

5-2010

A Comparative study on text categorization

Aditya Chainulu Karamcheti
University of Nevada Las Vegas

Follow this and additional works at: <https://digitalscholarship.unlv.edu/thesesdissertations>



Part of the [Databases and Information Systems Commons](#), and the [Library and Information Science Commons](#)

Repository Citation

Karamcheti, Aditya Chainulu, "A Comparative study on text categorization" (2010). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 322.
<https://digitalscholarship.unlv.edu/thesesdissertations/322>

This Thesis is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

A COMPARATIVE STUDY ON TEXT CATEGORIZATION

by

Aditya Chainulu Karamcheti

Bachelor of Technology in Information Technology
Jawaharlal Nehru Technological University, India
May 2007

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2010



THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Aditya Chainulu Karamcheti

entitled

A Comparative Study on Text Categorization

be accepted in partial fulfillment of the requirements for the degree of

Master of Science

Kazem Taghva, Committee Chair

Ajoy K. Datta, Committee Co-chair

Laxmi P. Gewali, Committee Member

Muthukumar Venkatesan, Graduate Faculty Representative

Ronald Smith, Ph. D., Vice President for Research and Graduate Studies
and Dean of the Graduate College

May 2010

ABSTRACT

A Comparative Study On Text Categorization

by

Aditya Chainulu Karamcheti

Dr. Kazem Taghva, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

Automated text categorization is a supervised learning task, defined as assigning category labels to new documents based on likelihood suggested by a training set of labeled documents. Two examples of methodology for text categorizations are Naive Bayes and K-Nearest Neighbor.

In this thesis, we implement two categorization engines based on Naive Bayes and K-Nearest Neighbor methodology. We then compare the effectiveness of these two engines by calculating standard precision and recall for a collection of documents. We will further report on time efficiency of these two engines.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS	viii
CHAPTER 1 INTRODUCTION	1
Thesis Overview	2
CHAPTER 2 BACKGROUND OF TEXT CATEGORIZATION	4
2.1 Text Categorization.....	5
2.2 General Approach to Text Categorization	6
2.3 Bayesian Categorization	8
2.3.1 Bayes Thoerem.	8
2.4 Naïve Bayes Categorization.....	9
2.4.1 Example for predicting a category using Naive Bayes Categorization.....	11
2.5 k-Nearest Neighbor Categorization.....	14
2.5.1 Example for predicting a category using K- Nearest Neighbor Categorization.	16
CHAPTER 3 IMPLEMENTATION.....	18
3.1 Document Collection	18
3.2 Document Processing.....	21
3.3 Algorithm Implementation – Naïve Bayes	31
3.3.1 Test Phase	32
3.4 Algorithm Implementation – k Nearest Neighbor	34
3.4.1 Test Phase	35
3.5 Precision and Recall	36
CHAPTER 4 RESULTS EVALUATION AND PROPERTIES	37
4.1 Results.....	37
4.1.1 Results – Naïve Bayes Categorization.....	38
4.1.2 Results – k Nearest Neighbor Categorization.....	40
4.2 Results Evaluation	42
4.3 Properties.....	48
4.3.1 Properties of Naïve Bayes Categorization	48
4.3.2 Limitations of Naïve Bayes Categorization.....	49
4.3.3 Properties of k Nearest Neighbor Categorization.....	50
4.3.4 Limitations of k Nearest Neighbor Categorization.....	51

CHAPTER 5 CONCLUSION AND FUTURE WORK	52
BIBLIOGRAPHY	54
VITA	57

LIST OF TABLES

Table 1. Training dataset that describes the weather conditions.....	13
Table 2. Reuters – 21578 collection categories	19
Table 3. Training data set collection.....	20
Table 4. Example text, each line represents a document	26
Table 5. A record level inverted index file for the text in table 4	27
Table 6. Terms with their document frequencies	28
Table 7. Test documents collection.....	34
Table 8. Precision and Recall values of Naïve Bayes when $\theta = 5\%$	38
Table 9. Precision and Recall values of kNN when $\theta = 5\%$	40

LIST OF FIGURES

Figure 1.	Pictorial representation of Categorization	6
Figure 2.	General approach for building a categorization model [6].....	8
Figure 3.	Feature Space and when $k=1$	15
Figure 4.	Feature Space when $k=2$ and $k=3$	16
Figure 5.	Screenshot of Acquisition category in SGML format.....	21
Figure 6.	Screenshot of a Parsed XML Document	22
Figure 7.	Screenshot of the list of tokens after Tokenization	23
Figure 8.	Screenshot of the list of stop words from the code	24
Figure 9.	Screenshot of the output after Stemming	25
Figure 10.	Screenshot of the terms with their documents frequencies ..	28
Figure 11.	Screenshot of term document frequencies and weights.....	31
Figure 12.	Pseudo code of the Naïve Bayes Algorithm [27]	33
Figure 13.	Pseudo code of the kNN algorithm implementation [29]	35
Figure 14.	Sample output of Naïve Bayes classifier.....	39
Figure 15.	Sample output of k Nearest Neighbor classifier.....	41
Figure 16.	Document D_1 categorization using Naïve Bayes and kNN.....	43
Figure 17.	Document D_{79} categorization using Naïve Bayes and kNN ...	44
Figure 18.	Document D_{107} categorization using Naïve Bayes and kNN ..	46
Figure 19.	Document D_{89} categorization using Naïve Bayes and kNN ...	47

ACKNOWLEDGEMENTS

Words would not suffice, but I would like to take this opportunity to express my gratitude to those who have helped me bring this thesis to a successful end with their knowledge, help and co-operation.

I would like to thank Dr. Kazem Taghva, my research advisor for all the support and guidance he has offered me during the course of my graduate studies at University of Nevada, Las Vegas. His encouragement and valuable suggestions have helped me immensely in seeking the right direction for research. I express the heartfelt debt that I owe to Dr. Ajoy K. Datta for corroborating me in various ways both as a Graduate coordinator and my thesis committee member. I am also grateful to Dr. Laxmi P. Gewali for being in my committee. I am glad that I have taken couple of his courses which have helped me in improving my technical and problem solving skills. I would also like to thank Dr. Venkatesan Muthukumar who has accepted to be my graduate college representative. A special thanks to Dr. Robert Abella for his ample support and help during my Research Assistantship work under him.

My deepest gratitude to my parents for their love, care and opportunities they have provided me at every stage in my life. They have always placed my academics and happiness first and that has given me the motivation and strength to do better at each step. Sowmi has always taken care of me and been the most wonderful sister and a great friend. I

shall remain ever obliged to Raj, Kinnu and Raghu and all my cousins for their continuous support in all my endeavors. Last but not the least, I would like to thank all my friends and roommates for their support.

CHAPTER 1

INTRODUCTION

Text Categorization is the automatic classification of text documents under predefined categories or classes. Information Retrieval (IR) and Machine Learning (ML) techniques are used to assign keywords to the documents and classify them in to specific categories. Machine learning helps us to categorize the documents automatically. Information Retrieval helps us to represent the text as an attribute. The task of automated text categorization has witnessed a thriving significance since a decade both from the researchers as well as the developers [1, 19].

Manually organizing large document bases is extremely difficult, time consuming, error prone, expensive and is often not feasible. Automated text categorization is a viable option for larger organizations which has got time and money as the main constraints. Automated text categorization has reached the highest accuracy levels with a combination of IR and ML techniques when compared with trained professionals and comes as a rescue for Modern Classification.

Document indexing, spam filtering, populating the hierarchical catalogues of web resources, document genre identification, automated essay grading, and categorizing news paper ads are some of the important applications of Text Categorization in the field of science and

technology. It is also used in the fields of finance, sports and entertainment and medical sciences [2].

This thesis deals with the comparative study on text categorization. It involves categorizing various documents collection using Naive Bayes based on Bayes theorem and K- Nearest Neighbor methodologies, well known data mining techniques. Naive Bayes method calculates the maximum and minimum possible probabilities for a document to belong to a category. K-Nearest Neighbor method finds the nearest neighbors that belong to the same category by calculating the Euclidean distance measures.

Both these methods initially start with the parsed documents and significant terms obtained from the training documents after preprocessing techniques. These documents are used to train the categorizer. Once the training phase is done, categorization engines based on Naive Bayes and K –Nearest Neighbor are implemented to predict the categories of the documents. We then compare the effectiveness of these two engines by calculating standard precision and recall for a collection of test documents.

Thesis Overview

This thesis is organized in to five different chapters. Chapter 1 gives us the Introduction and brief explanation about categorization. Chapter 2

deals with the background of categorization. It also defines Naive Bayes categorization and K-Nearest Neighbor categorization. Examples are discussed using small applications of these two techniques. We discuss different preprocessing techniques applied on documents collection and clear implementations in Chapter 3. All experimental results obtained are tabulated, compared and evaluated in Chapter 4. In Chapter 5, we conclude our thesis with a brief overview of future work.

