (3.5) illustrated for greater understanding. In the **Table. 3.2** "rel" is the abbreviation of "related work", "int" for "introduction" and "bod" for "body"

$$Position = Maximum(Frequency \times Weight_{eachSection}) \tag{3.5}$$

ID references	**Feature 1: frequency**	**position**	**Feature 2: Score_position**	**class**
1	2	(1*int, 1*rel)	1	Barely related
2	18	(5*int+13*bod)	65	Barely related
3	5	(1*int+1*bod+3*rel)	5	Related
4	1	(1*rel)	1	Related
5	9	(5*int+1*bod+3*rel)	5	Related

Table 3.2: Sample of feature in training data

Hunt's Algorithm

In general, a given set of attributes used to build a decision tree. Various algorithms have been developed to construct a reasonably accurate Decision Tree and these algorithms usually, apply a greedy strategy that grows a Decision Tree.

In Hunt's algorithm, a Decision Tree is grown in a recursive fashion by partitioning the training records into successively purer subsets. Let D_t be the set of training records that are associated with node t and y = $\{y_1, y_2,..., y_c\}$ be the class labels. The general procedures of Hunt's algorithm define as; If D_t contains records that belong to the same class y_t, then t is a leaf node labelled as y_t. If D_t is an empty set, then t is a leaf node labelled by the default class, y_d and if D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset [11].

There are two ways to split the data into smaller subsets. The following **Fig. 3.7** and **Fig. 3.8** figures illustrate how the algorithm works, regarding the two continues attributes to make the decision. If we start the tree with "Frequency" feature as node root, the test condition can be expressed as a comparisons test which stands for either equals to one, (citations that papers refer to them only once), or greater than one, (papers refer to them more than one time), and these tests will generate binary outcomes as giving Barely_class label or considering the next attribute which is Position, respectively.

In the second level of Decision Tree which starts from "Position" attribute, the test condition will be considered on weight value calculated from the table **Table. 3.2**. We define the test condition in a way that the value is either greater or equals to five or less than five to split the training records and consequently the class labels will be assigned. However, the question here is which one of these splits performs better and consequently gains more information.

Selecting the best Split

The issues with Decision tree rise when we need to determine how to split the records, what is the best split, when to stop splitting and how to specify the attribute test condition.

Attribute test condition depends on attribute type and the number of ways to split. We obtain continuous attributes whose lead to two-way split as the binary decision.

Since there are choices to specify the test conditions from the given training set, we need to specify a measurement to determine the best way to split the records. This measurement calls "Impurity". The larger the degree of purity, will obtain the better class distribution.

To recognize how well a test condition performs, we do the comparisons between the degree of parent's impurity of the before splitting with a degree of the child's impurity after splitting. The measurement of node impurity are Gini Index, Entropy and Misclassification Error.

If p(i|t) denotes the fraction of records belonging to the class "i" at a given node t. According to dataset which is a two-class problem (binary classification), the class distribution at any node can be presented as (p_0,p_1), such that, p_1=1-p_0. The smaller the degree of impurity, the more skewed the class distribution. Here we will use "Entropy" impurity measure as following formula (3.6), Where c is the number of classes and $0\log_2 0 = 0$ in

entropy calculations [11].

$$Entropy(t) = -\sum_{i=1}^{c-1} p(i|t) \log_2 p(i|t) \tag{3.6}$$

The choice of best split test condition is recognized by comparing the degree impurity of the parent node with the degree of impurity of the child nodes. If this difference is larger then the test condition is better. The information gain, Δ, is a criterion that can be used to determine the goodness of a split as the equation (3.7).

where I(.) is the impurity measure of a given node, N is the total number of records at the parent node, k is the number of attribute values, and N (v_j) is the number of records associated with the child node, v_j. Decision tree induction algorithms often choose a test condition that maximizes the gain Δ [11].

$$\Delta = I(Parent) - \sum_{i=1}^{k} \frac{N(v_j)}{N} I(v_j) \tag{3.7}$$

The information gained via splitting by weight position feature or frequency feature will be represented in the Result chapter.

