ENHANCED ONTOLOGY-BASED TEXT CLASSIFICATION ALGORITHM FOR STRUCTURALLY ORGANIZED DOCUMENTS

SUHA SAHIB OLEIWI

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2015

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUMCollege of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pengelasan Teks (TC) merupakan asas yang penting dalam dapatan semula maklumat dan perlombongan teks. Fungsi utama TC adalah untuk menentukan kelas teks mengikut kepada jenis label yang diberi lebih awal. Kebanyakan algoritma TC menggunakan istilah dalam mewakili dokumen yang tidak mengambil kira hubungan di antara istilah tersebut. Algoritma ini mewakili dokumen dalam satu ruangan di mana setiap perkataan diandaikan menjadi satu dimensi. Hal ini menyebabkan terjadinya kedimensian tinggi yang akan memberi kesan negatif terhadap prestasi pengelasan. Objektif kajian ini adalah untuk merangka algoritma pengelasan teks dengan mewujudkan ciri vektor yang sesuai dan mengurangkan dimensi data yang akan meningkatkan ketepatan pengelasan. Kajian ini menggabungkan ontologi dan perwakilan teks untuk pengelasan dengan membangunkan lima algoritma. Algoritma pertama dan kedua iaitu Vektor Bercirikan Konsep (CFV) dan Vektor Bercirikan Struktur (SFV), akan mewujudkan ciri vektor untuk menggambarkan dokumen tersebut. Algoritma ketiga iaitu Pengelasan Teks Berasaskan Ontologi (OBTC) dibangunkan untuk mengurangkan kedimensian kumpulan–kumpulan latihan. Algoritma keempat dan kelima iaitu Pengelasan Teks_Vektor Bercirikan Konsep (CFV_TC) dan Pengelasan Teks_Vektor Bercirikan Struktur (SFV_TC) akan mengelaskan dokumen tersebut kepada kumpulan–kumpulan pengelasan yang berkaitan. Algoritma yang dicadangkan ini telah diuji menggunakan data set dari lima dokumen saintifik yang berbeza yang dimuat turun dari pelbagai perpustakaan digital dan repository. Hasil pengujian pengelasan teks daripada algoritma CFV_TC dan SFV_TC menunjukkan nilai purata kepersisan, dapatan semula, ukuran-f dan ketepatan adalah lebih baik berbanding dengan pendekatan SVM dan RSS. Kajian ini menyumbang kepada bidang penyelidikan dalam dapatan maklumat dan perlombongan teks untuk mendapatkan dokumen yang lebih relevan melalui penggunaan ontologi dalam pengelasan teks.

Kata kunci: Klasifikasi teks, Ontologi, Struktur, Dokumen berstruktur.

Abstract

Text classification (TC) is an important foundation of information retrieval and text mining. The main task of a TC is to predict the text's class according to the type of tag given in advance. Most TC algorithms used terms in representing the document which does not consider the relations among the terms. These algorithms represent documents in a space where every word is assumed to be a dimension. As a result such representations generate high dimensionality which gives a negative effect on the classification performance. The objectives of this thesis are to formulate algorithms for classifying text by creating suitable feature vector and reducing the dimension of data which will enhance the classification accuracy. This research combines the ontology and text representation for classification by developing five algorithms. The first and second algorithms namely Concept Feature Vector (CFV) and Structure Feature Vector (SFV), create feature vector to represent the document. The third algorithm is the Ontology Based Text Classification (OBTC) and is designed to reduce the dimensionality of training sets. The fourth and fifth algorithms, Concept Feature Vector_Text Classification (CFV_TC) and Structure Feature Vector_Text Classification (SFV_TC) classify the document to its related set of classes. These proposed algorithms were tested on five different scientific paper datasets downloaded from different digital libraries and repositories. Experimental obtained from the proposed algorithm, CFV_TC and SFV_TC shown better average results in terms of precision, recall, f-measure and accuracy compared against SVM and RSS approaches. The work in this study contributes to exploring the related document in information retrieval and text mining research by using ontology in TC.

Keywords: Text classification, ontology, structural, structured documents.

Acknowledgement

It gives me great pleasure to express my gratefulness to everyone who contributed in completing this thesis. It was my pleasure to study under Associate Professor Dr. Azman Yasin’s supervision. I'm so grateful for his support during the last five years. I am so grateful for his all assistants that he gave me through these years. There are no words to express my gratitude for his guidance in helping me to achieve my goal. Without his valuable support, my thesis would not have been possible. I would like to tell him that thank you so much for everything you have been done for me to reach my goal. I would like to thank my co-supervisor Dr. Nor Idayu Mahat for her progressive thinking and her open mind. Her continuous advice and significant comments helped develop my work successfully.

To my father, whose surname I proudly carry – I am forever appreciative. I want to tell him thanks for all things you supported me and make me strong to across this stage of my life. To my mother, who gave me life and prayed for me all the time, may Allah continuously bless her with good health. To my sisters Sahar and Rafah, I would like to tell them thanks for your feelings and supporting. To my dear brothers Ali and Hassanin, thanks for their love and support. To my Husband Ghassan, who gave me power and patience during the last five years of study, I thank his from the bottom of my heart. I would also like to thank my two young babies Mohammed and Zainab, without whom my goal would not have been achieved. I dedicate this work to my family. I'm so glad to study at Universiti Utara Malaysia (UUM). During my time in UUM, I have gained a lot of friends, and studying there was like being in my hometown. My sincere gratitude to all of them for all the encouragement during my study. I want to tell all of them thank you so much for everything you help me.

Table of Contents

Permission to Use ii

Abstrak iii

Abstract .... …iii

Acknowledgement v

Table of Contents vi

List of Tables ix

List of Figures x

List of Appendices xii

CHAPTER ONE INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 5

1.3 Research Objectives 8

1.4 Significant of the Study 8

1.5 Scope and Limitation 9

1.7 Thesis Organization 11

CHAPTER TWO LITERATURE REVIEW 13

2.1 Introduction 13

2.2 Text Classification 14

2.2.1 Text Classification Algorithm 14

2.2.1.1 Support Vector Machines 14

2.2.1.2 Nearest Neighbor 16

2.2.1.3 Decision Trees 17

2.2.1.4 Naïve Bayes Algorithm 19

2.2.1.5 Neural Network 20

2.2.1.6 Rocchio' Algorithm 23

2.2.2 Approaches to Create Feature Vector for Text Classification 23

2.2.2.1 Part of Speech 24

2.2.2.2 N-gram 26

2.2.2.3 Term Frequency Inverse Document Frequency 28
2.2.3 Feature Selection Method to Reduce Dimension 46
2.2.3.1 Information Gain (IG) 47
2.2.3.2 Chi2-Test (CHI) 52
2.2.3.3 Document Frequency Thresholding (DF) 56
2.2.3.4 Mutual Information (MI)...60
2.2.3.5 Ontology to Reduce the Dimension 64
2.3 Ontology...91
2.3.1 Ontology for Text Classification..93
2.3.2 Applications of Ontology 93
2.3.3 Type of Ontology 95
2.3.3.1 Classification Based on Language Expressivity and Formality 96
2.3.3.2 Classification Based on the Scope of Ontology or on the Domain Granularity..96
2.3.4 Ontology as Classifier 97
2.4 Summary 111
CHAPTER THREE RESEARCH METHODOLOGY 113
3.1 Research Framework..113
3.2 Dataset Development 115
3.2.1 Dataset Creation. 116
3.2.2 Removing Stop Words 123
3.2.3 Stemming...124
3.2.4 Types of Terms Extracted 125
3.2.5 Ontology Construction 126
3.3 Create Set of Feature 126
3.4 Create Set of Concept from Created Ontology 127
3.5 Classify Document 127
3.6 Validation...128
3.7 Evaluation Measures 129
3.8 Summary 130

CHAPTER FOUR ENHANCED ONTOLOGY BASED TEXT CLASSIFICATION ALGORITHM FOR SCIENTIFIC PAPER....................132

4.1 Introduction 132

4.2 Ontology Structure 135

4.3 Feature Vector Creation Algorithm for Text Classification.............................136

4.3.1 Proposed Concept Feature Vector (CFV) 137

4.3.2 Proposed Structure Feature Vector (SFV)..144

4.4 Ontology Based Text Classification Algorithm (OBTC)................................150

4.5 Combine Feature Vector Creation Algorithm with Text Classification Algorithm...156

4.5.1 Proposed Concept Feature Vector for Text Classification CFV_TC Algorithm...156

4.5.2 Proposed Structure Feature Vector _Text Classification (SFV_TC) Algorithm 161

4.6 Summary 167

CHAPTER FIVE RESULTS AND ANALYSIS...169

5.1 Result and Analysis 162

5.2 Summary 198

CHAPTER SIX CONCLUSION AND FUTURE WORK 199

6.1 Contributions...199

6.2 Future Works...200

REFERENCES 202

List of Tables

Table 2.1 Literature summary on feature creation in text classification 37

Table 2.2 Literature summary on reducing dimension..71

Table 2.3 Literature summary on reducing dimension ontology as classifier for text classification task...105

Table 3.1 Query for creating dataset and its classes..119

Table 3.2 Datasets for the proposed work...121

Table 5.1 Evaluation of CFV_TC 171

Table 5.2 Evaluation of SFV_TC...173

Table 5.3 Evaluation of RSS 175

Table 5.4 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of precision 178

Table 5.5 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of recall. 180

Table 5.6 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of F_measure 182

Table 5.7 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of Accuracy.Evaluation of SFV_TC 184

Table 5.8 Evaluation of SFV_TC...186

Table 5.9 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of feature size 188

Table 5.10 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of precision 190

Table 5.11 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of recall...192

Table 5.12 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of F_Measure...194

Table 5.13 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of Accuracy 197

List of Figures

Figure 3.1 Proposed Research Architecture 114

Figure 3.2 Scientific paper structures 117

Figure 3.3 Confusion Matrix for Text Classification Evaluation Classification of RSS feed news items using ontology 130

Figure 4.1 General Architecture of the proposed work 133

Figure 4.2 The proposed text classification framework 134

Figure 4.3 Ontology for computer science domain (Classification concept) 135

Figure 4.4 Ontology for classification concept from Computer Science ontology Science Domain (RDF) 136