CHAPTER 2

BACKGROUND OF TEXT CATEGORIZATION

Today's world is weighed down with lots of data and information from various sources. Advanced IT field makes the collection of data easier than ever before. Data Mining is a process of extracting interesting patterns and knowledge from a huge amount of data. It is a new field of study and research and created large interests in business communities. In recent times, data mining not only attracted business organizations, but also the IT industry. It mainly helps the real world applications, to convert large amount of data to useful information. Data Mining is used in various field of scientific research, businesses, banking sector, intelligence agencies and many more. We have many well known Data Mining tasks. Categorization is one among them on which this thesis mainly concentrates on [3].

Categorization is one of the supervised machines learning technique. Machine learning is a self-ruling system which is capable of acquiring and integrating knowledge constantly. This ability to learn from previous experiences, analytical observation, and other means, results in a system that can endlessly self-improve to offer increased efficiency and effectiveness. There are different types of machine learning techniques.

- Supervised learning
- Unsupervised learning

- Semi-supervised learning
- Reinforcement learning [4].

This thesis deals with text categorization using Naive Bayes and K-Nearest Neighbor algorithms, which are supervised learning techniques. Supervised Learning is a technique in which results are deduced from a training set. Training set is one which contains pairs of input data and category labels to which they belong to. Train data is initially categorized by experts. Once the categorization engine is trained, it must be able to categorize the test data to its appropriate category [5].

2.1 Text Categorization

Categorization is classifying the data for its most effective and efficient use. It is one of the most popular and important supervised learning techniques in data mining. Let $(d_j, c_i) \in D \gg C$, where D is the collection of documents and $C = \{c_1, c_2, \dots, c_{|C|}\}$ are set of categories which are predefined. Then the main task of Text Categorization is to assign a Boolean value to each pair in D [9].

Consider the Figure 1, in which D is the Domain of documents and C_1, C_2 and C_3 are different categories. D contains three different kind of documents namely '@', '\$' and '&'. After categorization, each document is categorized in to its respective category.

a collection of input data set is used. This data set is sub divided into Training Data Set and Test Data Set [6].

Training Data Set refers to the collection of records whose class labels are already known and is used to build the categorization model. It is then applied to the test data set.

Test Data Set refers to the collection of records whose class labels are known but when given as an input to the built categorization model, should return the accurate class labels of the records. It determines the accuracy of the model based on the count of correct and incorrect predictions of the test records [6].

Figure 2 shows us the general approach to build a categorization model for solving categorization problems.

There are many categorization techniques in use. They are:

1. Bayesian Categorization.
2. K Nearest Neighbor Categorization.
3. Decision Tree Categorization.
4. Rule Based Categorization.
5. Support Vector Machines.
6. Neural Networks.

In this thesis, we discuss and implement two major categorization techniques, Bayesian and K Nearest Neighbor methodologies

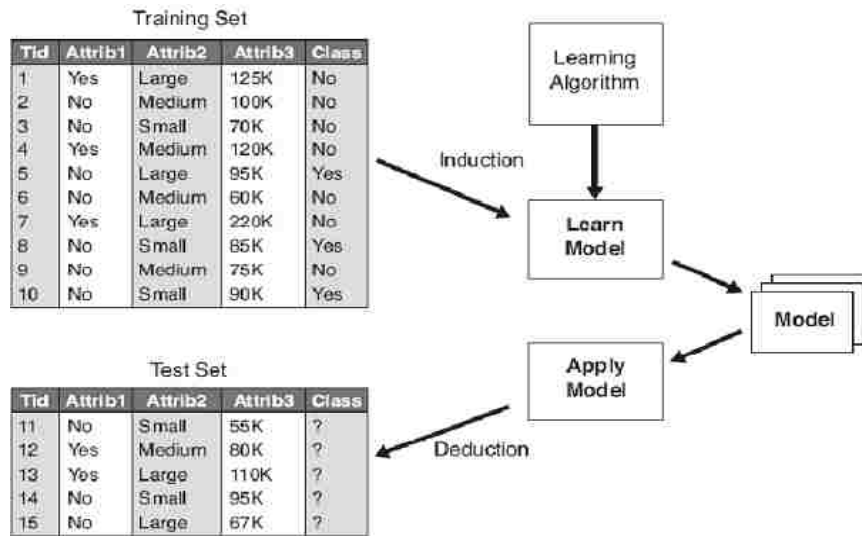


Figure 2. General approach for building a categorization model [6]

2.3 Bayesian Categorization

Bayesian is one of the most well known techniques of categorization. It is used to predict the class membership probabilities i.e. probability of a given record belongs to a particular category which is based on Bayes Theorem. Bayes theorem is a simple mathematical formula used for calculating conditional probabilities [7].

2.3.1 Bayes Thoerem.

Let us study about Bayes Theorem using a small example. X is a sample data record whose category is not known and H is some assumption. Let sample X belongs to a specified category C . If one needs to determine $P(H|X)$ the probability that the assumption H holds given the data sample X .

Bayes Theorem is given as:

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

Where $P(H|X)$ is the posterior probability of H on X. Posterior probability is based on information such as background knowledge rather than the prior probability which is independent of data sample X.

In the same way, $P(X|H)$ is the posterior probability of X on H. If the given data is huge data sample, it would be difficult to calculate above probabilities. Conditional independency was introduced to overcome this limitation.

2.4 Naïve Bayes Categorization

Naive Bayes categorization is one of the simplest probabilistic Bayesian categorization. It is based on an assumption that the effect of an attribute value on a given category is independent of the values of other attributes which is called as conditional independence. It is used to simplify complex computations [10]. The Naive Bayes classifier is a probabilistic classifier which is based on the Naïve bayes assumption.

From Bayes rule, the posterior probability can be given as

$$P(c | x) = \frac{P(c) p(x | c)}{P(x)}$$

Where x is a feature vector and $x = (x_1, \dots, x_n)$ and c is category.

Assume that the category c_{\max} yields to the maximum value for $P(c|x)$.

The parameter $P(c)$ is estimated as

$$P(c) = \frac{\text{Number of documents in } c}{\text{Number of documents}}$$

The categorization results are not affected because parameter $p(x)$ is independent of categories.

Assuming that the components of feature vectors are statistically independent of each other, $p(x|c)$ can be calculated as

$$p(x|c) = \prod_i p(x_i|c),$$

If the maximum estimation is used then

$$P(x_i|c) = \frac{N(x_i, c)}{N(c)}$$

Where $N(x, c)$ is the joint frequency of x and c ,

$$N(c) = \sum_x N(x, c)$$

If some data x_i disappears in the training data, the probability of any instance containing x_i becomes zero, without considering the other features in the vector.

Therefore to avoid zero probability, using Laplacian prior probabilities, $p(x_i | c)$ is estimated as follows

$$p(x_i | c) = \frac{N(x_i, c) + \lambda}{N(c) + \lambda |V|}$$

Where λ is a positive constant and is chosen as 1.0 or 0.5, and $|V|$ denotes the number of features.

The Naive Bayes classifier predicts the category C_{\max} with the largest posterior probability [11]:

$$\begin{aligned} C_{\max} &= \operatorname{argmax}_c P(c | \mathbf{x}) \\ &= \operatorname{argmax}_c P(c) p(\mathbf{x} | c) \end{aligned}$$

2.4.1 Example for predicting a category using Naive Bayes Categorization.

Consider the following table 1 containing training set of data which describes the weather conditions for playing tennis. Sample data records are represented by a set of attributes such as Outlook, Temperature, Humidity, Windy and categories by attribute play. Play is represented as either “Yes” or “No”. Each sample data record is represented as a vector. There are a total of fourteen vectors out of which nine vectors belong to the category “Yes” and five vectors belongs to the category “No”.

Suppose an unknown sample $X = (\text{Rain, Hot, Normal, Weak})$ is given. The

model computes to which category X belongs by calculating $P(X | \text{play} = \text{"Yes"})$, $P(\text{play} = \text{"Yes"})$ and $P(X | \text{play} = \text{"No"})$, $P(\text{play} = \text{"No"})$. Sample X is mapped to category having maximum posterior probability. Initially prior probability for each category can be computed based on the training sample. A Naive Bayes categorization model can now be built from the training data set as shown below.

Probability for playing: $P(\text{play} = \text{"Yes"}) = 9/14 = 0.642$

Probability for Not playing: $P(\text{play} = \text{"No"}) = 5/14 = 0.357$

Conditional probabilities for sample X are deduced as follows:

$P(\text{Rain} | \text{Yes})$, $P(\text{Hot} | \text{Yes})$, $P(\text{Normal} | \text{Yes})$, $P(\text{Weak} | \text{Yes})$,

$P(\text{Rain} | \text{No})$, $P(\text{Hot} | \text{No})$, $P(\text{Normal} | \text{No})$ and $P(\text{Weak} | \text{No})$.