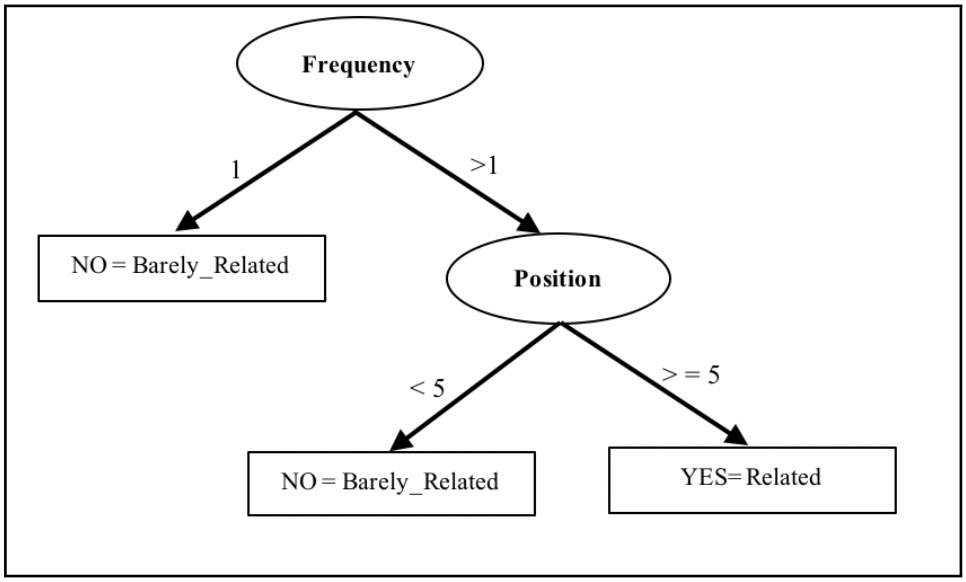

Figure 3.7: Hunt's algorithm for inducing a decision tree

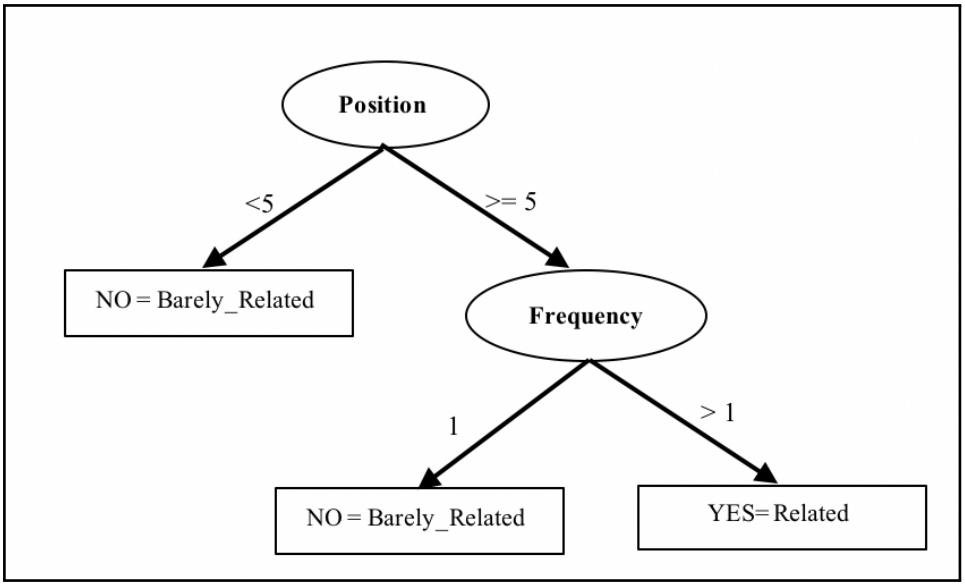

Figure 3.8: Hunta's algorithm for inducing a decision tree

3.4.2 Naive Bayes Algorithm

Naive Bayes classification presents a probabilistic relationship between the attribute set, citation's frequency and citation's position, with the class variable, and it works on Bayes theorem of probability to predict the class of unknown dataset with an assumption where feature's properties independently contribute to the probability. The Bayes theorem introduction for solving, classification problems will be explained in next section followed by a description of implementations for Naive Bayes classification.

Bayes theorem introduction

The Bayes Theorem is a statistical principle for combining prior knowledge of the classes with new evidence gathered from data by the following formula (3.8). Let X and Y be a pair of random variables, the Bayes theorem allows us to express the posterior probability in terms of the prior probability P(Y), the class-conditional probability P(X|Y), and the evidence P(X) [11].

$$P(Y|X) = \frac{P(Y|X)P(Y)}{P(X)} \tag{3.8}$$

The Bayes theorem can be used to solve the prediction and classification problem and in this thesis, the method will be used for classification. Let X denote the attribute set and Y denotes the class variable. P(Y) is the prior probability which can be estimated from the training set by the fraction of data that belong to each class and then we need to specify the class-conditional probability p(X|Y) which we apply Naive Bayes classification and finally, we need to learn the posterior probabilities P(Y|X) for every combination of X and

Y based on information gathered from the training data. A test record can be classified by finding the class that maximizes the posterior probability.

Naive Bayes classification

Naive Bayes classification is a statistical principle for combining prior knowledge of the classes with new evidence gathered from data.

A Naive Bayes classifier is a simple classifier. However, although it is simple, Naive Bayes can outperform more sophisticated classification methods. Besides that, it has also exhibited high accuracy and speed when applied to the large database [6].

In this section, we will discuss how the Naive Bayes classification applied regarding our training citation dataset. Naïve Bayes Classifier assumes the attributes are conditionally independent for the class-conditional probability and to classify a test record, it computes the posterior probability for each class Y. The general concept of the process illustrated in figure **Fig. 3.11**.

$$P(Y|\mathbf{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(\mathbf{X})}.$$

Figure 3.9: Naive Bayes algorithm [11]

To estimate conditional probabilities for continuous attributes, we assumed a Gaussian probability distribution for the continuous variable and using training data to estimate the parameters of the distribution as represented in the formula below. The distribution is characterized by two parameters, Mean and Variance.

$$P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

Figure 3.10: Gaussian probability distribution [11]

To classify a test record, the Naive Bayes classifier computes the posterior probability for each class Y as (“No”, “Yes”) whose stands for “Barely_related” and “Related” respectively. To estimate the class-conditional probability we considered the type of attributes, which in our dataset we only have continuous ones. We assumed a Gaussian distribution to represent the class-conditional probability as the formula represented in figure **Fig. 3.10**. We obtained ”Position” and ”Frequency” features to feed the model as before, we learn the model by 2/3 of the training data and then we will test the learned model by 1/3 of the data as testing data.