Figure 4.5 The proposed Concept Feature Vector (CFV) Algorithm 139

Figure 4.6 The proposed Structure Feature Vector SFV algorithm 146

Figure 4.7 The proposed text classification Ontology Based Text Classification Algorithm. 153

Figure 4.8 The proposed CFV_TC algorithm 159

Figure 4.9 The proposed SFV_TC algorithm… 164

Figure 5.1 The evaluation of the first proposed algorithm CFV_TC 172

Figure 5.2 The evaluation of the second proposed algorithm SVF_TC. 174

Figure 5.3 The evaluation of the RSS classification algorithm 176

Figure 5.4 The comparison between RSS, CFV_TC, and SF_TC algorithm in terms of precision 179

Figure 5.5 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of recall 181

Figure 5.6 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of F_measure. 183

Figure 5.7 The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of Accuracy 185

Figure 5.8 Results of the SVM Classification. 187

Figure 5.9 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of precision. 189

Figure 5.10 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of recall. 191

Figure 5.11 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of F_Measured. 193

Figure 5.12 The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of Accuracy..194

List of Appendices

Appendix A Samples of Data 228

Appendix B Output of Concept Feature Vector creation _text classification (CFV_TC) 234

Appendix C Output of Structure Feature Vector creation _text classification (SFV_TC)... 243

CHAPTER ONE
INTRODUCTION

1.1 Background

Text categorization is the task of assigning predefined categories to free-text documents. It can provide conceptual views of document collections and has important applications in the real world (Kaur & Jyoti, 2013). In the recent years, TC has gained tremendous attention and rapidly developed. Today, TC is widely used in applications such as “automatic indexing” for "Boolean information retrieval" systems, "document organization", "text filtering", and "word-sense disambiguation" (Rafi, et al, 2012; Shimodaira, 2014).

According to (Calvo, Lee, & Li, 2006), TC reduces the time required to classify vast amounts of documents without the need for experts. While TC methods may vary in terms of accuracy and computation efficiency, TC methods generally save time and expense required to perform TC. Classification algorithms can be used to extract models describing important data classes.

There are several algorithms used to classify text such as "k-nearest neighbors" (KNN), "naïve Bayes" (NB), and "Support Vector Machines" (SVM) (Patra & Singh, 2013). To build a classifier in text classification there is need to define set of example as training set. These sets are labelled with pre-defined classes (Li & Liu, 2003). Often, a data set sample contains both positive and negative examples of a concept to induce a classification rule use machine learning algorithm (Aytug, Boylu, & Koehler, 2006).

For training classifier in classification system, it time consuming to label a large amount of radical content "positive examples" and non-radical content "negative examples". This approach is known as supervised text classification (Yang & Chen, 2012). A classifier is then built by applying classification algorithm the training data. According to (Li & Liu, 2003), this approach to building classifiers is called supervised learning because the training documents all have its predefined classes.

Before the classification model automatically classifies the text, all the text should be represented. Traditional methods represent text as a point in the m-dimensional real-valued feature space, where m is the number of features or dimensions, which is a Vector Space Model (VSM) (Zuo, Wan, & Ye, 2011; Salton, Wong, & Yang, 1975). The VSM approach was presented by Gerard Salton and his group (Salton, 1971) (Salton et al, 1975) for "the SMART information retrieval system". SMART established many of the concepts that are used in modern search engines (Manning, Raghavan, & Schütze, 2008; Turney & Pantel, 2010). It is also known as the “bag of words" model and it is widely used for text representation. It is simple and easily implemented, also efficiency and effects of this model are quite good (Xiao, Shi, Liu, & Lv, 2010). In VSM, every document is represented as a vector of features where each feature corresponds to a unique word from the documents. After that methods for weighting this features are used to give the value for each feature to specify the significance of each feature (Yuan, Ouyang, & Xiong, 2013).

"Term Frequency – Inverse Document Frequency" (TF-IDF) is the common technique used to calculate the weight of these terms (Soucy & Mineau, 2005). VSM assumes every unique term from documents can be represented by each dimension of the vector. This method also has the assumption that each term in the document is independent (Wibowo, Handojo, & Halim, 2011). Therefore, the VSM ignores the semantic relations and the order of items which cause the loss of semantic. The amount of information expressed by VSM model exceeds a top limit (Xiao et al, 2010).

To deal with this problem, some researchers attempted to construct complex feature unit like base forms of morphological categories, phrases, word senses (Kehagias, Petridis, Kaburlasos, & Fragkou, 2001) and multi-word (Zhanga, Yoshidaa, & Tangb, 2008) to substitute the single words by "Natural Language Processing" (NLP) approach. Although these NLP based method apparently carry more information than bag of words, they can only gain small or no improvement for TC tasks (Yuan, Ouyang, & Xiong, 2013; Moschitti & Basili, 2004).

However, the high-dimensionality of the VSM is a big hurdle in applying many sophisticated learning algorithms in text classification (Yang & Liu, 2011). Therefore, dimensionality reduction has been a research hotspot in recent years (Yunhe, Yuan, & Chao, 2013). The feature selection is a simple and efficient methods widely used in dimensionality reduction. So far, there are many feature selection algorithms proposed by various literatures based on the theory of information and statistics, such as "Information Gain" (IG), "2-statistic" (CHI), "Improved Gini Index", and measure using "Poisson distribution" (XP2) (Liu & Yang, 2011). Multiclass or multinomial classification is the problem of classifying instances into more than two classes. While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies (Pote & Akarte, 2014).

More precisely, feature selection is "the process of selecting a subset of terms occurring in the training set and using only this subset in text classification (Manning, Raghavan, & Schütze, 2008). Although feature selection helps in having more efficient and accurate classification, the selected features are biased and limited to the training set (Mohaqeqi, Soltanpoor, & Shakery, 2009). Classification algorithms, combined with ontology are used to solve the semantic relation in text representation which are to be classified using concepts and semantic relations (Sajgal´ık, Barla, & Bielikov´a, 2013; Agarwal, Singhal, & Bedi, 2012; Zhan & Chen, 2012; Wu & Liu, 2009). Another work done is to construct and enrich the ontology using traditional classification algorithm by extracting more related concepts to a specific domain (Luong, Gauch, & Wang, 2009). Another work done is to construct and enrich the ontology using this classification algorithm by extracting more relating concepts to a specific domain (Luong, Gauch, & Wang, 2009). Others try to use ontology to reduce the dimension of training set (Elberrichi et al, 2012).

Ontologies can be used to enhance information retrieval and rendition greatly. They also enhance the machine readability and understand ability of web documents. Ontologies and ontology representation languages respectively, can efficiently deduce and describe relationships among information from their metadata in ventures like the Semantic Web.

In its general meaning, ontology is "the study or concern about what kinds of things exist - what entities there are in the Universe". An ontology is the working model of entities and interactions in some particular domain of knowledge or practices, such as electronic commerce or planning. It is a set of concepts that are selected in some way in order to create an agreed upon vocabulary for exchanging information. (Raghunathan, 2003). An ontology is also "defined as shared formal conceptualization of a particular domain". In other words, “An ontology is a specification of a conceptualization” (Gruber, 1993).

1.2 Problem Statement

Document representation is one of the key components that determines the effectiveness of text classification tasks. Traditional document representation methods typically adopt the general "bag of word" approach as the basic document representation (Achananuparp, Zhou, Hu, & Zhang, 2008). It used TF.IDF in representing the importance of terms in document. These models used terms in representing the document which does not consider the relations among the terms (Agarwal et al, 2012). Therefore, the results are less precise and noisy due to the problem of recognizing synonyms and polynyms which is critical for improving the precision and recall of semantic search (Mousavi et al, 2013).

Many models try to replace the terms with concepts from ontology to solve the problem of relations between terms in text classification (Ajgalik et al, 2013; Agarwal et al, 2012; Zhan & Chen, 2012; Wu & Liu, 2009). These studies used concept frequency to represent these documents. According to (Zhanguo, Jing, /Xiangyi, Yanqin, & Liang, 2011), in a scientific literature the structure of the document cannot be ignored in document representation. However approaches which used concept frequency for document classification ignore this important feature. The reason for using a structure in scientific document is that it is a mean the used of efficiently communicating findings to the broad community in a uniform manner (Huth et al, 1994). A new way to overcome this problem is by finding the weight of the concepts itself in term of document representation and ontologies using semantic relations.

Another problem in text classification is that it involves high dimensionality of document (Agarwal et al, 2012). Most of the classification algorithms represent documents in a space in which every unique word is assumed to be a dimension. To improve the efficiency of classification algorithms, the dimensionality of this space should be reduced (Yan et al, 2006). The solutions to this problem are generally known as feature selection method. Feature Selection is the process of selecting a subset of terms occurring in the training set and using only this subset in text

(*Manning et al, 2008). Many work try to use ontology with traditional classification algorithm to reduce the dimensionality of training set by replacing terms with set of concepts. According to (Dollah & Aono, 2011) replacing terms with concepts is a substantial dimensionality reduction of document's feature space. While Ajgalik et al, (2013) used concepts from ontology to create feature set instead of terms to accordingly classify document using SVM classification algorithm. It is possible that when input features to document classification are from unknown sources, the effect will produce negative results since training and testing set are not from the same source (Mohsenzadeh et al, 2010; Mohaqeqi et al, 2009).

In a bid to overcome the limitation of working with high dimensional data in text classification, an algorithm that includes feature reduction using document representation is proposed. There are two ways to enhance the classification performance. First, the utilization of concepts in the real documents as feature vector. Second, is the selection of suitable concepts located on ontology instead of training example. Based on the problems that deals with classification poblems described in the previous section, the study tries to answer the questions as below:

- How to create a suitable feature vector depending on document structure for text classification?
- How to solve the high dimensionality problem for text classification without effect on the accuracy?

- How to combine Ontology concepts with document structure in text classification?
- Can the proposed algorithms enhance the performance of text classification in term of precision, recall, accuracy and feature size?

1.3 Research Objectives

The main objective of this research is to develop an enhanced ontology- based text classification algorithms for structurally organized documents.