$P(\text{Rain} | \text{Yes}) = 3/9$

$P(\text{Rain} | \text{No}) = 2/5$

$P(\text{Hot} | \text{Yes}) = 2/9$

$P(\text{Hot} | \text{No}) = 2/5$

$P(\text{Normal} | \text{Yes}) = 6/9$

$P(\text{Normal} | \text{No}) = 1/5$

$P(\text{Weak} | \text{Yes}) = 6/9$

$P(\text{Weak} | \text{No}) = 2/5$

Using the above probabilities, we get:

$P(X | \text{play} = \text{"Yes"}) = 3/9 * 2/9 * 6/9 * 6/9 = 0.329$

$P(X | \text{play} = \text{"No"}) = 2/5 * 2/5 * 1/5 * 2/5 = 0.012$

$$P(\text{play} = \text{"Yes"} \mid X) = P(X \mid \text{play} = \text{"Yes"}) P(\text{play} = \text{"Yes"})$$

$$= 0.329 * 0.642 = 0.211$$

$$P(\text{play} = \text{"No"} \mid X) = P(X \mid \text{play} = \text{"No"}) P(\text{play} = \text{"No"})$$

$$= 0.012 * 0.357 = 0.004$$

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Table 1. Training dataset that describes the weather conditions

Probability of $P(\text{play} = \text{“Yes”} \mid X)$ is greater than probability of $P(\text{play} = \text{“No”} \mid X)$.

Therefore, the Naive Bayes categorization model maps sample X to category “Yes” [17].

2.5 k-Nearest Neighbor Categorization

Nearest Neighbor search is an optimization problem for finding closest points in metric spaces. It is also known as similarity search or closest point search. For a given set of points S in a metric space M and a query point q , the problem is to find the closest point in S to q . Usually the distance is measured by Euclidean distance [12].

The k-Nearest Neighbor (k-NN) categorization is the simplest among all the supervised machine learning techniques but widely used method for classification and retrieval. It classifies the objects based on the closest training examples in the feature space. It is an instance based learning and often called lazy learning algorithm. Here the object instance query is classified based on the majority of k nearest neighbor category. All the k nearest neighbors in a database of a query are found by calculating Euclidean distance measure. The neighbors of a query instance are taken from the data set of objects which are already categorized of the classification is previously known [14].

The k-nearest-neighbor classifier is based on the Euclidean distance between a test sample and the specified training samples. Let \mathbf{x}_i be an input sample with P features $(x_{i1}, x_{i2}, \dots, x_{iP})$, n be the total number of input samples ($i = 1, 2, \dots, n$) and P the total number of features ($j = 1, 2, \dots, P$). The Euclidean distance between sample \mathbf{x}_i and \mathbf{x}_l ($l = 1, 2, \dots, n$) is defined as [13]

$$d(\mathbf{x}_i, \mathbf{x}_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \dots + (x_{iP} - x_{lP})^2}.$$

In this way, the class which is represented by the largest number of points among the neighbors ought to be the class that the sample belongs to. Nearest Neighbor algorithm is a particular instance of k-NN where $k=1$. Consider the following figures which illustrate the sample point in the feature space and neighbors for $k = \{1, 2, \text{ and } 3\}$. [15]

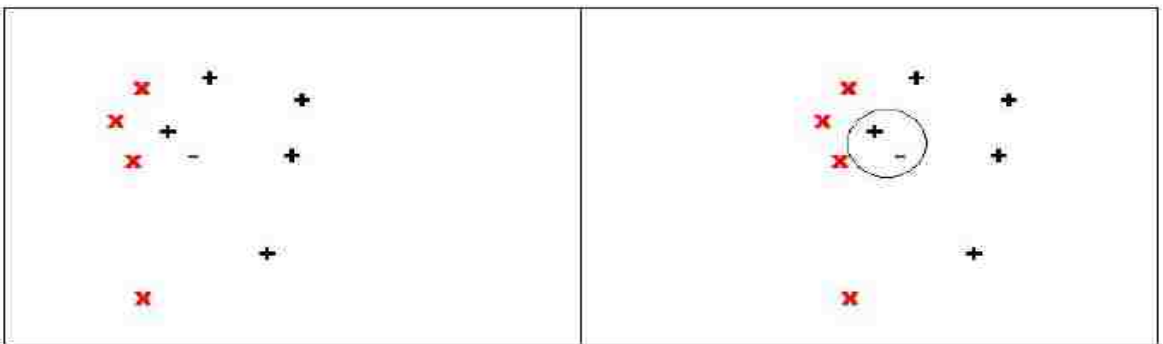


Figure 3. Feature Space and when $k=1$

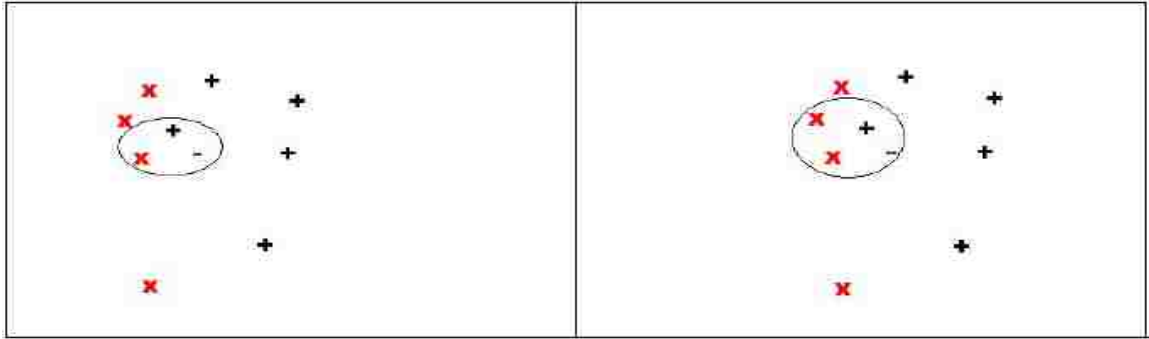


Figure 4. Feature Space when $k=2$ and $k=3$

2.5.1 Example for predicting a category using K- Nearest Neighbor Categorization.

Let us consider the same table 2.1 containing training set of data which describes the weather conditions for playing tennis. Sample data records are represented by a set of attributes such as Outlook, Temperature, Humidity, Windy and categories by attribute play. Play is represented as either “Yes” or “No”. Each sample data record is represented as a vector. There are a total of fourteen vectors out of which nine vectors belong to the category “Yes” and five vectors belongs to the category “No”.

Suppose an unknown sample $X = (\text{Rain}, \text{Hot}, \text{Normal}, \text{Weak})$ is given. Let us assume the value of $k = 3$. Now the model computes to which category X belongs by calculating Euclidean distances of 3 nearest neighbors taken from the sample data. Sample X is mapped to the category to which maximum number of its neighbors belongs to.

Our Sample $X = (\text{Rain}, \text{Hot}, \text{Normal}, \text{Weak})$

The three nearest neighbors are:

Neighbor #1: (Overcast, Hot, High, Weak, Yes)

Distance = 1.48610

Neighbor #2: (Overcast, Hot, Normal, Weak, Yes)

Distance = 1.48610

Neighbor #3: (Sunny, Hot, High, Weak, No)

Distance = 1.89960

Play Tennis = {Yes, Yes, No} / 3

$$= (1.0 + 1.0 + 0.0) / 3$$

$$= 0.6666 > 1/2$$

= Yes.

In the above example, after applying k-NN technique, we get two “Yes” neighbors and only 1 “No” neighbor. So the majority of “Yes” neighbors are more than the majority of “No”. Therefore the sample X is classified in to the category “Yes” [16].

CHAPTER 3

IMPLEMENTATION

This thesis mainly focuses on classifying the electronic documents in to their respective categories by applying two major supervised learning techniques, Naive Bayes and K- Nearest Neighbor categorizations. We then report on the accuracy and effectiveness of the classification in both the cases by calculating the recall and precision values for each of the categorization models. Java is used as the primary programming language for coding and implementation of these two categorization models.

3.1 Document Collection

For the implementation of these categorization techniques, we obtained the document collection from “Reuters-21578, Distribution 1.0 test collection”. There are a total of 21578 newswire stories from Reuters, classified in to several categories by personnel from Reuters Ltd. and Carnegie Group, Inc in 1987 and were further formatted by David D. Lewis and Peter Shoemaker in 1991 [18].

There are a total of 674 categories in Reuters – 21578 collections. They are totally divided in to five fields. Each field has several categories of document collection. The table below shows the number of fields and categories in each field for Reuters – 21578 collections.

Field	Categories
Topics	135
Organizations	56
Exchanges	39
Places	176
People	269

Table 2. Reuters – 21578 collection categories

This thesis mainly concentrates on the Topics field for our research and chose 5 categories out of 135 available. They are listed as follows.