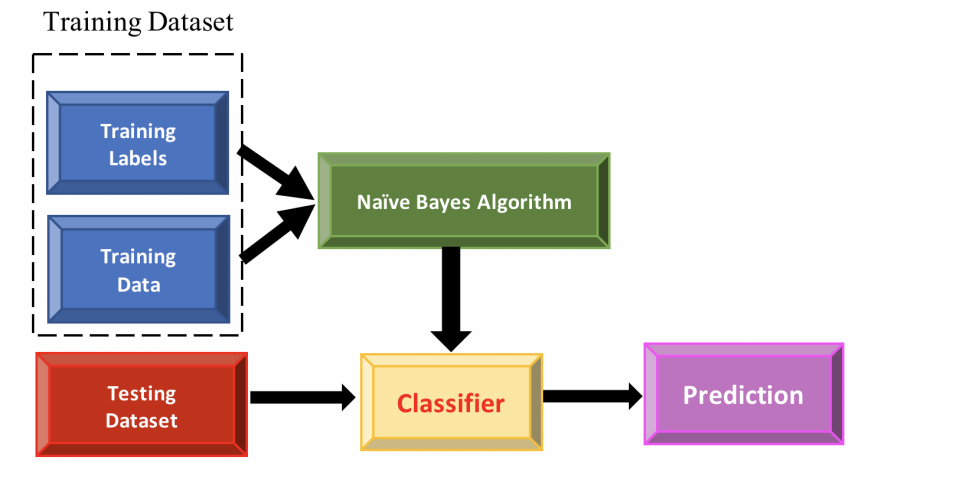

Figure 3.11: Naive Bayes algorithm process

4

Results and Evaluation

In this chapter, first, we provide statistical analysis of the three Machine Learning classification methods in the next three Sections and furthermore we explain the experiments which are conducted to evaluate their performance in section 4.4.

4.1 Prior algorithm result

Here, we will consider the result of Prior Algorithm which has been employed on our selected experimental data set. There are a dozen of papers available in the variety of resources, however because of the time limitation in the thesis project and a time-consuming transformation process and preprocessing the dataset to achieve the proper format from PDF format, we have decided to select few papers registered in the same conference as we explained it in chapter 2.

To apply the model we considered 180 citations and as the figure **Fig. 4.1** and **Fig. 4.2** shows one part of the results, where the model could successfully classify the training data into two classes, "Related" and "Barely_related" according to the Mean value of citation's frequency. The numbers in the figures represent the citation's ID along with the paper's title.

The model classifies the training dataset considering three statistical measurements base on citation's frequency. The percentage of the class labelled regarding the Prior algorithm's conditions showed in the following figures **Fig. 4.3**, **Fig. 4.4** and **Fig. 4.5**.

```
**********Mean value comparisons**********

the class of related references to the paper:
1
Transferring an Interactive Display Service to the Virtual Reality
9
Multi-Client Searchable Encryption over Distributed Key-Value Stores
7
Multi-Client Searchable Encryption over Distributed Key-Value Stores
16
Multi-Client Searchable Encryption over Distributed Key-Value Stores
14
Multi-Client Searchable Encryption over Distributed Key-Value Stores
23
Ubiquitous Tracking Using Motion and Location Sensor With Application to Smartphone
1
Hierarchical Demand Response for Colocation Data Centers
2
Hierarchical Demand Response for Colocation Data Centers
3
Hierarchical Demand Response for Colocation Data Centers
4
Hierarchical Demand Response for Colocation Data Centers
11
Hierarchical Demand Response for Colocation Data Centers
18
Hierarchical Demand Response for Colocation Data Centers
22
```

Figure 4.1: Sample Related class label of citations according to mean value

```
the class of barely related references to the paper:
2
Transferring an Interactive Display Service to the Virtual Reality
3
Transferring an Interactive Display Service to the Virtual Reality
4
Transferring an Interactive Display Service to the Virtual Reality
5
Transferring an Interactive Display Service to the Virtual Reality
6
Transferring an Interactive Display Service to the Virtual Reality
7
Transferring an Interactive Display Service to the Virtual Reality
8
Transferring an Interactive Display Service to the Virtual Reality
9
Transferring an Interactive Display Service to the Virtual Reality
10
Transferring an Interactive Display Service to the Virtual Reality
1
Multi-Client Searchable Encryption over Distributed Key-Value Stores
2
Multi-Client Searchable Encryption over Distributed Key-Value Stores
3
```

Figure 4.2: Sample Barely_Related class label of citations according to mean value

```
*****************************Mean value comparisons**************************

Percentage of Barely related citaion class regarding Mean:  80.0
Percentage of Related citaion class regarding Mean:  20.0
```

Figure 4.3: Percentage of class labeled of citations according to Mean value

```
*****************************Median value comparisons*************************

Percentage of Barely related citaion class regarding Median:  8.88888888888889
Percentage of Related citaion class regarding Median:  91.11111111111111
```

Figure 4.4: Percentage of class labeled of citations according to Median value

```
*****************************Variance value comparisons*************************

Percentage of Barely related citaion class regarding Variance:  26.666666666666668
Percentage of Related citaion class regarding Variance:  73.33333333333333
```

Figure 4.5: Percentage of class labeled of citations according to Variance value

The following chart **Fig. 4.6** illustrates the results of the Prior statistical classification. According to these experimental outcomes, the Mean value condition labelled 80% of citations into ”Barely_related” class, on the other hand, Median value classifies the citations in opposite way and labels 91,1% of citations as ”Related” class, however the Variance condition holds a moderate trend in classifying compare to the Mean and Median trends.