The following specific research objectives are to be fulfilled:

- To design feature vector creation algorithms that can handle the problem of semantic relation between terms and structure of a document.
- To design classification algorithm which that deals with high dimension data using ontology concepts.
- To evaluate the performance of the proposed enhanced text classification algorithm.

1.4 Significance of the Study

This section presents a quick description about the significance of the research where benefits that can be added to the document classification society are discussed. Improving ontology-based text classification methods, especially structured document, have significant positive effects on text classification performance. The potential benefits of this research can be summarized as follows:

Many benefits can be get from this research, the first one is to the wide range of text classification applications. While the second will be the direct beneficiary in term organizing the digital library in semantic way using ontology, where the user can access the category directly. By using concept as class can save time in query application to retrieve the document for user access. And also classify the document to set of classes can help the user to direct access the digital libraries such as scientific paper. The performance of the proposed methods are improved by applying the concept of ontology and its semantic relation where the classification performance becomes more semantic and accurate than before.

The outcome of this study is a novel text classification algorithm that can increase the performance with the semantic feature and help end users to directly access with accurate values.

1.5 Scope and Limitations

This thesis focuses on using ontology for a computer science domain and its property (concepts and relations) for classifying text to set of class on specific domains by create set of concept from ontology to reduce the dimensions of data without the need to create the examples for each class.

Ontology (concepts and relations) is combined with document representation to create sets of features for document representation to enhance the precision and recall because create feature is important and effects directly on the performance of classification. Another enhancement is enhancing classification by detecting new classes semantically. Conditional property used to do classification because it is efficient and effective in calculating the similarity between two samples without depending on the frequencies of terms.

All experiments were carried out from the scientific paper downloaded from five different datasets IEEE, Google Scholar, ACM, World Scientific, and CiteSeerx where the domain is computer science. Several studies have been conducted on TC using scientific papers (Dollah & Aono, 2011) and (Nuipian et al, 2011) used datasets with abstracts while (Zhanguo et al, 2011) used the whole document. The DT_TREE model by (Rizvi & Wang, 2010) downloads different datasets from the Internet. The ontology is created manually from a Wikipedia. The proposed algorithm were compared with (Agarwal et al, 2012) and (Dollah & Aono, 2011).

This thesis deals with solving two problems of text classification. These are: creating suitable feature vector and solve the high dimensionality problem caused by training set.

1.6 Thesis Organization

This thesis has Six chapters, including the introductory chapter, which covers the background information related to the problem that this thesis attempts to solve.

Chapter Two is a literature review of related studies on definition of text classification. The second part of this chapter focuses on traditional text classification and its approaches. Meanwhile, the third part presents the studies that used ontology to handle the semantic problem and high dimensionality problems in text classification.

Chapter Three presents the methodology that was used in conducting this research. It is divided into six sections. The first part outlines the research framework that has been used in this thesis. The second section presents the dataset' development that has been used in this thesis which in turn is further divided into four subsections. The first subsection presents the loaded dataset that has been used; the second subsection is related to cleaning dataset, while the third subsection presents the processing that is needed for some data set. The fourth subsection presents the type of term to be used. Third section of this chapter, the methodology employed to create a suitable feature vector is discussed. Fourth section of this chapter, the methodology employed to reduce the dimensionality of training set is discussed. The fifth section presents the methodology that has been used to classify document to set of class, while sixth subsection presents the evaluation of the proposed work in this thesis. Finally the summary of this chapter is given in the seventh section.

Chapter Four presents the algorithms for the proposed work. It has been divided into five subsections. The first subsection presents the introduction. The second subsection presents the ontology description. Third subsection presents the algorithms proposed to create a feature vector. The fourth subsection presents the algorithm to solve the Curses of Dimensionality and the fifth subsection presents the algorithm proposed to classify the text into set of class.

Chapter Five presents the result and analysis of the proposed algorithms. It is divided into two subsections. The first, subsection presents and compares the result in terms of precision, recall, F-measure and accuracy. While the second part compares the result in terms of feature size.

Finally, chapter Six includes the conclusion of this study, while the proposed future work is also given in the same chapter to open new ideas that can be developed and applied in many to get the benefit from this study.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

Text classification (TC) "is the process of assigning documents to one or more predefined classes based on their contents". Classification process is based on assigning set of document previously to different classes for assigning new document. Classification learning is called "supervised" when some external mechanisms provide information on the correct classification during the training of the algorithm. TC dates back to the early ’60s, but only in the early ’90s it became a major subfield of the information systems discipline, this is because of increased applicative interest and to the availability of more powerful hardware. Now, TC is widely used in different fields such as "document indexing" based on a controlled vocabulary, to "document filtering", "automated metadata generation", "word sense disambiguation" and "population of hierarchical catalogue of Web resources. In the ’90s, this approach has lost its popularity in favour of the "machine learning" (ML) paradigm. The ML is" a general inductive process which automatically builds an automatic text classifier by learning, from a set of pre classified documents, the characteristics of the classes of interest" (Sebastiani, 2002). The advantages of this approach are an accuracy comparable to that achieved by human experts, and a considerable savings in terms of expert manpower since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories.

2.2 Text Classification

There are three phases involved when constructing a text classifier. In the first phase, a term selection is performed whereby the related term is identified. This is followed by the second phase where the weight for each selected term in the document is computed. After that in phase three a classifier is constructed using the terms selected from the training set. This phase is also known as classifier learning and it involves supervised training in which information from each class is used (Debole & Sebastiani, 2003).

In the following subsections, text classification approach, document representation approach for text classification and feature selection method for high dimensionality problem will be explained.

2.2.1 Text Classification Algorithm

There is a wide variety of text classification algorithms ranging from simple and effective to more computationally demanding (Dhillon, Mallela, & Kumar, 2003). A number of algorithms has been constructed for text classification, including SVM, KNN , NB, Decision Tree , NN and Rocchio' Algorithm.

2.2.1.1 Support Vector Machine

A Support Vector Machine is a supervised classification algorithm that has been used widely and effectively to classify items into predefined classes. To learn a classifier, large number of features should be processed. SVM algorithm try to handle this problem by using over fitting protection. In machine learning, over fitting occurs when a statistical model describes random error or noise instead of the underlying relationship. SVM is a ML technique which is based on "structural risk minimization principle" from the "computational learning theory". Introduced by Lapnik in 1979, the SVM splits the data from training set into two classes and making decision depending on the "Support Vectors" where the effective elements are selected from the training set (He, Tan, & Tan, 2000).

SVMs belong to the general category of kernel methods. A kernel method is an algorithm that depends on the data only through dot-products. When this is the case, the dot product can be replaced by a kernel function which computes a dot product in some possibly high dimensional feature space. This has two advantages: First, the ability to generate non-linear decision boundaries using methods designed for linear classifiers. Second, the use of kernel functions allows the user to apply a classifier to data that have no obvious fixed-dimensional vector space representation (Hur & Weston, 2010). In addition, It uses the kernel trick, so can build in expert knowledge about the problem via engineering the kernel. An SVM is defined by a convex optimization problem (no local minima) for which there are efficient methods. Lastly, it is an approximation to a bound on the test error rate, and there is a substantial body of theory behind it which suggests it should be a good idea. The disadvantage of SVM is that it is time consuming process, especially when training a large corpus (Chirawichitchai et al, 2009).

2.2.1.2 Nearest Neighbor

The KNN is an algorithm which classifies objects based on the distance between objects. Famed for its simplicity, it is a widely employed technique for text classification. The KNN performs well even when multi-categorized documents are used. However, under large training examples, the KNN requires much longer time to perform text classifications. To address that, the KNN should select objects from training set by calculating the distance between objects from training examples (Pawar & Gawande, 2012).

The intuition underlying Nearest Neighbour Classification is quite straightforward, examples are classified based on the class of their nearest neighbours. It is often useful to take more than one neighbour into account so the technique is more commonly referred to as k-Nearest Neighbour (k-NN) Classification where k nearest neighbours are used in determining the class. Since the training examples are needed at run-time, i.e. they need to be in memory at run-time, it is sometimes also called Memory-Based Classification. Because induction is delayed to run time, it is considered a Lazy Learning technique (Cunningham & Delany, 2007). The advantages of k-NN are variable sized hypothesis space, learning is extremely efficient and can be online, and tree can be expensive and very flexible decision boundaries. The disadvantages of this classifier are the computation complexity, memory limitation, and being a supervised learning, it is a lazy algorithm that runs slowly and easily fooled by irrelevant result (Bhatia & Vandana 2010).

2.2.1.3 Decision Trees

Decision Tree is another technique used in text classification. A decision tree is a classifier expressed as a recursive partition of the instance space. A Decision Tree consists of internal nodes. For each node on Decision Tree, set of terms is defined. Branches departing from these internal nodes are assigned with terms from text document while the leaves label the classes. The Decision Tree is constructed using the "divide and conquer" strategy where set of cases is associated with each node. Under this strategy, all training examples poses the same label. If a training example has a different label then a term will be selected from the pooled classes of documents which carries the same values. This class is then placed on a separate sub-tree (Pawar, & Gawande 2012).

"Decision Trees" are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The main problem in DT classification algorithm is how to construct the optimal classifier. Generally, may DT classifier can be built from a set of features. In classification task when the size of search space is exponential, the accuracy of some trees are more precise than the others, and it is computationally infeasible to find the optimal tree. However, different algorithms have been developed to construct a reasonably accurate, "albeit suboptimal", "decision tree" in a practical time and efficiently. These algorithms usually use a "greedy strategy" that develops a "decision tree" by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example, “Hunt's algorithm”, “ID3”, “C4.5”, “CART”, and “SPRINT” are “greedy decision tree induction” algorithms.

DTs are easy to interpret and understand. DT can be imagined and need a few data training. Other classification methods need data normalization, there is need to create dummy variables and removing blank values. However, this technique does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. More advantage is using a “white box” model. When the situation is visible in a model, the interpretation is explained by “Boolean logic” easily. While, in a “black box” model, “Artificial Neural Network”, it is difficult to explain the result. Moreover, it is possible by using statistical test validate this model. That makes it easy to account for the consistency of the model. Also performs well even if its assumptions are somewhat violated by the true model from which the data were generated. While the disadvantages of DTs include: the creation of over-complex trees that do not generalized the data well. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. Consequently, practical decision-tree learning algorithms that are based on heuristic algorithms such as the greedy algorithm cannot guarantee to return the globally optimal decision (Dietterich, 2000).