1. Acquisition
2. Grain
3. Interest
4. Jobs and
5. Trade

There are a total of 504 documents mapped to these 5 categories. We further divide these 504 documents in to two sets, a Training set and a Test set collection consisting of 304 documents and 200 documents respectively.

In this thesis, we use the Training set collection to train both the categorization models and the Test set collection is then applied for the classification of documents in to respective categories. Training set with 304 documents is divided in to 5 categories as shown in the table 3 below.

Category	Number of documents
Acquisition	70
Grain	60
Interest Rate	70
Jobs	34
Trade	70

Table 3. Training data set collection

Documents obtained from Reuters – 21578 collections are in “Standard Generalized markup Language” format. Figure 5 shows a sample screenshot of a document in SGML format from Reuters – 21578 collections.

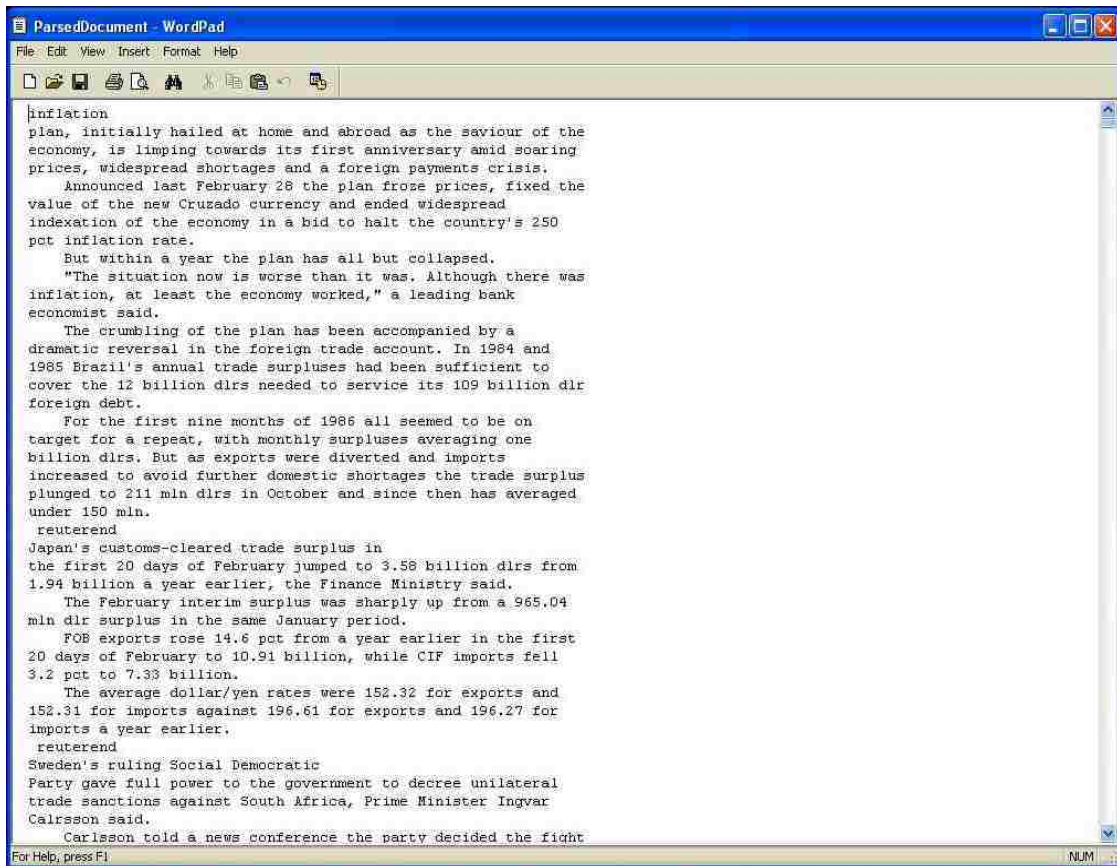


Figure 6. Screenshot of a Parsed XML Document

- Tokenization

After the XML document is parsed, the output text document looks bulky or clumsy. The parsed document is further break down in to words or terms called tokens [21]. This process is called Tokenization. In this process, all the punctuations are removed and entire text is lowercased. After tokenization, the above parsed document is tokenized in to the list of tokens as shown below.



Figure 7. Screenshot of the list of tokens after Tokenization

- Stop Words Removal

Next step in the document processing is Stop Words Removal. Stop words such as “and”, “the”, “are”, “from”, “to” etc are some of the common words which occur in most of the documents. These need to be removed because these stop words does not help to decide the category of

- Stemming

Stemming is a process of reducing the terms to their stems or variants. For example “working”, “worker”, “worked” is reduced to “work” and “crumbling”, “crumbled” is reduced to “crumb”. This process is used to reduce the computing time and space as different forms of words are stemmed in to a single word. In fact this is the main advantage of this process. The most popular stemmer in English is Martin Porter’s stemming algorithm [22]. In this thesis, we worked on porter stemmer in java to implement stemming algorithm [23]. Here is the sample screenshot of the output after stemming.



Figure 9. Screenshot of the output after Stemming

- Inverted Index

After stemming, we need to build an inverted index. An inverted index is an index data structure storing a mapping from content, such as terms or numbers to its locations in a document or a set of documents [23]. There are two types of inverted index. This thesis deals with the record level inverted index which comprises of a list of references to the documents for each term. Let's discuss a small example in which we consider 3 simple documents.

Document	Text
1	Today is Sunday
2	Sunday is a holiday
3	Tomorrow is Monday

Table 4. Example text, each line represents a document

Now, the inverted index is build for the above example text and is shown in the table 5.

Term	Documents
Today	{1}
Sunday	{1,2}
Holiday	{2}
Tomorrow	{3}
Monday	{3}

Table 5. A record level inverted index file for the text in table 4

The above inverted index is build after stemming or removing stop words “is” and “a”. These two techniques help to reduce the terms which results in faster processing. With the help of inverted index built, we can calculate document frequencies which give out the significant terms for our collection. Number of documents that contain a particular term is known as Document Frequency. For example, document frequencies for the text in table 5 are listed below.

Term	Document Frequency
today	1
sunday	2
holiday	1
tomorrow	1
monday	1

Table 6. Terms with their document frequencies

A sample screenshot of the training documents significant terms with their document frequencies is shown in the figure 10 below.

word	docfreq
such	34
congress	13
plc	18
sector	11
protection	6
quota	6
impos	6
businessmen	6
cooper	6
temporari	6
chancellor	6
lawson	6
winter	7
compar	34
work	19
central	30
feed	5
barlei	5
elig	5
spring	5
drought	5
lyng	5
recent	26
busi	24
washington	15
canada	12
until	22
infat	8
subsidiari	8
japanes	14
sen	7
trader	7
subject	10
transact	10
access	12
taiwan	11

Figure 10. Screenshot of the terms with their documents frequencies

The term “sector” has a document frequency of 11, because it occurs in 11 documents in 304 training documents. There are a total of 3608 terms obtained after building inverted index. After the collection of the most significant terms along with their document frequencies, the next task is the Dimensionality Reduction, which is the most tedious task of the text categorization. This is due to high dimensionality of feature space, i.e. total number of terms considered. There are hundreds of thousands of unique terms even for a moderate sized data collection [25]. So, our task is to reduce the number of terms which is done by dimensionality reduction. This thesis works on the document frequency thresholding, which is one of the simplest methods of dimensionality reduction to reduce terms in the collection. A predefined threshold values are assigned such that terms from the collection which are in the given range are used.

This thesis predefined the threshold values as 5 and 100, i.e. the terms whose document frequency is greater than 5 and less than 100 are considered as the significant terms and those help to categorize the document. There are 768 terms left out of 3608 terms after excluding all the terms which does not satisfy the threshold range of values.

- Weighing terms ($TF * IDF$)

Weight of a term is often used in information retrieval and text mining. It is also referred to as the Term Frequency (TF) and Inverse Document

Frequency (IDF). Weight is a measure of how important a word is to a document in a collection [26]. Term Frequency of a given document is the total count of a term that appears in the document. Hence it is defined as

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Where n_{ij} is count of the term t_i that occur in the document d_j , and the denominator is the sum of the counts of all the terms that occur in the document d_j .

The Inverse Document frequency of a given document is obtained by dividing the total count of documents by the count of documents containing the term, and then taking the logarithm of that quotient. Hence it is defined as,

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

Where, $|D|$ is the total number of documents in the collection.

$|\{d : t_i \in d\}|$ is the number of documents where the term t_i appears (that is n_{ij} is not equal to 0). If the term is not in the collection, this will lead to a division by zero. Thus, it is common to use $1 + |\{d : t_i \in d\}|$. The weight or TF*IDF value of a term is always greater than or equal to zero.