Technically the classification result of the experimental dataset depends on the number of citations. We applied the Prior algorithm to five papers containing 33 citations and as the result presented in the chart **Fig. 4.7**, 100% of citations for paper number 46 classified to the ”Related” class according to the Median and Variance criterion and as we increased the number of citations for classification model the Variance criterion decreased to 73,3%, although the Median criterion keeps its trend stable on continuously the highest ”Related” class by 91,1%.

The argument behind the Variance’ behaviour might be in its mathematical expression, since it is defined as the sum of the squared distances of each term in the distribution from the Mean value, divided by the number of terms in the distribution and the Mean value in the definition may influences the result as the model grows since the Mean value classifies most of the citations as ”Barely_related”.

Ideally, the relation between the paper and its citation must be classified according to the Median feature to have the highest chance of relevancy, although in reality, authors cite general information or previous related work to their papers as well and in that sense may Variance value is the better feature for the classification algorithm.

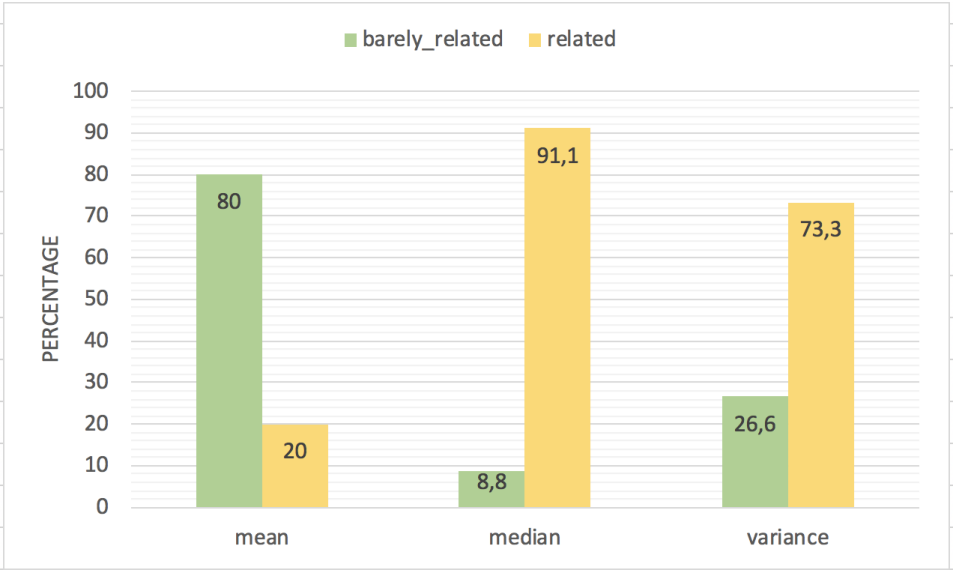

Figure 4.6: Comparison's result chart of Prior algorithm

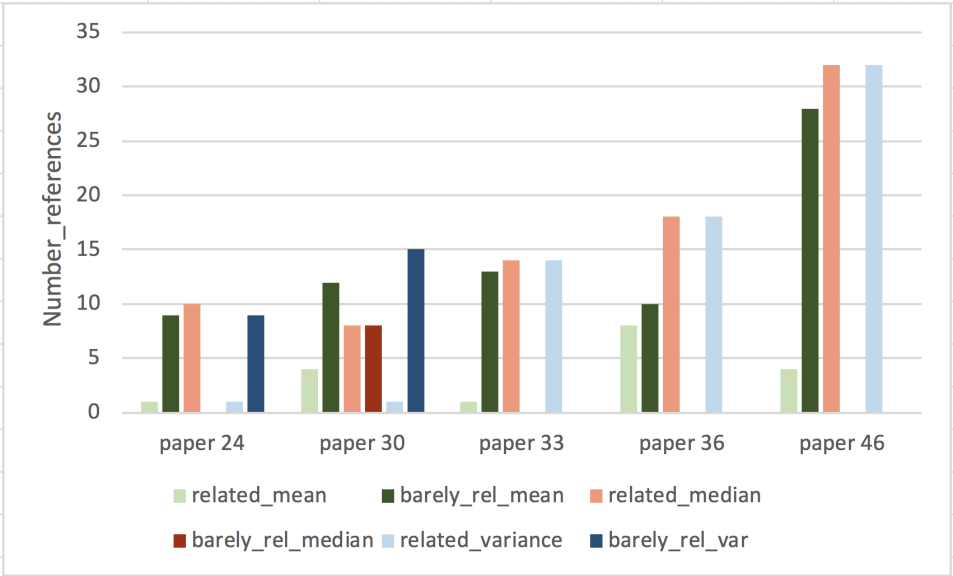

Figure 4.7: Prime classification applied on a sample of five papers

4.2 Decision Tree result

The Decision tree model considering 131 citations has been employed and the figure **Fig. 4.8** illustrates the result.