2.2.1.4 Naïve Bayes Algorithm (NB)

"Naïve Bayes" classifier is a simple probabilistic classifier based on applying Baye’s Theorem with strong independence assumptions". In this algorithm, "posterior probability" is calculated to each document belonging to different classes. The document is classified to class which has the highest "posterior probability". This model is known as independent feature model, because present of some feature will not effect on the other features. (Aggarwal, Zhai, & Xiang, 2012).

A “naive Bayes” classifier suggest that the presence (or absence) of a specific feature of a class is independent to the presence (or absence) of other feature. “naive Bayes” classifiers is efficiently trained in a “supervised learning” setting depending on the probability model. In many practical applications, parameter estimation for “naive Bayes” models uses the method of maximum likelihood; in other words, one can work with the “naive Bayes” model without believing in “Bayesian probability” or using any “Bayesian” methods.

A “Bayes classifier” is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. Analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification methods showed that Bayes classification is outperformed by more current approaches, such as boosted trees or random forests (Korada, Kumar, & Deekshitulu, 2012).

Advantages of the Naive Bayes Classier can learn 'soft' nonlinear concepts, setting it apart from logistic regression. With a small data set, Naive Bayes will converge more quickly to a solution than logistic regression. Naive Bayes can work with both discrete and continuous features together in the same data set. Because only variances are considered when selecting appropriate features, the covariance matrix need not be considered. The Naive Bayes Assumption allows the classification problem to be reduced from O (2n) to O (n). Despite of its many advantages, a main disadvantage of using the “Naïve Bayes” classifier is that the data from the real-world may not all the time accepts the independence assumption among attributes. This strong hypothesis could make the prediction accuracy of the “Naïve Bayes” classifier highly critical to the correlated attributes.

A major limitation of using the Naïve Bayes classifier is that the real-world data may not always satisfy the independence assumption among attributes. This strong assumption could make the prediction accuracy of the Naïve Bayes classifier highly sensitive to the correlated attributes (Holloran, 2009).

2.2.1.5 Neural Network (NN)

"Neural networks" is an as important text classification tool. In recent years, numerous research activities in neural classification have implemented and shown NN are promising alternatives to conventional classification methods. The primary advantage of NN is "its data-driven self-adaptive methods" which allows them to adjust accordingly to data without the needs for explicit specification of functional or distributional form for the underlying model. Neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Also, neural networks are able to estimate the posterior probabilities, which provides the basis for establishing classification rule and performing statistical analysis. Due to its nonlinear and flexible characteristics, neural networks are widely used to model real world complex relationships. The effectiveness of NN has been tested, and today NN is applied to a variety of industrial, business and science applications (Zhang, 2000).

NN have developed to be a vital tool for classification task. The modern research activities in NN classification have proven that NNs are a committed alternative to different traditional classification methods (Negoita & Mircea, 2004). Since any classification procedure seeks a functional relationship between the group membership and the attributes of the object, accurate identification of this underlying function is doubtlessly important.

NN process records one at a time, and "learn" by comparing their classification of the record (which, at the outset, is largely arbitrary) with the known actual classification of the record. The input layer is composed not of full neurons, but rather consists simply of the values in a data record, that constitute inputs to the next layer of neurons. The next layer is called a hidden layer; there may be several hidden layers. The final layer is the output layer, where there is one node for each class. It is thus possible to compare the network's calculated values for the output nodes to these "correct" values, and calculate an error term for each node. These error terms are then used to adjust the weights in the hidden layers so that, hopefully, the next time around the output values will be closer to the "correct" values (Guhan & Selvirajan, 2014).

In “neural networks” the main thing is the process of learning iteratively where the data are presented to the classifier one at a time. Moreever, the weights correlating with these data are changed each time. NN learning is also denoted as "connectionist learning," which comes from the connections between the neurons. The main advantages of NN is the “high tolerance to noisy data”, also classify the pattern which have not been trained. The “back-propagation” algorithm is the most known NN algorithm which is suggested in the 1980's (Han, Kamber, Pei, 2013).

NN is nonlinear model that is easy to use and understand compared to statistical methods. NN is non-parametric model while most of statistical methods are parametric model that need higher background of statistic. However, ANN is a black box learning approach that is it cannot interpreter relationship between input and output and cannot deal with uncertainties. In addition, some networks never learn. This could be because the input data do not contain the specific information from which the desired output is derived. Networks also don't converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a validation set.

2.2.1.6 Rocchio' Algorithm

"Rocchio's Algorithm is a classic method for document routing or filtering". It adapted Relevance feedback methods for text categorization. This algorithm builds a prototype vector for each class and classifies a document vector by calculating the distance between the document vector and the prototype vectors. To calculate the distances between vectors, the dot product or Jaccard similarity measure is used. The prototype vector for the class on the other hand is computed as the average vector over all training document vectors that belong to this class (Munteanu, 2007).

Prototype vector does not need to be averaged or otherwise normalized for length since cosine similarity is insensitive to vector length. Classification is based on similarity to class prototypes. The advantage of this method is that it only considers the effect of feedback on the unseen relevant documents but the main disadvantage is that the feedback results are not comparable with the original ranking. This is because the residual collection has fewer documents than the original collection (Kruse, Rosner, & Nakhaeizadeh, 2001).

2.2.2 Approaches to Create Feature Vector for Text Classification

Document representation is one of the key components for determining the text classification effectiveness (Achananuparp et al, 2008). In traditional document representation, "Bag of Word" (BOW) approach is adopted to represent the document as vector of terms. Some study weights these terms by adopting TF.IDF, N_gram, Part of Speech (POS) or the structure of the document.

2.2.2.1 Part of Speech (POS)

Part of speech is a key term in any book about grammar, and even any dictionary, for that matter. Common examples of a word's part of speech include noun, verb, adjective, and so on. A definition of a part of speech is a class of words based on the word's function, the way it works in a sentence. Any of the classes of words of a given language to which a word can be assigned: different kinds of grammar have different criteria for classifying words, as form, function or meaning, or combinations of these. In traditional English grammar, patterned after Latin grammar, the parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection.

The difficult of POS affected considerably difficulty in common linguistics and in the analysis of separate languages. However it has been studied for more than 200 years, the principles for defining POS still not work. Conventionally grammar provided a semantic description of POS, depending on the meaning only. Nevertheless, depending on using the meaning only is no a consistent measure for defining POS since the same meaning comes from different POS and vice versa.

Many works try to use the POS in text classification models, a recent attempt for text classification proposed by (Celik & Gungor, 2013), focus on the contribution of semantic features using POS rather than feature selection and machine learning techniques. In this work, the researchers consider noun, verb, adjective and adverb, thus POS of a term will be both analyzed and used in this work. It uses a lexiconbased part-of-speech tagger. Given a sentence, the tagger returns the tokenized terms with part-of-speech information. "Word sense disambiguation" (WSD) is used to solve the problem of synset. The word sense disambiguation method is evaluated by using two different relations, hypernyms and topics. It uses SVM as a classifier. It performs experiments on five standard datasets, widely used in text classification research. "20 News group", "Classic 37 Sectors WebKB 5396 Reuters-21578". The results show that using POS tagging without raw features rarely gives better results, where Raw are the base line for this study which does not consider any semantic information at all.

In another study conducted by Che &Teng (2009), the authors proposed a method where the POS is taken as the concept of the term to avoid disambiguation errors. Under this method, a concept is used for concept-based representation while words are used for word-based representation. Both representations are then combined in a C-Tree from the HowNet dictionary. In this study, TC Chinese corpus of "Fudan Univ" used to create the training set. To evaluate the performance of this study, different selection method and NB and SVM classification algorithms are used. In their study, the Chi (*x2*) feature selection method gives the lowest value in the experiment comparing with the other methods.

A study by (Xia, Chai, & Wang, 2012) suggest that terms appeared in the title and other parts are given different weight. An enhancement of SVM is proposed. The method named "Title Vector based SVM" (TV-SVM) is used to classify web page documents. The corpus collected from portal sites by the VIPS module is used for creating document. To evaluate this study, training set is created for testing the TV-SVM by using "optimal Gaussian ARD kernel adaptation". To reduce the dimensions of feature set Chi (*x2*) is used. The experimental results show that the performance of TV-SVM is 91.6% comparing to the SVM.

2.2.2.2 N-gram

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. An N-gram model uses the previous n-1 words to predict the next one. N-gram model can be trained by counting and normalizing. Normalizing means dividing by some total count so resulting probabilities fall legally between 0 and 1. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram". N-Grams with N > 3 are not practical, because the number of parameters of an N-gram is V N, where V is the size of the vocabulary.

Advantages of n-gram are encode not just keywords, but also word ordering, automatically, models are not biased by hand coded lists of words, but are completely dependent on real data and learning features of each affect type is relatively fast and easy. While the disadvantages are: Long range dependencies are not captured, Dependent on having a corpus of data to train from sparse data for low frequency affect tags adversely affects the quality of the n-gram model (Singh, Vishal Goyal, & Rani, 2014).

Many models used n-gran to create feature vector for text classification task, Luo et al, (2011) employed N-gram frequency feature when performing TC on Chinese characters. The impact of various assumptions on N-gram was investigated and the authors have proposed a hybrid of N-gram frequency features to perform TC. The technique proposed uses SVM light package, and three different SVM kernel functions namely, "Linear Kernel", "Polynomial Kernel" and "Radial Basis Function" are adopted. In the experiment, the TanCorpV1.0 is used. TanCorp1.0 is a corpus specifically used for Chinese TC. The results show that new method improved classification performance when using part-of-speech approach.