Now we define TF-IDF given by

$$(\text{tf-idf})_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$$

A sample screenshot of the training documents significant terms with their document frequencies and TF * IDF values is shown in the figure below.

word	docfreq	invdocfreq	termfreq	weight
such	34	2.1906671767900603	43	27.2784253934653
congress	13	3.152078343944685	18	27.2366440302892
plc	18	2.826665943510057	31	27.213049454245006
sector	11	3.319132428607851	16	27.199868004169822
protection	6	3.9252682321781664	6	26.398872272066647
quota	6	3.9252682321781664	6	26.398872272066647
impos	6	3.9252682321781664	8	26.398872272066647
businessmen	6	3.9252682321781664	7	26.398872272066647
cooper	6	3.9252682321781664	9	26.398872272066647
temporari	6	3.9252682321781664	6	26.398872272066647
chancellor	6	3.9252682321781664	7	26.398872272066647
lawson	6	3.9252682321781664	13	26.398872272066647
winter	7	3.7711175523509084	25	26.187971918939827
compar	34	2.1906671767900603	43	26.004369908687167
work	19	2.772588722239781	28	25.96347378706748
central	30	2.3158303197440664	33	25.913124711082315
feed	5	4.107589788972121	7	25.875719901809227
barlei	5	4.107589788972121	9	25.875719901809227
ellig	5	4.107589788972121	8	25.875719901809227
spring	5	4.107589788972121	9	25.875719901809227
drought	5	4.107589788972121	13	25.875719901809227
lyng	5	4.107589788972121	12	25.875719901809227
recent	26	2.4589311633847397	38	25.632762145898695
busi	24	2.538973871058276	35	25.626038801777664
washington	15	3.008977500304012	16	25.480668712009557
canada	12	3.2321210516182215	21	25.426949112674485
until	22	2.625985248047906	24	25.31056018911557
infiat	8	3.6375861597263857	12	25.29597437200149
subsidiari	8	3.6375861597263857	11	25.29597437200149
japanes	14	3.077970371790963	25	25.287076873295934
sen	7	3.7711175523509084	8	25.186311841320677
trader	7	3.7711175523509084	15	25.186311841320677
subject	10	3.414442608412176	10	24.94090843686681
transact	10	3.414442608412176	11	24.94090843686681
access	12	3.2321210516182215	16	24.682360736712663
taiwan	11	3.319132428607851	26	24.58061064976701

Figure 11. Screenshot of term document frequencies and weights

3.3 Algorithm Implementation – Naïve Bayes

As discussed earlier in Chapter 2, Naïve Bayes categorization is one of the simplest Bayesian categorization based on conditional independence.

The posterior probability can be given as

$$P(c) p(x|c)$$

$$P(c|x) = \frac{P(c) p(x|c)}{P(x)}$$

Where x is a feature vector and $x = (x_1, \dots, x_n)$ and c is category.

Assume that the category c_{\max} yields to the maximum value for $P(c|x)$.

The Naive Bayes classifier predicts the category c_{\max} with the largest posterior probability:

$$C_{\max} = \operatorname{argmax}_c P(c|x)$$

$$= \operatorname{argmax}_c P(c) p(x|c)$$

By implementing the above and earlier discussed process, the most of the conditional probabilities for different terms in the documents are calculated. Each conditional probability indicates the weight of the term in a given document for a particular category. The classifier predicts the category with the largest posterior probability. This ends up the Training phase of the categorization model. Here is a sample screenshot of the pseudo code of the Naïve Bayes algorithm as shown in the figure 12.

3.3.1 Test Phase

During test phase when a new document is given to the trained categorization model, it should predict the correct category of the document. A collection of test documents are shown in the table 7 below. There are a total of 200 test documents which are categorized in to 5

categories. All the preprocessing techniques should be applied to the test documents such as parsing, tokenization, stop words removal and stemming. List of significant terms are obtained. Weights of each significant term are calculated to get the feature vectors and conditional probabilities. The largest posterior probability categorizes the document into specific category.

```

TRAINMULTINOMIALNB(C, D)
1  V ← EXTRACTVOCABULARY(D)
2  N ← COUNTDOCS(D)
3  for each c ∈ C
4  do Nc ← COUNTDOCSINCLASS(D, c)
5     prior[c] ← Nc/N
6     textc ← CONCATENATETEXTOFALLDOCSINCLASS(D, c)
7     for each t ∈ V
8     do Tct ← COUNTTOKENSOFTERM(textc, t)
9     for each t ∈ V
10    do condprob[t][c] ←  $\frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$ 
11  return V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg maxc ∈ C score[c]

```

Figure 12. Pseudo code of the Naïve Bayes Algorithm [27]

Category	Total Documents
Trade	58
Grain	4
Interest	56
Acquisition	70
Jobs	12

Table 7. Test documents collection

3.4 Algorithm Implementation – k Nearest Neighbor

As discussed earlier in Chapter 2, k – nearest neighbor algorithm is the simplest among all supervised learning techniques and it classifies the objects based on the closest training examples in the feature space.

The k- nearest neighbor classifier is based on the Euclidean distance between a test sample and the specified training samples. Let \mathbf{x}_i be an input sample with p features $(x_{i1}, x_{i2}, \dots, x_{ip})$, n be the total number of input samples ($i = 1, 2, \dots, n$) and p the total number of features ($j = 1, 2, \dots, p$). The Euclidean distance between sample \mathbf{x}_i and \mathbf{x}_l ($l = 1, 2, \dots, n$) is defined as [13]

$$d(\mathbf{x}_i, \mathbf{x}_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \dots + (x_{ip} - x_{lp})^2}.$$

After loading the corpus, all the preprocessing techniques are applied. TF*IDF calculations helps us to create document vectors in feature space. This ends up the training phase for kNN algorithm.

Here is a sample screenshot of the pseudo code of the k- nearest neighbor algorithm implementation as shown in the figure 13.

```

TRAINING PHASE (D)
1. D ← PREPROCESSING(D)
2. k ← SELECT-K(D, D)
3. TRAINING(D, k)

ALGORITHM (D, D, k, d)
1. S1 ← CONVERT-TO-VECTOR(D, k, d)
2. FOR EACH di ∈ D
3.   di ← |S1 - di| / k
4. RETURN SORTED INDEX, Pi
  
```

Figure 13. Pseudo code of the kNN algorithm implementation [29]

3.4.1 Test Phase

In this thesis, we take the k value as 30. So we are calculating the Euclidean distances for a given document to its 30 nearest neighbors. A collection of test documents as shown in the table 3.8 are used for the testing phase. After all the preprocessing techniques are applied, significant terms are obtained. TF*IDF calculations helps to create document vectors in feature space. Then the Euclidean distances between the test document and specific training documents are

calculated. The majority of the same kind of nearest neighbors decides the category of the test sample.

3.5 Precision and Recall

Precision and Recall values evaluate the performance of the categorization model. Precision computes exactness where as Recall computes completeness [28]. Let TP be number of true positives, i.e. number of documents correctly labeled and as agreed by both the experts and the model. Let FP be the number of false positives, i.e. the number of documents that are wrongly categorized by the model as belonging to that category. Let FN be the number of false negatives, i.e. the number of documents which are not labeled as belonging to the category but should have been [25].

Hence, Precision is defined as

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall is defined as

$$\text{Recall} = \frac{tp}{tp + fn}$$

For various Threshold values “ θ ”, precision and recall values are calculated.

CHAPTER 4

RESULTS EVALUATION AND PROPERTIES

This chapter mainly discusses and illustrates the results obtained for the two algorithms after the implementation of the same in chapter 3. We will also evaluate the results obtained, list out the properties and limitations of both the algorithms and then we will talk about their time complexities.

4.1 Results

The effectiveness of the categorization engines based on Naive Bayes and K Nearest Neighbor methodologies or the accuracy of the results obtained after the implementation of both these algorithms are compared by calculating their standard precision and recall. As we already discussed in chapter 3, precision and recall values evaluates the performance of the categorization model. There are a total of 200 test documents, labeled for example D_1 to D_{200} , as shown in the table 7. Documents D_1 to D_{58} belongs to the category Trade, documents D_{59} to D_{62} belongs to Grain, documents D_{63} to D_{118} belongs to Interest, documents D_{119} to D_{130} belongs to Jobs and documents D_{113} to D_{200} belongs to Acquisition category. The predefined threshold values are 5 and 100, and the dimensionality is reduced to 500. Let us now have a look at the individual results of each algorithm.

4.1.1 Results – Naïve Bayes Categorization

Out of 200 test documents given to the Naïve Bayes categorization model, 190 documents were categorized correctly and the rest of 10 documents were categorized incorrectly. So, the total true positives (TP) for Naïve Bayes are 190, total false negatives (FN) are 10 and total false positives (FP) are 8. All the true positives, false negatives and false positive values for the individual categories are shown in the table 8 below.