By considering two feature vectors as ”Frequency” and ”Position”, the Machine Learning algorithm partitions the data into two subsets. The results are comparable and relatively close whose more than half of the citations classified as ”Barely_related”, although the Entropy value is none when the Decision Tree splits by the ”Position” considering 5 value as splitting condition.

The information gained by splitting according to the ”Frequency” feature is 0.12 where splitting by ”Position” feature gives 0.93 information and these results conclude the best feature vector for partitioning the dataset is ”Position”.

```
Splitting by Frequency
        Barely related citaion class. Size = 84        Entropy = 0.7266645172796349
        Related citaion class. Size = 47       Entropy = 0.9601186626422924
Percentage of Barely related citaion class:  0.6412213740458016
Percentage of Related citaion class:  0.3587786259541985
Splitting by weight_pos
        Barely related citaion class. Size = 85        Entropy = 0.0
        Related citaion class. Size = 46       Entropy = 0.0
Percentage of Barely related citaion class:  0.648854961832061
Percentage of Related citaion class:  0.3511450381679389
Number of references:  131
Info gain if splitting by weight_pos: 0.9350865164204485
Info gain if splitting by Frequency: 0.12466364164428767
```

Figure 4.8: Information gained, entropy and classification result employed by a Decision Tree

We have applied the Decision Tree with different ”Position” feature vector to obtain how the model behaves and as it has shown in the figure **Fig. 4.9**, despite the information gained, has been reduced from 0,93 to 0,75 when we change the position’s condition value from 5 to 3, there is an increase in the number of citation in the ”Related” class from 46 to 51. With greater value for position, 10 instead of 5, the focus of the model will be only in the most important citations, therefore the model classifies most of the citations as ”Barely_related”, since their position weight is less than 10. Furthermore, we have a decrease in information gained and for the mentioned reasons the most accurate condition value is 3 for the ”Weight” feature to split the Decision Tree.

Consequently, we changed the ”Frequency” feature value from 1 to 3 and as the figure below **Fig. 4.10** shows, the information gained has a significant increase from 0,12 to 0,75 however, the number of citations to be classified has been decreased since by increasing the frequency feature from 1 to 3 we eliminate all of the references have been cited less than 3. Therefore we conclude the best condition’s value for splitting the dataset are 5 and 3 for ”Weight” and ”Frequency” feature vectors respectively.

```
Splitting by Frequency
         Barely related citaion class. Size = 84        Entropy = 0.7266645172796349
         Related citaion class. Size = 47       Entropy = 0.9601186626422924
Percentage of Barely related citaion class:  0.6412213740458016
Percentage of Related citaion class:  0.3587786259541985
Splitting by weight_pos
         Barely related citaion class. Size = 109       Entropy = 0.7605024019419502
         Related citaion class. Size = 22       Entropy = 0.0
Percentage of Barely related citaion class:  0.83206106870229
Percentage of Related citaion class:  0.16793893129770993
Number of references:  131
Info gain if splitting by weight_pos: 0.30230207510997087
Info gain if splitting by Frequency: 0.12466364164428767


Splitting by Frequency
         Barely related citaion class. Size = 84        Entropy = 0.7266645172796349
         Related citaion class. Size = 47       Entropy = 0.9601186626422924
Percentage of Barely related citaion class:  0.6412213740458016
Percentage of Related citaion class:  0.3587786259541985
Splitting by weight_pos
         Barely related citaion class. Size = 80        Entropy = 0.0
         Related citaion class. Size = 51       Entropy = 0.4627490585781739
Percentage of Barely related citaion class:  0.6106870229007634
Percentage of Related citaion class:  0.3893129770992366
Number of references:  131
Info gain if splitting by weight_pos: 0.7549323027755106
Info gain if splitting by Frequency: 0.12466364164428767
```

Figure 4.9: Information gained, entropy and classification result employed by a Decision Tree with Frequency=1, Weight=10 in the top and Weight=3 in the bottom

```
Splitting by Frequency
         Barely related citaion class. Size = 84        Entropy = 0.7266645172796349
         Related citaion class. Size = 47       Entropy = 0.9601186626422924
Percentage of Barely related citaion class:  0.6412213740458016
Percentage of Related citaion class:  0.3587786259541985
Splitting by weight_pos
         Barely related citaion class. Size = 85        Entropy = 0.0
         Related citaion class. Size = 46       Entropy = 0.0
Percentage of Barely related citaion class:  0.648854961832061
Percentage of Related citaion class:  0.3511450381679389
Number of references:  131
Info gain if splitting by weight_pos: 0.9350865164204485
Info gain if splitting by Frequency: 0.12466364164428767


Splitting by Frequency
         Barely related citaion class. Size = 13        Entropy = 0.961236604722876
         Related citaion class. Size = 17       Entropy = 0.672294817075638
Percentage of Barely related citaion class:  0.09923664122137404
Percentage of Related citaion class:  0.1297709923664122
Splitting by weight_pos
         Barely related citaion class. Size = 85        Entropy = 0.0
         Related citaion class. Size = 46       Entropy = 0.0
Percentage of Barely related citaion class:  0.648854961832061
Percentage of Related citaion class:  0.3511450381679389
Number of references:  131
Info gain if splitting by weight_pos: 0.9350865164204485
Info gain if splitting by Frequency: 0.7524522587740116
```

Figure 4.10: Information gained, entropy and classification result employed by a Decision Tree with features Weight=5 and Frequency=3

According to the Experimental results as expressed earlier the optimal model of Decision is, when the model split by the ”Position” feature and the condition value 5. This model illustrated in **Fig. 4.11** where the blue dots show the ”Related” class, as well as red dots, represent ”Barely_related” class.