Nguyen, Gao, & Andreae, (2011) developed a new text representation based on phonological study of Vietnamese syllables to capture the sound information. Each syllable is broken into three parts and each part forms a gram in the N-gram representation. In this study, tests were conducted on the representation on four non-topic based classification tasks including Vietnamese language identification, Ancient /Modern Vietnamese identification, author identification and poem identification. The tests were conducted using multiple classifiers including NB, K-NN and SVM. It built the datasets by downloading web pages and manually labeling them into classes. The experiments conducted proved that the proposed phoneme-based representations are helpful for categorization. Also, it is suitable for some non-topic based text classification problems such as language identification, author identification, and poem identification. The results show a significant improvement in terms of effectiveness and efficiency compared to the traditional syllable based representation in most cases.

Data sets that share many common keywords between classes on classification affects the performance on TC. Therefore, a novel term weighing scheme, named probrf was proposed by Ping et al, (2010). The probrf weighs a term differently when it appears in a different segment. They further proposed a term weighing scheme with distributed coefficient (DC-probrf) that is based on the "Global Log Inverse" (GLI). In their study, the authors used the Reuters-21578 collection and 20 Newsgroup collections for evaluation. The results show that both the distributional coefficient and term weighing scheme contributed significantly to improve TC effectiveness. A comparison between both however suggests that an appropriate term weighting scheme would have a better capability in classification.

2.2.2.3 Term Frequency Inverse Document Frequency

TF.IDF stands for weight is a statistical measure used calculate the importance of word in a collection of documents. The weight of the word is increases depending on the times where the word is appeared in the document. While this weight is decrease where this word is appeared many times in different documents. TF.IDF weighting method are used as an essential tool in scoring and ranking a document by search engines TF.IDF values can be used for stop-words removing in diffirent fields such as text classification and summarization.

All text classification researchers use only the product of TF and IDF. A drawback of IDF is that all texts that contain a certain term are treated equally, i.e., the IDF does not distinguish between one occurrence of a term in a text and many. Another drawback of TF.IDF is that when a new document occurs, recalculation of weighting factors to all documents is needed since it depends on the number of documents (Deisy, Gowr, Baskar, Kalaiarasi, & Ramraj, 2010).

The study by (Sharma & Kuh, 2008) suggests that one of the most important feature in document classification is that for each word find the class document frequency (dfc). In their study, they proposed two algorithms Algd1 and Algd2. Algd1 designed based on dfc, while Algd2 is designed in which words with high dfc have high contribution than those with low dfc. TF and dcf used to train SVM, KNN, PrTFIDF and NB classification algorithms. The document sets considered are from Reuters-21578 collection for text classification. The result shows that the computational cost is reduced when Algd1 is used.

Zhang, Yoshida, & Tang, (2008) study the performance of three different indexing methods to create set of features for text classification task. "multi- word", TF-IDF and LSI are examined. In this study, SVM classification algorithm is used to perform the classification task. Chinese document collection and from four different classes are selected as training set. Also, "Reuters-21578" corpus is used in this study. The results indicate that multi-word in English dataset performed better than TF-IDF, while TF-IDF is the better when it goes to Chinese collection.

To address remedying the defects of traditional mutual information method, Xiaoming & Yan, (2013) attempts to improve the methods to measure mutual information. A framework for feature selection named "Minimum Redundancy Maximum Relevance" (MRMR) is proposed. MRMR expands the feature set with the features that are maximally dissimilar to each other. This method uses a KNN as the class prediction method. To evaluate the improved method, this study performs experiments on datasets selected from "Fudan University". The result show that the feature selection method proposed MRMR is effective.

A new term weighting algorithm which used class information proposed by (Zhanguo, Jing, Liang, Xiangyi, & Yanqin, 2011). This algorithm is named as "TF.IDF class information" TF.IDFci. Two parts, intra class information and inner class information are developed to for the TF.IDFci. The intra class information is increasing with the sum of documents assigned to the class. While the inner class information is calculated by give the largest value to the term founded equally in the documents of the class. To evaluate the study, dataset are downloaded from the sogou website. To classify text, the NB classifier is built. The experimental results from this study show that there is enhancement in its performance.

A feature selection framework named minimum redundancy-maximum relevance (MRMR) was proposed minimum feature redundancy measure. A deficiency of this simple ranking approach is that the features could be correlated among themselves. There are two aspects of this problem. Because the features are selected according to their discriminative powers, they are not maximally representative of the original space covered by the entire dataset. The feature set may represent one or several dominant characteristics of the target phenotypes, but these could still be narrow regions of the relevant space.

Xi, Hang and Mingwen (2012) proposed the "Regularized Least Squares Multi Angle Regression and Shrinkage" (RLS-MAR) Model. Along with this model, "Term Frequency-Inverse Document and Class" (TF-IDC) is a new term weighing method assigning low weight to useless features in the classification problem. The method is then used alongside SVM Light and KNN to test its effectiveness. The dataset for evaluation is Reuters-21578 for comparison, used "Normalized log TFIDF" weighting method as the representation. The experiments show that results have some fluctuations.

Yan Li & Chen, (2012) developed a new system to classify the Chinese documents. This classification system uses "High Term Frequency and Weighted Document Frequency" (HTF-WDF) algorithm which was proposed by authors. Two efficient modifications are made. One is a formulation term frequency for each term from document to get the related features. The other formulation, calculates the document frequency coefficient for all term in the class. To evaluate the proposed algorithm, SVM classification algorithm is used. The results from this study indicate that the classifying accuracy of the DF feature selection method is enhanced by using WDF method, and the classifying accuracy is upgraded by adding the terms with high HTF.

A supervised feature selection scheme is proposed by Basu and Murthy (2012) for text classification. Known as "Term Significance" (TS), this scheme is built upon the concept that there are two types of probabilities. First probability is the terms present a document, and the second probability is the classes contain the document as well as the number of documents contain the same terms and from the same class. The TS is applied on Reuters-215781, data sets tr31 and TREC-5 and TREC-6h dataset. The effectiveness of this study is judge by using the KNN classifier results from this study has shown that the performance is improved.

A novel word scoring metric called the GU Metric was proposed by (Uchyigit, 2012). This scoring metric computes the difference between number of documents from the relevant sets which contain words and those from the irrelevant sets. The difference is computed by dividing the proportion of words from the relevant document by the words from the irrelevant sets. The data sets were obtained from "20 Newsgroup" data sets which consist of "Usenet articles". The experiment was conducted using NB classifier for each group. The experiments show that GU Metric obtained the best result.

Zhangou et al, (2011) used the information contained in the title, abstract and conclusion in a scientific literature to perform term weighing. To perform fuzzy search, this study used the natural language dictionary created by "Harbin University of Technology". NB algorithm is used to test the proposed term weighing method using corpus from various fields. The results from this study indicate that there is some improvement in its performance. It was noted that the important terms role has risen while the role of the negligible terms term has become less.

The author of (Nguyen, Chang, & Hui, 2011) propose term weighting method called "term frequency Kullback- Leibler" (tf x K L). The generalized tf × KL calculates the weights for each term depending on the ratio of the positive and negative class conditioned word probabilities. It considers the Kullback- Leibler (KL) divergence, a broader class of divergences known as "Ali-Silvey distances" or "f-divergence". SVM classification algorithm tested on four datasets namely, Movie Review, Sentiment, 20-Newsgroups is used for testing the proposed term weighted method. The performance of the generalized tf x KL gives up to 20% better in terms of Fl measure.

Zhu and Xioa, (2011) have proposed R-TFIDF to give equal chances when long and short documents are retrieved. This method decreases the weight-term in short documents thus document length normalization side effects will be alleviated. Three classes were designed and for each class assigning 25 documents to test the R-TFIDF. The experiment results show that the performance of R-TFIDF was improved.

"Term Frequency and Class Relevancy Factor" (TFCRF) is a novel feature weighing method for TC is proposed by Maleki, (2010). This method adds the class information in computing the weight of each feature and makes the feature weight more reasonable. Simulations were conducted and results show significant improvement in the performance of SVM classifier by using TFCRF feature weighting method in comparison to the other implemented standard feature weighting methods such as TF, IDF and Class based methods for text classification. Evaluation the proposed feature weighting method using the INEX dataset and DF threshold feature selection method.

Another work by (Wang, Wang, & Zhang, 2010) study the classic TF-IDF weight function. These two methods try to solve the problem of previous TF-IDF function. TF-IDF represents the relationship between the terms and text and it ignores the relationship between terms. The relationship between the terms is described using two methods. First method describes the relationship depending on the distribution information among classes. The second method describe the relationship between the terms depends on information inside a class. In their study, set of text created randomly in three different classes. The conclusion from this study shows that the improved weight function is effective.

Jin, Xiong, & Wang, (2010) proposed a method to perform Chinese TC. This method which filters features coarsely combines the superiorities of TF-IDF and Chi feature selection method. For the inner-class measure TD-IDF is used while for the inter-class measure Chi is used. Also this study used the Swarm Intelligence to present feature selection method for TC. To evaluate the proposed work, SVM classification algorithm is used. The experiments were conducted using "Fudan University Chinese

TC Corpus". The result from this method indicates that the dimension of features represents the Chinese text is effectively reduced by combining inner-class and inter-class.

Semantics-Based Feature Vector creation, is proposed by (Khan, Baharudin, & Khan, 2010). Under this method, the terms are extracted using POS then the feature vector is created from these terms using MFS algorithm. Also more terms added to this vector depending on frequent phrases. Using WordNet, these terms are converted into concept. The "Reuters-2l578" dataset is used for the proposed method. The results from this study show that the improvement is about 15 % as compared to the BOW.

Hui & Siqing, (2010) proposed a term weighing algorithm which takes into consideration factors such as word length, location. The authors opined that word positions play an important role in text classification. Word position reflects the role of the word in the text while word frequency reflects the characteristics of the text. The dataset used in this work is downloaded from "Fu Dan University Department of Computer Information and Technology Centre". To study evaluate the proposed work, KNN classification algorithm is used. The result shows that the performance of this algorithm is improved comparing with the traditional KNN algorithm in terms of accuracy.