Category	Total Document	True Positives (TP)	False Positives (FP)	False Negatives (FN)	Precision	Recall
Trade	58	54	2	4	0.96	0.93
Grain	4	4	1	0	0.80	0.92
Interest	56	52	3	4	0.94	0.92
Acquisition	70	69	2	1	0.97	0.98
Jobs	12	11	0	1	1	0.91

Table 8. Precision and Recall values of Naïve Bayes when $\theta = 5\%$

After listing all the TP, FP and FN of each category, precision and recall values are calculated based on the formulae which were discussed earlier. The standard Precision and Recall values obtained for Naïve Bayes categorization are 0.93 and 0.94 respectively. This implies that, based on Naive Bayes methodology our categorization model shows 93% exactness and 94% completeness of accuracy levels. Here is an output sample screenshot of the Naïve Bayes classifier shown in figure 14.

```

Naive Sample - WordPad
File Edit View Insert Format Help
corpus loaded - stop word removal: true; stemming: Porter Stemmer

number of documents: 304
number of labeled documents: 304

number of categories: 5

number of test documents: 200
number of labeled test documents: 200

dimensionality has been reduced to d=500

--- classification results for document C:\Documents and Settings\iditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest2.txt ---
naive bayes classifier:
category: trade      probability: 0.23026309267524267
category: interest  probability: 2.998503822717682E-9
category: acq       probability: 1.229597169344442E-17
category: grain     probability: 5.3332277433080226E-8
category: jobs      probability: 7.70548773510873E-43
-> best: trade

--- classification results for document C:\Documents and Settings\iditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest15.txt ---
naive bayes classifier:
category: trade      probability: 0.23025519555596236
category: interest  probability: 7.958769970429535E-6
category: acq       probability: 2.817275429108937E-19
category: grain     probability: 8.342547198970563E-10
category: jobs      probability: 1.2606747695862084E-9
-> best: trade

--- classification results for document C:\Documents and Settings\iditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest9.txt ---
naive bayes classifier:
category: trade      probability: 2.1976077013822286E-4
category: interest  probability: 0.23004339712459862
category: acq       probability: 2.1716928087380743E-69
category: grain     probability: 3.8040843475993195E-76
category: jobs      probability: 1.6116750334467357E-118
-> best: interest
  
```

Figure 14. Sample output of Naïve Bayes classifier

4.1.2 Results – k Nearest Neighbor Categorization

Out of 200 test documents given to the k Nearest Neighbor categorization model, 192 documents were categorized correctly and the rest of 8 documents were categorized incorrectly. So, the total true positives (TP) for k Nearest Neighbor are 192, total false negatives (FN) are 8 and total false positives (FP) are 8. All the true positives, false negatives and false positive values for the individual categories are shown in the table 9 below.

Category	Total Document	True Positives (TP)	False Positives (FP)	False Negatives (FN)	Precision	Recall
Trade	58	53	2	5	0.96	0.91
Grain	4	4	0	0	1	1
Interest	56	54	4	2	0.93	0.96
Acquisition	70	69	1	1	0.98	0.98
Jobs	12	12	1	0	0.92	1

Table 9. Precision and Recall values of kNN when $\theta = 5\%$

After listing all the TP, FP and FN of each category, precision and recall values are calculated based on the formulae which were discussed earlier. The standard Precision and Recall values obtained for k Nearest Neighbor categorization are 0.95 and 0.97 respectively. This implies that, based on k Nearest Neighbor methodology our categorization model shows 95% exactness and 97% completeness of accuracy levels. Here is an output sample screenshot of the k Nearest Neighbor classifier shown in figure 15.

```

KNN Sample - WordPad
File Edit View Insert Format Help
[Icons]
corpus loaded - stop word removal: true; stemming: Porter Stemmer

number of documents: 304
number of labeled documents: 304

number of categories: 5

number of test documents: 200
number of labeled test documents: 200

dimensionality has been reduced to d=500

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest1.txt ---

30-nearest neighbors classification:
category: trade      #docs: 23.0 distance: 3.525276130894293
category: interest  #docs: 2.0  distance: 0.18015878362960144
category: acq       #docs: 3.0  distance: 0.2727869935795447
category: grain     #docs: 2.0  distance: 0.25268707371851845
category: jobs      #docs: 0.0  distance: 0.0
-> best: trade

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest10.txt ---

30-nearest neighbors classification:
category: trade      #docs: 22.0 distance: 2.9501030013932996
category: interest  #docs: 1.0  distance: 0.09622780304281094
category: acq       #docs: 6.0  distance: 0.6330009608678737
category: grain     #docs: 1.0  distance: 0.13444120531593967
category: jobs      #docs: 0.0  distance: 0.0
-> best: trade

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest59.txt ---

30-nearest neighbors classification:
category: trade      #docs: 5.0  distance: 1.2106735809287077
category: interest  #docs: 0.0  distance: 0.0
category: acq       #docs: 1.0  distance: 0.16327981836487698
category: grain     #docs: 22.0 distance: 5.101693173277659
category: jobs      #docs: 2.0  distance: 0.22863215230559328
-> best: grain

For Help, press F1
NUM

```

Figure 15. Sample output of k Nearest Neighbor classifier

4.2 Results Evaluation

Evaluation of results includes some discussion about certain documents on how and to what category they are categorized. Whenever we give a test document to the categorization model, it should predict the correct category label of that document based on the previous training. Here we discuss 4 different cases illustrating true positives and false negatives for Naïve Bayes and k Nearest Neighbor categorizations.

Case 1: True Positives for both Naïve Bayes and kNN:

Documents for example D_1 , D_2 are true positives for both Naïve Bayes and kNN categorization models. Originally D_1 and D_2 documents belong to Trade category. After the testing phase, they are correctly categorized in to Trade category. Document D_1 , when given as an input to the Naïve Bayes model, the posterior probabilities of that document to be in all the categories are calculated. The category with the highest probability will be the tested category for that document. In this case, document D_1 has the probability of 0.230263 to be in Trade category which is highest among all the probabilities calculated. So the Naïve Bayes model categorized document D_1 in to Trade category. Document D_1 , when given as an input to the k Nearest Neighbor model, the Euclidean distances between the testing sample and all the nearest training vector documents in the feature space are calculated. Maximum number of the nearest neighbors which belong to the same category will predict the category of

the testing sample. In this case, out of 30 nearest neighbors of document D_1 found by calculating their Euclidean distances, 23 neighbors belong to Trade category, 4 neighbors belong to Interest category and 3 neighbors belong to Acquisition category. So, document D_1 is categorized into Trade category by kNN model which has highest number of neighbors from the category Trade. Here is the sample screenshot of document D_1 categorization using Naive Bayes and kNN methodologies as shown in figure 16. Similarly, we can explain the categorization of document D_2 .

```

Case 1 - WordPad
File Edit View Insert Format Help

| corpus loaded - stop word removal: true; stemming: Porter Stemmer.

number of documents: 304
number of labeled documents: 304

number of categories: 5

number of test documents: 200
number of labeled test documents: 200

dimensionality has been reduced to d=500

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest1.txt ---

naive bayes classifier:
category: trade      probability: 0.23026315769473684
category: interest  probability: 8.003048361412678E-22
category: acq       probability: 1.4102475976389994E-53
category: grain     probability: 1.8420266555942992E-38
category: jobs      probability: 9.618243409317584E-101
-> best: trade

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest1.txt ---

30-nearest neighbors classification:
category: trade      #docs: 23.0 distance: 4.46484828872976
category: interest  #docs: 4.0 distance: 0.447866851945618
category: acq       #docs: 3.0 distance: 0.3762642433394817
category: grain     #docs: 0.0 distance: 0.0
category: jobs      #docs: 0.0 distance: 0.0
-> best: trade

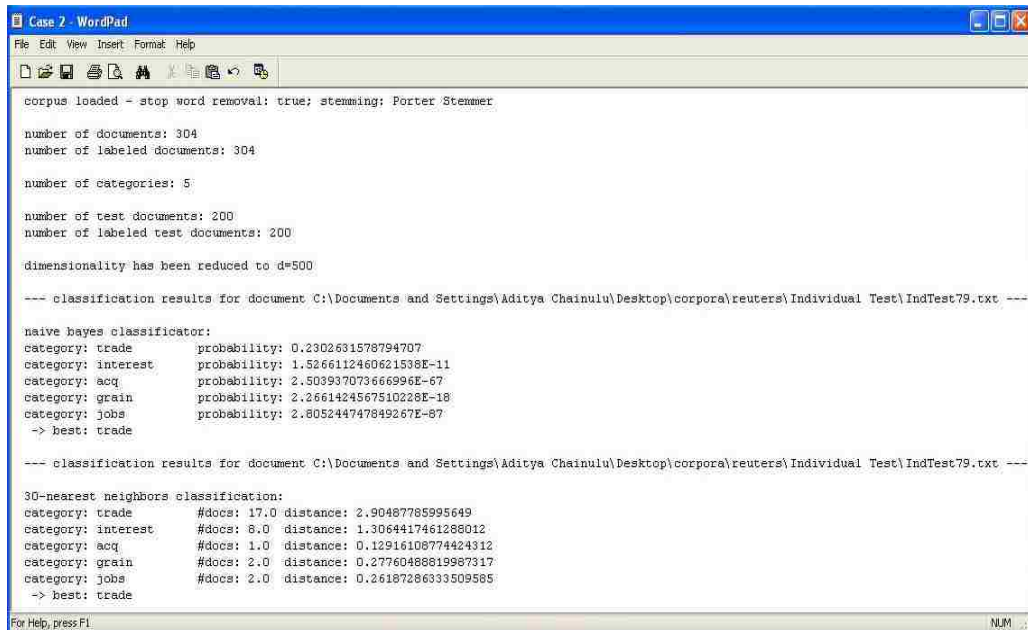
For Help, press F1
NUM

```

Figure 16. Document D_1 categorization using Naïve Bayes and kNN

Case 2: False Negatives for both Naïve Bayes and kNN.