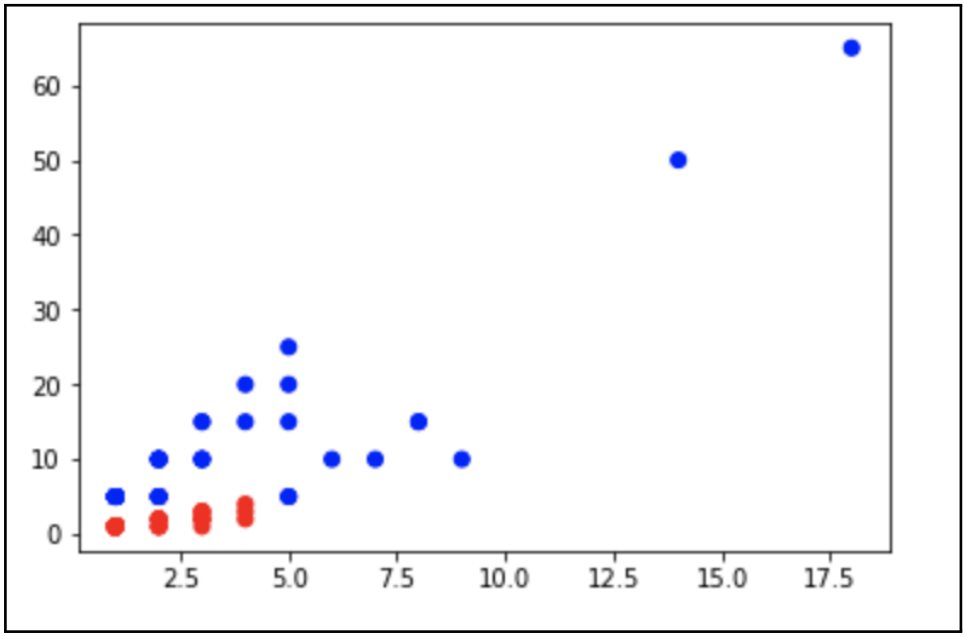

Figure 4.11: Classification result regarding splitting by ”position” feature

4.3 Naive Bayes result

Here we discuss the result from Naïve Bayes classifier, the Posterior probability, considering same feature sets as Decision Tree algorithm with the independent class conditional assumption, the result depicted in figure **Fig. 4.12**, ”No” label represents ”Barely_related” and ”YES” label stands for ”Related” class, where 67% of training citations classified as ”Barely_related” and 32% are ”related” class. The result provides the Mean and Variance of the data grouped by each class so that we can estimate the Posterior probabilities on the assumption that each of these classes can be represented by a Gaussian distribution. We observed that the results from Naïve Bayes classification are similar to the Decision Tree method as we evaluate their performance in the next section.

As the algorithms applied to our dataset do not predict or estimate, thus, there are no classification errors considered. From the transformed dataset, we selected a test set and a training set with 2/3 portion of data for training and 1/3 for testing the model, then Naive Bayes algorithm trained on the training dataset and evaluated against the test set, moreover there are no misclassifications due to consistencies in the dataset.

```
{'NO': {'P(Y)': 0.6704545454545454,
        'ref_freq': {'mean': 1.2881355932203389, 'var': 0.51019821890261408},
        'ref_pos': {'mean': 1.1864406779661016, 'var': 0.28727377190462516}},
 'YES': {'P(Y)': 0.32954545454545453,
         'ref_freq': {'mean': 3.4827586206896552, 'var': 12.594530321046378},
         'ref_pos': {'mean': 11.379310344827585, 'var': 132.58026159334128}}}
```

Figure 4.12: Hunta's algorithm for inducing a decision tree

4.4 Evaluation Machine Learning algorithms' results

There are performance measurements that assist to evaluate the models and find out which method offers higher performance. For instance, classification Accuracy measurement for classification problems, which computed by the total correct correction class labelled divided by the total predictions made. Although in our domain problem, it does not make sense to calculate the accuracy which will be 1, since the Machine Learning algorithms were Unsupervised methods.

The graph **Fig. 4.13** shows all the methods applied to the dataset. The results present that Machine Learning algorithms performed fairly similar by classifying 64,12% for the citations that treated as "Barely_related" according to the Decision Tree considering "Frequency" feature for splitting, 64,88% to the "Barely_related" class according to the Decision Tree considering "Position" feature for splitting and 67,04% "Barely_related" citations according to the Naive Bayes classification. On the other hand in the Prior algorithm only Median and Variance, comparison's measures, considering the majority of the citations related to paper cites them.

We successfully classified the citation dataset by two approaches, in the first approach we proposed the Prior algorithm with considering Mean, Median and Variance values and despite the Mean measurements, the method classified the dataset mostly as "Related" class, although in the second approach, we applied Machine Learning algorithms as Decision tree and Naive Bayes methods, whose behave pretty similar in the classification. They mostly identified the citations "Barely_related".

Although these performances directly related to how big is the dataset, we selected the small dataset for this thesis, since the transformation and preprocessing steps were time-consuming.