Jiang, Li, Hu, & Wang, (2009) attempts improve the TF.IDF term weighting approach by proposing new weighting method. A supervised term weighting scheme, which directly makes use of a kind of information ratio to judge a term's contribution for a class is proposed. In this method tf.idf technique used to calculate the weight, where rare terms got the higher value. In this study, KNN multi-classifiers were designed where every class has a classifier. Reuters-21578 is used as benchmarking data set in this study. The results show that when the size of feature set small, performance of text classification get best. Moreever, the performance of The KNN algorithm will degrade as the feature set grows. The improved method largely outperforms the traditional TF.IDF method.

Table 2.1

Literature summary on feature creation in text classification

AUTHOR	TECHNIQUES	CLASSIFICATION ALGORITHM	DATASET	STRENGTH	WEAKNESS
(Celik & Gungor, 2013).	POS	SVM	20News group ,Classic 37 Sectors, WebKB 5396 Reuters-21578.	Micro-F measure with more than 50%.	Time for training classifier is high.
Che & Teng (2009).	POS,HowNet Chi,IG	NB ,SVM	TC Chinese corpus of "Fudan Univ".	drop the concepts not suitable for representation while not losing the lexical semantic information.	For the wrong sense, CHI method is considrabley sensitive, Chi feature selection method gives the lowest value in the experiment.

Table 2.1 *Continued*

(Xia, Chai, & Wang, 2012).	Chi	SVM	portal sites by the VIPS module.	The experimental results show that the performance of the proposed work is 91.6%.	relevance vector methods is in the complexity of the training phase.
Luo et al, (2011).	N-gram part-of-speech	SVM	TanCorpV1.0	Micro averaged F-measure was to 85.05%.	For chines document.
Nguyen, Gao, & Andreae, 2011).	N-gram	NB,K-NN, SVM	Ancient /Modern Vietnamese.	Interest for some nontopic based TC, enhancement in effeceincy and effectiveness.	Time for training classifier is high.
Ping et al (2010).	GLI CHI	Distributed coefficient based on the "Global Log Inverse" (GLI).	Reuters-21578 collection and 20 Newsgroup.	Distributional coefficient and term weighing scheme improve TC.	Sense-based representation using CHI performs rather poor and acquires the lowest score in the experiment.

Table 2.1 *Continued*

(Sharma & Kuh, 2008).	Tf.idf	SVM, KNN, NB	Reuters-21578 collection.	Algorithms developed solely on the basis of Tf.idf shows performance that compares closely with that of more complex machine learning algorithms.	The term frequency does not add to the performance compared to the class document frequency., ignore all the weighting technique just binary value is zero or one so that mean the low frequency and high frequency same.
Zhang, Yoshida, & Tang, (2008).	TF-IDF and LSI	SVM	Chinese document collection "Reuters-21578".	TD*IDF and multi-word have comparable. .	LSI is not sensitive to the scaling factor. LSI can produce a comparable recall precision and F-measure.

Table 2.1 *Continued*

Xiaoming & Yan, (2013).	TF.IDF MI	KNN	Datasets from "Fudan University"	MI is effective and feasible	It does not deal with the type of the dependency, but only with the quantity of dependency.
(Zhanguo, Jing, Liang, Xiangyi, & Yanqin, 2011).	TF.IDF	NB	sogou website	The macro-average precision is 79.93% with the new algorithm on Naive Bayes.	Mutual information method focuses on the correlation between terms and categories, without considering the connections between terms. It may select redundant terms.
Xi, Hang and Mingwen (2012).	Normalized log TFIDF	SVM Light and KNN	Reuters-21578	Better experimental results in SVM.	In the KNN, the experimental results have some fluctuations.

Table 2.1 *Continued*

Yan Li & Chen, (2012).	fuzzy rule, term frequency.	SVM	Chinese documents	The advantage of using fuzzy feature is it simplifies the relationship between the classifying result and the feature set.	Whereas the shortcoming is the number of total features is increased compared with the original feature set.
Basu and Murthy (2012).	DF	KNN	Reuters-215781, data sets tr31 and TREC-5 and TREC-6h dataset.	Effectiveness of this study is judge by using the KNN classifier.	Computation Complexity, limitation.
(Uchyigit, 2012).	DF	NB	"20Newsgroup".	GU Metric obtained the best result.	It does not give us an indication if the word is more favored in documents from the relevant set or irrelevant set.

Table 2.1 *Continued*

Zhangou et.al (2011).	Fuzzy search	NB	Scientific literature	The macro-average precision is 82.09%.	A disadvantage with Naive-Bayes is that if no occurrences of a class label and a certain attribute value together then the frequency-based probability estimate will be zero.
(Nguyen, Chang, & Hui, 2011).	TF-Kullback-Leibler	SVM	20-Newsgroups	20% better in terms of Fl measure.	KL has the disadvantage that it need to have access to the entire non-negative matrix. In problems where the state space is large, this may be problematic.

Table 2.1 *Continued*

Zhu and Xioa (2011).	TFIDF		Documents	R-TFIDF was improved, limitation is that a more comprehensive comparison among the proposed R-tfidf algorithm and other term weighting variations.	With a thorough comparison among the term weighting variations by using different datasets, document representation more suitable for information retrieval and text categorization/clustering may be improved.
Maleki (2010).	TF, IDF.	SVM	INEX dataset.	Significant improvement in the performance of SVM.	Time consuming.
(Wang, Wang, & Zhang, 2010).	TF-ID relationship between terms inside a class.	-	Text created randomly in three different classes.	The conclusion from this study shows that the improved weight function is effective.	If the size of each category is different this will effect on the result of term weighting.

Table 2.1 *Continued*

Jin, Xiong, & Wang (2010).	TF-IDF Chi.	SVM	"Fudan University Chinese TC Corpus".	Features represents the Chinese text is effectively reduced by combining inner-class and inter-class.	It is hard to analyze the computational complexity of these algorithms. Therefore, it is difficult to tell whether a swarm intelligence algorithm will be suitable for certain problems.
(Khan, Baharudin, & Khan, 2010).	POS MFS WordNet.	-	"Reuters-2l578".	Improvement is about 15 %.	WordNet has only a limited number of connections between topically related words.
Hui & Siqing (2010).	Word frequency.	KNN	"Fudan University Chinese TC Corpus".	The performance of this algorithm is improved comparing with the traditional KNN algorithm in terms of accuracy.	In this work the weight id given to keyword greater than key phrases while there are many key phrases effect on the result.

Table 2.1 *Continued*

Jiang, Li, Hu, & Wang, (2009).	TF.IDF	KNN multi-classifiers.	Reuters-21578.	The results show that when the feature set size is small, classification performance performs best.	The KNN algorithm's performance will decline as the number of features grows.

Even these studies give a comparable result in terms of feature vector creation for text classification task, some of these studies used rules and other used term frequency. From the studies presented in the previous section, all these studies used terms single or multi word to create feature vector for text classification process. POS, N-gram and term frequency are different techniques used to calculate weight of these terms. For each technique there are many limitations. First for POS, there is a need to create rules to describe grammar for each language to find the relations between the terms from a text. For n-gram it is easy way to find the terms single and multiple terms. An n-gram is a contiguous sequence of n items. While in TF.IDF the importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. These techniques ignore the semantic feature of the terms. For instance a synonym is a word with the same or similar meaning of another word and polynyms it is a words which have multiple meanings. And also there is more semantic relation between these terms. This effect on the results in term of classification performance. Table (2.1) shows different work that used many techniques to create set of feature for text classification task.

2.2.3 Feature Selection Method to Reduce Dimension

High dimensionality of feature space is one main problem in text classification. A feature space is "a set of unique terms or words that occur in a text document". To reduce the number of attributes in a feature set, feature selection is used. Reduction of attributes leads to higher processing speed. Within the text domain, the most popular FS algorithms include IG (Xu, 2012) , DF Yan Li & Chen (2012), Mutual

Information name (Hong-we, Jian-fang, & Feng, 2010) and CHI by (Zhang, Xu, & Wu, 2012).

2.2.3.1 Information Gain (IG)

Information Gain (IG) refers to the amount of information acquired for class prediction. This is achieved by noting the existence or not of a term in the sampled document. The IG value is computed and then compared with predefined threshold value. If IG is less than threshold, the term is moved.

IG is a frequently employed word scoring metric in machine learning. IG measures the number of bits of information obtained for class prediction by knowing the presence or absence of a word in a document. IG is another word scoring metric which shows conflicting results. Yang and Pedersen reported that IG was one of the methods which performed best compared with the others in their experiments. Mladenic reported that IG was one of the worst performers, its performance was similar to or worse than random method. Information gain has the disadvantage that it prefers attributes with large number of values that split the data into small, pure subsets. IG measure is that it is biased towards selecting attributes with many values a large number of distinct values (Wang & Jiang, 2007). Also, IG rewards features whose presence or absence tends to match well with the document's membership of one class or the other, and features that occur in very few documents do not score well according to this criterion. For text classification, many works try to reduce the dimension of data by using IG feature section.

A framework for Meta feature selection was proposed by Li (2013). To reduce the dimension of feature set, this framework combines attribute reduction in rough set theory with multiple feature selection algorithms. When combined with other feature selection algorithms, the classification accuracy is improved. Data is obtained from CORPUS provided by the "Chinese Academy of Computing Sciences". Articles are selected from six classes, the politics, economy, military, culture, industry and computer. For the validation this study the SVM classifier is used with feature selection method.

Another study deals with evaluation of different feature selection methods for reducing the dimension of feature set for filtering the spam was conducted by (Xu, 2012). Using NB and SVM, this study also employs other methods such as IG, CHI, "odd ratio (ODD)", "Expected Cross Entropy (ECE)" and "weight of evidence" for feature selection. To make a comparison among these different algorithms experiments were designed. To evaluate this study, dataset from public e-mail corpus are used. The results show that for filtering the spam, the feature selections methods ODD and WET are very competitive.

Nuipian, Meesad, & Boonrawd, (2011) used Chi, IG and "Gain Ratio (GR)" feature section methods to make a comparison between keywords and key phrases. To evaluate this study set of abstract from the "ACM Digital Library" was downloaded. Different techniques such as "data mining", "distributed systems", "knowledge representation formalisms" are used in this study. For experiment, different classification algorithms are presented such as Decision Tree, NB, BN, SVM and K-NN. Results showed that the best classification algorithm is SVM where the performance of single word in term of accuracy was 84% and for key-phrase the accuracy value was 74%.