Documents D_{50} and D_{79} are the false negatives for both Naïve bayes and kNN categorization models. Originally documents D_{50} and D_{79} belong to Trade and Interest categories respectively. After the testing phase D_{50} is wrongly categorized into Interest category and D_{79} is wrongly categorized into Trade category. Let us take document D_{79} for the evaluation. Document D_{79} , when given as an input to the Naïve Bayes model, highest probability calculated is 0.23026315 for the category Trade. But the document D_{79} originates from Interest category. Document D_{79} when given as an input to the kNN model, out of 30 nearest neighbors found by Euclidean distance, 17 neighbors are from Trade category.



```
Case 2 - WordPad
File Edit View Insert Format Help
[Icons]
corpus loaded - stop word removal: true; stemming: Porter Stemmer
number of documents: 304
number of labeled documents: 304
number of categories: 5
number of test documents: 200
number of labeled test documents: 200
dimensionality has been reduced to d=500
--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest79.txt ---
naive bayes classifier:
category: trade      probability: 0.2302631578794707
category: interest  probability: 1.5266112460621538E-11
category: acq       probability: 2.503937073666996E-67
category: grain     probability: 2.2661424567510228E-18
category: jobs      probability: 2.805244747849267E-87
-> best: trade
--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest79.txt ---
30-nearest neighbors classification:
category: trade      #docs: 17.0 distance: 2.90487785995649
category: interest  #docs: 8.0 distance: 1.3064417461288012
category: acq       #docs: 1.0 distance: 0.12916108774424312
category: grain     #docs: 2.0 distance: 0.27760488819987317
category: jobs      #docs: 2.0 distance: 0.26187286333509585
-> best: trade
For Help, press F1
```

Figure 17. Document D_{79} categorization using Naïve Bayes and kNN

In both the cases, document D_{79} is categorized in to Trade category which is incorrect. Categorization is done after reducing the dimension to 500, ignoring the terms whose frequency is less than 5 and finding the significant terms. The weight of significant terms which are left from document D_{79} after all the pre processing techniques might be almost same as the terms left from the documents belong to the Trade category. This might be one of the reasons for wrong categorization. Here is the sample screenshot of the document D_{79} categorization using Naïve Bayes and kNN methodologies as shown in figure 17. Similarly, we can explain the categorization of document D_{50} .

Case 3: True positive for Naïve Bayes and False Negative for kNN.

Document D_{107} is a true positive for Naïve Bayes model and a false negative for kNN model. D_{107} is a document originally belongs to the Interest category. When given as an input to the Naïve Bayes model, the highest posterior probability calculated is 0.228599 for Interest category which is correctly categorized. Document D_{107} when given as an input to the kNN model, out of 30 nearest neighbors found by Euclidean distance, 12 neighbors belong to Trade category. So, the document D_{107} is categorized in to Trade category by the kNN model which is incorrect. Again here the nearest neighbors to the D_{107} document vector in the feature space might be weighted approximately equal to the other document vectors from the Trade category. So the document D_{107} might

be wrongly categorized to the category Trade. Similarly, we can explain the categorization other true positives for Naïve Bayes and false negatives for kNN models. Here is the sample output screenshot of the document D_{107} categorization using both the categorization models as shown in the figure 18.

```

Case 3 - WordPad
File Edit View Insert Format Help
corpus loaded - stop word removal: true; stemming: Porter Stemmer

number of documents: 304
number of labeled documents: 304

number of categories: 5

number of test documents: 200
number of labeled test documents: 200

dimensionality has been reduced to d=500

--- classification results for document C:\Documents and Settings\aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest107.txt ---

naive bayes classifier:
category: trade      probability: 0.0016633631558650136
category: interest  probability: 0.22859979473390513
category: acq       probability: 9.480692861141968E-25
category: grain     probability: 4.257154803853753E-12
category: jobs      probability: 6.516258597520941E-45
-> best: interest

--- classification results for document C:\Documents and Settings\aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest107.txt ---
30-nearest neighbors classification:
category: trade      #docs: 12.0 distance: 1.1835494159065847
category: interest  #docs: 5.0 distance: 0.729366659283224
category: acq       #docs: 0.0 distance: 0.0
category: grain     #docs: 10.0 distance: 1.1833540093880494
category: jobs      #docs: 3.0 distance: 0.2788935167987289
-> best: trade
  
```

Figure 18. Document D_{107} categorization using Naïve Bayes and kNN

Case 4: True positive for kNN and False Negative for Naïve Bayes.

Document D_{89} is a true positive for kNN model and a false negative for Naïve Bayes model. Document D_{89} originally belongs to the Interest category. When given as an input to the kNN model, out of 30 neighbors found by calculating Euclidean distances, 15 neighbors are from the

category Interest and it is correctly categorized. But when given as an input to the Naïve Bayes model, the highest probability is calculated as 0.226088 for Acquisition category which is incorrect. Document D_{89} is incorrectly categorized in to Acquisition category instead of Interest category by the Naïve Bayes model. Again here document D_{89} after the preprocessing techniques might have been affected. It might have lost some important significant terms due to dimensionality reduction and term frequency which are important for that document to get categorized correctly. Here is the sample output screenshot of the document D_{89} categorization using both Naïve Bayes and kNN models as shown in the figure 19.

```

Case 4 - WordPad
File Edit View Insert Format Help

corpus loaded - stop word removal: true; stemming: Porter Stemmer

number of documents: 304
number of labeled documents: 304

number of categories: 5

number of test documents: 200
number of labeled test documents: 200

dimensionality has been reduced to d=500

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest89.txt ---

naive bayes classifier:
category: trade      probability: 9.999669689001123E-12
category: interest  probability: 0.004174434713831776
category: acq       probability: 0.2260881887615875
category: grain     probability: 4.5806512960805894E-7
category: jobs      probability: 1.9952892024670815E-19
-> best: acq

--- classification results for document C:\Documents and Settings\Aditya Chainulu\Desktop\corpora\reuters\Individual Test\IndTest89.txt ---

30-nearest neighbors classification:
category: trade      #docs: 3.0 distance: 0.259134884648078
category: interest  #docs: 15.0 distance: 1.598791800483527
category: acq       #docs: 5.0 distance: 0.5803164691642607
category: grain     #docs: 6.0 distance: 1.1355165393645381
category: jobs      #docs: 1.0 distance: 0.07926956451568354
-> best: interest

For Help, press F1
NUM: ..

```

Figure 19. Document D_{89} categorization using Naïve Bayes and kNN

4.3 Properties

4.3.1 Properties of Naïve Bayes Categorization

1. Based on this thesis, the performance on the Naïve Bayes Categorization engine is given as Standard Precision 93% and Recall is 94%.

2. Naïve Bayes categorization is a simple probabilistic categorization based on Conditional Independence between features.

3. Naïve Bayes classifies an unknown instance by computing the category which maximizes the posterior.

4. Naïve Bayes categorization is flexible and robust to errors. The prior and the likelihood can be updated dramatically with each training example.

5. Probabilistic hypothesis which refers that it outputs not only classification, but a probability distribution over all categories [30].

6. Naïve Bayes is very efficient and linearly proportional to the time needed just to read in all the data.

7. It is easy to implement and computation when compared with other algorithms.

8. Naïve Bayes has low variance and high bias.

9. Time Complexity:

Training Time: $O(|D|L_{ave} + |C||V|)$

The complexity of computing the parameters is $O(|C| |V|)$ because the set of parameters consists of $|C| |V|$ conditional probabilities and $|C|$ priors. The preprocessing computations on the parameters can be done in one pass through the training data. The time complexity of this component is therefore $O(|D| L_{ave})$, where $|D|$ is the number of documents and L_{ave} is the average length of a document [27].