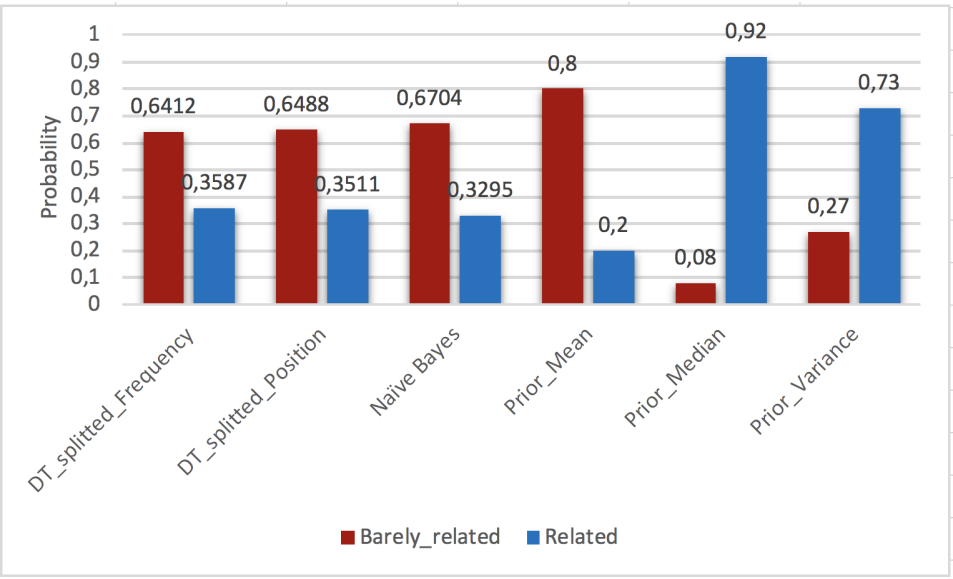

Figure 4.13: Comparing the performance according to two Machine Learning models

5 Conclusion

This chapter consists of two sections. The first section, provides a brief overview of what we covered in this report. Section 5.2 discusses our suggestions for future work and our further investigation which will be considered in the article related to this thesis.

5.1 Achievements

In this study, we successfully classified the citation dataset in order to define the relation between paper and its citations. We developed three different models, Prior algorithm, Decision Tree and Naïve Bayes as learning algorithms for classification task and create the feature set as ”Frequency” and ”Position”. The classifiers were tested on 131 sample citations. We selected small dataset due to The time limitation for this project along with time-consuming preprocessing steps for transforming format and feature extraction.

To implement the Prior algorithm according to the statistical measurements, we faced challenges such as data preparation and transformation process to obtain a suitable format for the model, this algorithm has limitation since it applied on dataset considering only “frequency” feature.

Naive Bayes classifier has been extremely fast compared to the two other methods and it does not require a very large training set to obtain a good estimate of the probability, however, decision tree was easier to implement and understand since it does not have much calculations. Our goal was to separate citation’s data and group the instances together in the classes they belong to. The Decision tree was constructed from the training dataset. To create the Decision Tree and finding the best split, the node must be as pure as possible by minimizing the Entropy and consequently gaining more information which the ”Position” feature by the 5 value weight obtained the best split condition.

5.2 Future work

In this section, we present the following suggestions for future work as this project holds high potential to investigate furthermore. Regarding optimization, one can consider the bigger dataset and test the models in order to explore how the Machine Learning methods behave with respect to their performance for classification and time-consuming.

Unlike classification and regression, which analyze class-labeled (training) datasets, Clustering analyzes data objects without consulting class labels and can be used to generate class labels for a group of data [6].

Another suggestion to extend the project will be considering the Clustering methods such as the k-nearest neighbor, hierarchical clustering, Gaussian mixture models, hidden Markov models and etc, as we will be employing this concept in the article related to this thesis later on. Basically, clustering involves grouping data with respect to their similarities, the distance measures and clustering algorithms will be captured to calculate the difference between instances and group them consequently.

Supervised and unsupervised learning methods have traditionally focused on data consisting of independent instances of a single type. However, many real-world domains are best described by relational models in which instances of multiple types are related to each other in complex ways. For instance, in the experimental dataset containing scientific paper, they are related to each other via citation and are also related to their authors [12].

One interesting task to investigate for future work will be discovering a loop in a network of citations with regard to explore the accurate relevancy between two papers cited each other as it has illustrated as a small network of citations in figure **Fig. 5.1**, however, this method demands availability of the paper's citations' actual context. .

Another extensions to this work, will be selecting different classification models other than mentioned methods have been developed, like super vector machine, neural networks, Gaussian mixture and so on, any improvement to the performance

Regarding the feature set, one can extend the set to more vectors like considering the "Author" or "Title" as well as considering different pair of feature set instead of the aforementioned feature set.

The resulting analysis can be expanded as well, the Machine Learning algorithms applied to classify the dataset into two classes "Barely_related" and "Related", however for the citations classified as "Barely_related" group, with the help of accessing to the references context and authors, one can consider text mining techniques like Term-Based Method, Phrase-Based Method and Concept-Based Method [4], to explore any citations that are not related to the paper cited those particular references at all and the reason for this might be author selected those citations from his or her scientific network intentionally or chose those citations by misinterpreting. We will investigate this concept furthermore in the article related to thesis's topic subsequently.