Haruechaiyasak et al, (2008) proposed the "Sansarn News Search Engine" implemented by using an open platform named "Sansarn Look!", The "Sansarn News Search Engine" focuses on Thai texts and it considers three algorithms namely SVM, NB and Decision Tree. Three feature selection methods were adopted. There are DF, IG and CHI. the "normalized Term Frequency-Inverse Document Frequency" (NTF.ID) is used to calculate the weight of each feature. Experiments using a collection of news articles obtained the Web were performed. The result from this study shows that the SVM algorithm combined with IG feature section gives the best result in term of F1, which recorded 95.42%.

Almeida et al, (2009) presented a comparative study for anti-spam filtering using different feature selection methods combined with different original NB classification algorithms. This study used different selection methods named DF, IG, MI, Chi, and "odds ratio" for reducing the dimension of feature. To evaluate this study, TREC dataset are conducted. It found that IG and Chi statistic were most effective method for reducing the feature size and gives the best performance in terms of classification accuracy.

The work presented by (Haifeng, Shousheng, & Zhan, 2010) introduces an improved KNN text classification algorithm based on feature dimension reduction pattern. Text feature selection is conducted by an improved IG method for more efficient using the classification distribution information in the sample training set. A classification is conducted by an improved K-NN algorithm based on the sample class selection. The dataset used for testing this method are created from "Sina and Xinhuanet". The experiment result shows that the values of precision and recall value are improved with 8% in average, so the result is satisfied.

Xue et al, (2010) introduced a new text classification method known as the "IIKPC". This method highlights a new IG-based method for feature selection called "Improved-IG" (IIG), and a new algorithm for text classification called "Improved k-NN" (IKNN) basing on traditional k-NN classification method. The author opined that the relationship between the feature and class is of high importance and should therefore take precedence when classification is performed. In an experiment conducted on "570 US patents of pneumatic" tools predetermined into different classes, IIKPC gives the best result. The result indicates that the performance of the traditional K-NN algorithm or "IKNN" is better than the other classification algorithms.

Study names "Detecting Phishing Emails Using Hybrid Features" by (Ma, Ofoghi, Watters, & Brown, 2009) presented an approach to detect phishing emails using hybrid features. The work mainly consists of the usage of hybrid features namely content, orthographic and derived, and the feature selection method. Orthographic features reflect the author's styles and habit, so that the features are also informative as discriminators. Derived features are mined and discovered from emails which also provide clues for classification. To discover the importance of each feature, the IG of Induction is used for the feature selection for each feature. It is implemented using SVM, Decision Tree", "random forest", "multi-layer perceptron" and. To evaluate this method, dataset created from the live emails. Experimental results indicate that the effective classifier is generated after removing the redundant features. The result comes that the highest performance is come from Decision Tree algorithm.

Bagging classifier can also be used to design a TC as shown by Zhang et al, (2009). Bagging is a "bootstrap ensemble method that creates individuals for its ensemble by training each classifier on a random redistribution of the training set". Through this method, outputs from multiple classifiers are combined. VSM used to represent the document and for feature section IG method is used. In this study, the dataset is obtained from "Sogou La (http://www. sogou.com/labs/dl/c.html)". A comparison is made and it was observed that the recall rate, precision rate and F1 gained through attribute bagging are better.

The K-NN algorithm is also used for TC as shown by Bin et al, (2008). In their study, the K-NN is used to score essays that build on TC model. The essays were transformed into VSM while TF-IDF and IG were applied for FS. "Global Shortage of Fresh Water" was chosen as dataset which contains 271 applicable essays. Through this study, it was shown that by combining these different methods of Feature selection the accuracy achieve up to 76% accuracy.

Islam & Islam (2008) proposed a new method named random walk for weighting the terms to be used in text classification task. In their study, the relationship of local information and global information was exploited to weigh a term. For the local information term position and TF are used, while for global information IDF, IG is used. To evaluate this approach, Rocchio text classification algorithm is used. The experimental results show that method performs better than other random walk models.

2.2.3.2 Chi2-test (CHI)

Chi is based on the statistical theory. It measures the lack of independence between the term and the class. Chi is a normalized value and can be compared across the terms in the same class. For each class the Chi statistic between each unique term in a training corpus and that class, and then combined the class-specific scores.

The ”Chi-Squared” (x2) statistic was primarily used in statistical analysis to find how the results of an observation differ (i.e. are independent) from the results expected according to an initial hypothesis (higher values indicate higher independence). In the context of text classification Chi Statistic is used to measure how independent a word and a class.

Chi value get zero if the classes are independent. For the words appears in different classes frequently get the low value which indicate the relation between the word and class are high independence. While the Chi value get high value, if the word occurs in few classes (i.e." high dependence").

Chi method is one of the feature selection approaches which has good performance for TC. The "Chi statistic" supports words which are indicative of association of class but furthermore those words which are indicative of non-association of class (Varela, 2012; Erenel, Altincay, & Varoglu, 2011). The experiments shown that the Chi approach has two limitations. First one is that even the "document frequency" for many feature words is low in a class, they have a lower "document frequency" even almost to zero in the other classes. So, it is a characteristic word of such class, and is a useful characteristic of this documents. While the evaluation value calculated by Chi statistical method is very low so that this kind of words doesn't belong to the extracted feature subset. Secondly, for the feature which is common in many classes and appeared rarely in specific classes, the weight is increased. Many researchers used the Chi to reduce the dimension of text in classification.

Work by (Zhang, Xu, & Wu, 2012) proposed "dual feature selection method" using N-gram and Chi method for feature selection. Binary text classification is produced by using SVM classification algorithm. N-gram is chosen to perform feature selection because feature selection method use by itself results in difficulty in distinguishing between two documents. Experimental results show that, the performance of the proposed methods achieved higher performance comparing with the traditional SVM algorithm in terms recall.

The study by (Mesleh & Kanaan, 2008) focused on generating feature subset for text classification tasks. In this work, "Ant Colony Optimization algorithm" is presented to improve the Chi- Feature Subset Selection process. It preferred to trade some solution quality for computational complexity and decided to implement an "Ant Colony Optimization based FSS" Algorithm for Text Classification. To evaluate the performance of the proposed "Ant Colony optimization based FSS" method, a corpus collected from "online Arabic newspaper archives", including "Al-Jazeera", "Al-Nahar", "Al-hayat", "Al-Ahram", and "Al- Dostor". This proposed method improved the performance SVM classifier for of Arabic text. The experiment result shows that by using different features selection methods the classification performance achieved best recall, precision and F1 values.

The study by (Kadhim & Omar, 2012) conducted a study to improve the performance of Bayesian learning classifiers. Several "Bayesian Learning" Classifiers, such as "Multivariate Guess Naïve Bayes" (MGNB), "Flexible Bayes" (FB), "Multivariate Bernoulli Naïve Bayes" (MBNB), and "Multinomial Naïve Bayes" (MNB) were used to perform TC on Arabic Texts. These techniques were applied along with CHI, MI, OR and GSS methods and were then analyzed. The texts used consist of 3172 documents which are divided into four categories, namely Arts, Economics, Politics and Sports. Through their experiment, it was discovered that FB achieves the best recall value when Chi FSM is applied using 1-gram representation.

Mouratis & Kotsiantis, (2009) attempt to enhance the performance of the "Discriminative Multinomial Bayesian" classification algorithm combined with a feature selection methods. In their study, the "Naive Bayes Multinomial algorithm" and "Discriminative Multinomial Naive Bayes" classifier were used and tested with Chi methods to select the best discriminated features in training set. The text used was accessed from "http://www.hpl.hp.com/personal/George_Forman//Hewlett-Packard". Through their study, it was revealed that the "Discriminative Multinomial Bayesian" Classifier resulted in higher accuracy particularly when calculating the Chi value for attributes with respect to the class.

Another study by (Meena & Chandran, 2009) proposes a novel text classification methods with features selected. "CHhoice of Internal Representations" (CHIR). CHIR is a supervised learning algorithm based on Chi method. This algorithm not determines type of relation between the terms and classes. In their study, the authors used the "20Newsgroups corpus" to test the proposed work. The NB classifier used to evaluate this method. The results show that the performance of this method is enhanced in terms of accuracy when using Chi selection method.

Kim and Chang (2007) explored a novel way to improve NB text classifier by combining learning algorithm and feature weighting. For each class, the feature weighting assigns more weights on the best features, by using Chi based feature ranking. This study highlighted that the features weights is determined depending on their distribution across different classes. The study also suggests that to achieve incremental feature weighting, it must connect statistical properties of the weighted feature set and the classifier performance. Using the Chi based feature ranking, assigning weights over class-specific topic features allows the NB learning to increase the uniqueness of classes. In this study, the ”Reuters-21578” and “20 Newsgroups” were used. The experiment shows that enhancing the model by weight assignment is a good approach to the NB classifier for performance improving.

2.2.3.3 Document Frequency Thresholding (DF)

DF is the number of documents in which a term occurs. For each terms from training set represent a class, the DF is calculated. Then all terms with DF values which are less than predefined threshold value are removed. The removal is due to the basic assumption that "rare words are, either no informative for class prediction, or not influential in global performance".

DF method ignores the term frequency which effects on the ranking of the feature. DF threshold is the simplest technique for vocabulary reduction. It easily scales to very large corpora with computational complexity approximately linear in the number of training documents. However, it is usually considered an ad hoc approach to improve the efficiency, not a principle criteria for selecting predictive feature. Also DF typically is not used for aggressive term removal because of a widely received assumption in information retrieval. So, Low DF threshold are assumed to be relatively informative and therefore should not be removed aggressively (Yang &

Pederson, 1997). For text classification, many works try to reduce the dimension of data by using DF feature section (Mathy, 2010; Yang, & Pederson, (1997).

Yan Li & Chen (2012) proposed a method called "High Term Frequency and Weighted Document Frequency" (HTF-WDF) to address the shortcomings of the original DF method. The HTF-WDF uses SVM to perform TC. The Chinese documents obtained from "Fudan University" are selected to create dataset for training testing the proposed method. The study shows that the WDF algorithm enhances the performance of text classification in term of accuracy comparing with traditional DF algorithm.