Testing Time: $O(|C| M_{ave})$

The time complexity is $O(|C| M_{ave})$, where M_a is the average length of the test document. So, both training and testing complexities are linear in the time it takes to scan the data to have optimal time complexity.

4.3.2 Limitations of Naïve Bayes Categorization

1. The assumption of Conditional Independence is violated by the real world data.
2. Poor performance when the features are highly correlated.
3. It does not consider the frequency of the word occurrences.
4. Another problem with Naive Bayes is that the features are assumed to be independent which results, even when the words are dependent, each word contributes individually.
5. It is not capable for solving more complex classification problems.

6. Naive Bayes selects poor weights if the class has more training examples than the other. This is due to low bias that shrinks weights for the classes with few training examples.

4.3.3 Properties of k Nearest Neighbor Categorization

1. Based on this thesis, the performance of the k Nearest Neighbor Categorization engine is given as Standard Precision 95% and Recall is 97%.

2. Unlike Naïve Bayes, kNN doesn't rely on prior probabilities. It is computationally efficient and easy to learn.

3. KNN computes the similarity between a testing instance and all the nearest training examples in a collection.

4. It does not explicitly compute a generalization or category prototypes.

5. It is also called as Case-based, Instance-based, Memory-based and Lazy learning algorithm.

6. K Nearest Neighbor is the most robust alternative to find k-most similar examples and return the majority of these k instances.

7. It can work with relatively little information.

8. Nearest Neighbor method depends on the similarity or distance metric.

9. K Nearest Neighbor algorithm has the potential advantage for the problems with large number of classes.

10. Time Complexity:

Training Time: $O(|D|L_{ave})$

The time complexity of this component is therefore $O(|D|L_{ave})$, where $|D|$ is the number of documents and L_{ave} is the average length of a document. Training a kNN classifier consists of simply determining k and preprocessing documents [32].

Testing Time: $O(|D|L_{ave}M_a)$

It is linear in size of the training set as we need to compute the distance of each training document from the test document. Testing time is independent of number of classes.

4.3.4 Limitations of k Nearest Neighbor Categorization

1. Classification time is too long.
2. It is difficult to find the optimal value of k .
3. If the training data is large and complex, such target functions may reduce the speed in sorting out queries and irrelevant attributes may fool the neighbor.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis, “A Comparative Study on Text Categorization” studies the two basic methodologies of the text categorization. We implemented two categorization engines based on Naïve Bayes and k Nearest Neighbor methodology.

We discussed the background of the categorization where the two methodologies are defined and explained theoretically with an example each in Chapter 2. Training and testing documents sets are taken from Reuters 21578 document collection. All the preprocessing techniques applied on the documents and the implementation of each algorithm is explained clearly in Chapter 3. All the experimental results obtained are tabulated, compared and evaluated in Chapter 4.

We compared the effectiveness of Naive Bayes and kNN categorization engines by conducting various experiments on some document sets of Reuters 21578 collection of documents. The standard precision and recall values are obtained for both the engines using a constant threshold value. We then discussed about the time efficiencies, advantages and disadvantages of two engines.

From our entire study, we observe that the standard precision and recall values of k Nearest Neighbor categorization engine are better than Naïve Bayes engine. It has been observed that the kNN has the better

time efficiency and slightly higher performance (not statistically significant based on our study) even when complex data sets are used. Naive Bayes is simple to implement and easy learning algorithm but performs poor when the features are highly correlated and for the complex classifications. After evaluating the results in Chapter 4, we understood that the significant terms are really important for categorizing the document in to its correct category.

Text Categorization is an active area of research in the field of information retrieval and machine learning. In future, this study can be extended by implementing the categorization engines on larger datasets or probably on the entire Reuters 21578 collection of documents. Also, these two categorization models can be compared with other categorization models available and determine which model has the best performance.

BIBLIOGRAPHY

- [1]. Arturo Montejo-Raez, Thesis on 'Automated Text Categorization of documents in the High Energy Physics domain.'
<http://hera.ugr.es/tesisugr/15903837.pdf>
- [2]. Fabrizio Sebastiani, 'Text Categorization', University of Padova, Italy, 2005.
<http://nmis.isti.cnr.it/sebastiani/Publications/TM05.pdf>.
- [3]. Data Mining and its applications.
<http://dataminingwarehousing.blogspot.com/2008/10/what-is-data-mining.html>
- [4]. Wikipedia, the free Encyclopedia, Machine Learning.
http://en.wikipedia.org/wiki/Machine_learning
- [5]. Wikipedia, the free Encyclopedia, Supervised Machine Learning.
http://en.wikipedia.org/wiki/Supervised_learning
- [6]. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 'Introduction to Data mining', Chapter 4 Pearson Addison Wesley, 2005.
- [7]. James Joyce, 'Bayes Theorem' Stanford Encyclopedia of Philosophy, June 2003.
<http://plato.stanford.edu/entries/bayes-theorem>
- [8]. Tom M. Mitchell. Machine Learning, McGraw Hill, 1997.
- [9]. Fabrizio Sebastiani, 'Machine Learning in Automated Text Categorization', Italy 2002.
<http://nmis.isti.cnr.it/sebastiani/Publications/ACMCS02.pdf>.
- [10]. Wikipedia, the free Encyclopedia, Naive Bayes Classifier, 2008.
http://en.wikipedia.org/wiki/Naive_Bayesian_classification
- [11]. Thesis on 'Clustering Approaches to Text Categorization' by Hiroya Takamura
http://www.lr.pi.titech.ac.jp/~takamura/pubs/dthesis_original.pdf
- [12]. Wikipedia, the free Encyclopedia, Nearest neighbor search.
http://en.wikipedia.org/wiki/Nearest_neighbor_search#K-nearest_neighbor

- [13]. Scholarpedia, K Nearest neighbor
http://www.scholarpedia.org/article/K-nearest_neighbor
- [14]. Wikipedia, the free Encyclopedias, K Nearest neighbor Algorithm
http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm
- [15]. k-Nearest Neighbor (kNN) Algorithm
https://kiwi.ecn.purdue.edu/rhea/index.php/KNN_Algorithm_Old_Kiwi
- [16]. K- Nearest Neighbor algorithm application in JAVA
<http://paul.luminos.nl/update/408>
- [17]. Naive Bayes algorithm application in Visual Basic
http://paul.luminos.nl/documents/show_document.php?d=198
- [18]. David D. Lewis, 'Reuters 21578, Distribution 1.0 Test collection' (n.d.) www.daviddlewis.com/resources/testcollections/reuters21578/
- [19]. Text Categorization with SVM: Learning with Many Relevant Features by Thorsten Joachims
http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf
- [20]. Reading an XML file in java.
<http://java.sun.com/developer/Books/xmljava/ch03.pdf>
- [21]. Tokenization source code in java.
http://www.java.happycodings.com/Core_Java/code84.html
- [22]. Document Indexing Tutorial.
<http://www.miislita.com/information-retrieval-tutorial/indexing.html>
- [23]. Porter Stemmer Algorithm.
<http://tartarus.org/~martin/PorterStemmer/>
- [24]. Inverted Index Wikipedia.
http://en.wikipedia.org/wiki/Inverted_index
- [25]. Yiming Yang, Jan O. Pederson, 'A comparative study on feature selection in text categorization', Proceedings of the fourteenth international conference on machine learning, pages: 412-420, 1997.
- [26]. Term Frequency and Inverse Document Frequency – Wikipedia.
http://en.wikipedia.org/wiki/Term_frequency

- [27]. Naïve Bayes Text Classification.
<http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html>
- [28]. Precision and Recall – Wikipedia.
http://en.wikipedia.org/wiki/Precision_and_recall
- [29]. kNN Algorithm.
<http://nlp.stanford.edu/IR-book/html/htmledition/k-nearest-neighbor-1.html>
- [30]. Properties of Naïve Bayes Classifier.
http://homepages.inf.ed.ac.uk/keller/teaching/connectionism/lecture10_4up.pdf
- [31]. Time complexities of Naïve Bayes and kNN categorizations.
<http://www.cs.umbc.edu/~nicholas/676/MRSslides/>
- [32]. Time complexity of kNN algorithm.
<http://nlp.stanford.edu/IR-book/html/htmledition/time-complexity-and-optimality-of-knn-1.html>

VITA

Graduate College
University of Nevada, Las Vegas

Aditya Chainulu Karamcheti

Degrees:

Bachelor of Technology in Information Technology, 2007
Jawaharlal Nehru Technological University, India

Thesis Title: A Comparative Study on Text Categorization

Thesis Examination Committee:

Chairperson, Dr. Kazem Taghva, Ph.D.

Committee Member, Dr. Ajoy K. Datta, Ph.D.

Committee Member, Dr. Laxmi P. Gewali, Ph.D

Graduate College Representative, Dr. Muthukumar Venkatesan,
Ph.D.