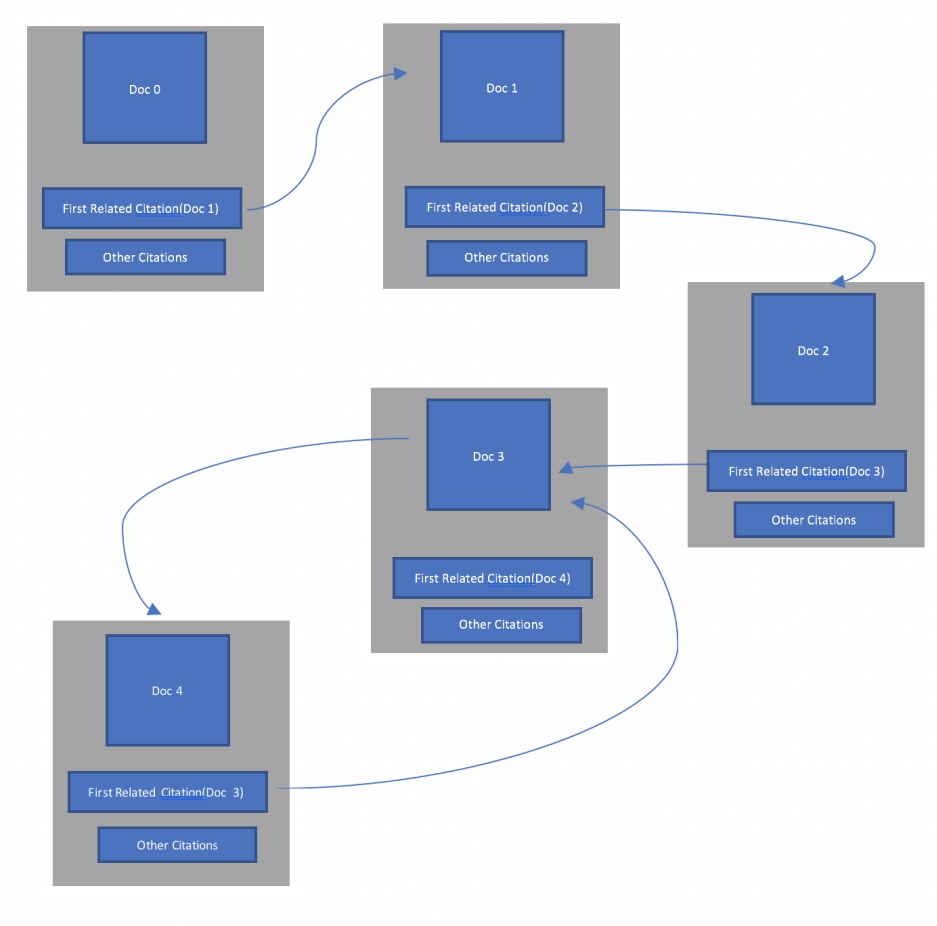

Figure 5.1: Sample of small network of citations having a loop

Bibliography

[1] Hanadi Alfraidi, Won-Sook Lee, and David Sankoff. Literature visualization and similarity measurement based on citation relations. In *Information Visualisation (iV), 2015 19th International Conference on*, pages 217–222. IEEE, 2015.

[2] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml) 1.0, 2008.

[3] Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 177–180. ACM, 2013.

[4] Sonali Vijay Gaikwad, Archana Chaugule, and Pramod Patil. Text mining methods and techniques. *International Journal of Computer Applications*, 85(17), 2014.

[5] Bela Gipp, Jöran Beel, and Christian Hentschel. Scienstein: A research paper recommender system. In *Proceedings of the international conference on emerging trends in computing (icetic'09)*, pages 309–315, 2009.

[6] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[7] Haifeng Liu, Zhuo Yang, Ivan Lee, Zhenzhen Xu, Shuo Yu, and Feng Xia. Car: Incorporating filtered citation relations for scientific article recommendation. In *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*, pages 513–518. IEEE, 2015.

[8] Tetsuya Nakatoh, Hayato Nakanishi, Kensuke Baba, and Sachio Hirokawa. Focused citation count: a combined measure of relevancy and quality. In *Advanced Applied Informatics (IIAI-AAI), 2015 IIAI 4th International Congress on*, pages 166–170. IEEE, 2015.

[9] Yuliant Sibaroni, Dwi Hendratmo Widyantoro, and Masayu Leylia Khodra. Survey on research paper's relations. In *Information Technology Systems and Innovation (ICITSI), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[10] Kritsada Sriphaew and Thanaruk Theeramunkong. Measuring the validity of document relations discovered from frequent itemset mining. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 293–299. IEEE, 2007.

[11] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, and ZhaoHui Tang. Introduction to data mining.

[12] Benjamin Taskar, Eran Segal, and Daphne Koller. Probabilistic classification and clustering in relational data. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 870–878. Lawrence Erlbaum Associates LTD, 2001.

[13] Mark Ware and Michael Mabe. The stm report: An overview of scientific and scholarly journal publishing. 2015.

[14] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[15] Yan Yang and Long Yun. Literature recommendation based on reference graph. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 3, pages V3–400. IEEE, 2010.

[16] Hou-kui Zhou, Hui-min Yu, and Roland Hu. Topic discovery and evolution in scientific literature based on content and citations. *Frontiers of Information Technology & Electronic Engineering*, 18(10):1511–1524, 2017.