The TFCRF presented by (Maleki, 2010) is a new method for feature weighting, specifically designed for text classification. The TFCRF uses weight of a feature as a function of its distribution within different documents. It also uses DF threshold method for feature selection. In the experiments, a total of 12107 articles sourced from the IEEE Computer Society's publications between the period of 1995 and 2005 were used. The simulation employed the parameter setting used in the LIBSVM library version 2.82. The best values of "micro-averaged F1" and "macro-averaged F1" of SVM classifier by using TFCRF feature weighting method are 0.933and 0.939 for 4000 features respectively but these values for TFRF method are 0.883 and 0.889 for 2000 features.

Gang and Jiancang, (2009) evaluated the performance of feature selection using SVM classification algorithm. The authors carried out experiments centered on the testing the performance of DF, Chi, DF+Chi feature selection. The classification first takes pre-classification by titles and preconditioned preset key words. If pre-classification turns out to be successful, the results will be yielded without undergoing the discrimination by the classification machines, or even generating feature vectors, thus accelerating dramatically the categorization. However, if pre-classification fails, feature vectors will be generated and SVM is constructed by using LIBLINEAR database. The F1 value of SVM algorithm has risen respectively from 0.826 to 0.935, while the categorization time is reduced from 1576 seconds to 863 seconds.

Xia et al, (2009) proposed "Text Categorization Method Based on Local Document Frequency" (TCBLDF) method for text classification. In order to reduce high dimensionality, DF feature selection is implemented before training algorithm. The score that each item in the feature set contributes to all classes to build a term-class contribution table is computed. For each term from the testing set document, the scores it contributes to all classes are looked up and the scores of different terms within the same class are combined. Finally, the document is classified into the class with the largest score. A DF for local dimensionality reduction was used. In this experiment, the authors used the Reuters-21578 Text Collection and the 20 Newsgroups Text Collection.

Yusof and Hui, (2010) uses "Artificial Neural Network" ANN and employed DF methods and "Class Frequency Document Frequency" CF-DF feature reduction methods applied "Bloom’s taxonomy" to classify question items. The CF-DF method presents a measure named "the class frequency" that allows the discrimination value of a feature to be considered in the feature reduction process. The reduction of feature sets will lead to the reduction of ANN input complexity. In their study, training set which defined as set of questions is created using "the Bloom’s cognitive level" of each question before a classifier can be trained. The experiments conducted show that the proposed method enhanced the performance of classification in terms of time.

Harrag et al, (2010) presented and compared the results achieved from Arabic text collection using "Dimension Reduction techniques" with "Back-Propagation Neural Network" (BPNN) algorithm. A three layer "feed-forward neural network" with "hyperbolic tangent activation function" in the hidden layer, and linear output layer is presented. In this study, to reduce the feature size, "Stemming", "Light-Stemming", DF, TF.IDF and "Latent Semantic Indexing" (LSI) methods were used. The dataset is downloaded from "Hadiths (Sayings of The Prophet Mohammad Peace Be Upon Him)" collected from "the Prophetic encyclopedia" and "(Alkotob Altissâa, The Nine Book)". The results indicate that the proposed method achieved high performance in terms of Macro- Average F1 measure for Arabic text classification. Experiments on Arabic datasets show that the TF.IDF, DF and LSI approaches are favorable in terms of their efficiency.

2.2.3.4 Mutual Information (MI)

MI is used to represent the correlation between two variables such as feature and class. MI is performed by first finding the number of documents which belongs to class and contains specific feature. Then find the number of documents which does not belong to this class but contains this term is counted. Finally, the number of documents which belongs to some class but does not contain that term is calculated. This method is used in many researches to find the feature subset to classify text to some predefined cases.

Studies show that there are some shortcomings of mutual information. First, mutual information only considers the document frequency of terms appearing in texts, without considering the word frequency. It may make different terms have the same mutual information, which may lead to lose a lot of useful information while the system deleting the features, which have the same weight as the previous, but unfortunately at the back of the entry. At the same time, it could select the rare words as representative of texts. It may make the weight of words that appear multiple times in small part of the corpus less than the weight of words that appear few times in most part of the corpus, which could make the representative words cannot be selected. Second, mutual information method focuses on the correlation between terms and categories, without considering the connections between terms. It may select redundant terms. The traditional mutual information bases on one condition that the amount of texts in each category must be roughly equal (Hong, 2014; Zhang, Yoshida, Tang & Hu, 2009). For text classification, many works try to reduce the dimension of data by using MI feature section.

In their study, name (Hong, Jian, & Feng, 2010) used TF.IDF method and to calculate the characteristic weight. They also used SVM algorithm to classify the text, and compared two types of weight calculation approaches. In order to solve the problem that arises from TF.IDF method, it combines information, word position, word relations, word frequency, and document frequency. MI feature selection method is used to reduce the dimension of the word vector. The training corpus consists of 9605 documents which are created manually and divided into 20 different classes. To evaluate the proposed method, SVM classification algorithm is used. The experimental results show that the performance of this method is improved in terms of accuracy.

The study proposed by (Lu, Shi, Zhang, & Yuan, 2009) aims to reduce the dimension of feature size using MI method. For the traditional MI method give emphasis to the term with low weights. To address this, first filtering all uncommon terms with low frequency by using dictionary created manually. The authors have employed several methods such as "Back-and-forth maximum-match", "shortest-path", "omni-segmentation" and "maximum-probability". For testing, K-NN and SVM classification algorithms are used. In the experiments, corpus for Chinese documents is created. The result shows that there are some improvement from the new method in term of precision ,recall and F1 comparing with traditional MI in some experiments, but of traditional MI method gives the best result in terms of Maco F1 for both K-NN and SVM.

Pang et al, (2007) study the effects of using "Maximum Entropy" (ME) for "paper comment classification" (PCC). In their study, they proposed a novel method that combines both "entropy" and "Maximum Entropy" (ME) perform feature extraction in DC and PCC. The authors also presented a SVM method to perform the PCC. In this study, the MI is used to find relationship between a term and a PCC or class. ME is a general technique used to estimate the probabilistic distribution from the training data. Using the ME, several experiments such as "Baseline", "Maximum Entropy" with MI, ME with "Average Mutual Information" (AMI) and MI with CE are performed. The corpora used in these experiments selected from set of journals. Through the experiments, the best performance is noticed in ME with CE, and the improvement recorded is 2.78% better than Baseline.

Warintarawej et al, (2011) proposed a new approach to select the top-k classes to classify the text. This approach uses two feature selection methods to perform classification. The first method is based on the DF concept named "syllable frequency" (SF). Under this approach, the syllable frequency is counted and ranked. The second approach uses MI. In this study, the authors used word corpus gathered from two sources, namely the French Larousse thesaurus. It was argued that syllables play important roles for classification model, and performs considerably better than MI.

Wei, Gao, and Wu, (2010) used several statistical classification and machine learning techniques to perform text document classification. The techniques include different classifications algorithms such as regression models, K-NN, Decision Tree, Bayesian, SVM and NN. In their study, the Feature selection methods were applied to TC. This includes DF, IG, MI, Chi, "Cross Entropy" (CE) and" Primary Component Analysis" (PCA). Three datasets were used in this experiment, namely the dataset 3S and dataset 3D which were derived from the Reuters-21578 corpus. Through the experiment it was discovered that MI reduces high dimensionality better that the other feature selection methods.

A new method for feature selection to classify the text is proposed by Pei et al, (2010). This method, called "Mutual Information and Information Entropy Pair Based Feature Selection Method" (MIIE _FS) aims at maintaining MI values while reducing redundant features in the feature selection process. In this study, the authors used the Reuters-21578 Top datasets. The Naïve Bayes and KNN algorithms were used along three different SVM methods. By comparing MIIE _FS to Chi and MI, the feature size is reduced and the value of Macro F1 is enhanced from 84.6% to 87.7%.

Xiaoming and Yan, (2013) proposed a method which reduces the feature size by reducing the redundant features from feature vector. This method is dependent on the "maximal relevance and minimal redundancy criteria" (mRMR). To evaluate this method the Reuters-21758 dataset is used which was divided into 10 popular categories. The study employed NB and SVM classification algorithms. The authors then propose another selection model named text-based word frequency which performs considerably better than the earlier model.

Fu, Chen, Gong, & Bie, (2008) performed different methods to select feature from several feature selection methods for text classifications such as Chi, IG and GR, to select the terms from web pages. This is conducted to enhance the performance of text classification and to reduce the complexity of this algorithms. In the experiments, web pages classification and the comparison of Bayesian classification methods such as NB, BN, "Averaged One-Dependence Estimators "(AODE), "Homologous NB" (HNB) and "Classifier NB" (CNB) were performed. The experimental results show that the AODE and HNB performance are competitive comparing with the other algorithm.

2.2.3.5 Ontology to Reduce the Dimension

One of the main approaches used to represent the content of document for text classification algorithm is "Bag of Words" (BOW). The drawback of BOW however, it calculate the frequency of each term from document and neglecting the semantic relations between these terms. Numerous attempts have been made to reduce the high dimensionality, particularly by replacing terms with concepts in the training set.

To address the limitation of high dimensionality, ontology has been used specifically for content-based classification in large document corpora. The researchers conducted using ontology is discussed below.

In a recent study name "Text Categorization of Biomedical Data Sets Using Graph Kernels and a Controlled Vocabulary" conducted by Bleik et al, (2013) attempts to represent text documents as concept graphs that preserve semantic relationships between the concepts presented in the text. The graph construction involves mapping biomedical terms that are extracted from 563 full-text articles selected from six journals of medical sciences into predefined concepts of a controlled vocabulary Unified Medical Language System (UMLS) database (an external ontology of biomedical concepts). To limit the size of the concept graphs, only half of the text content of each document is used to build the corresponding graph. The full-text documents were used with the text-based NB, SVM, and k- NN classifiers. To reduce the dimensionality of the feature space, edges having weights below a certain threshold were dropped from the feature set. The results show that the rich graph representation of documents improves the classification performance significantly, particularly when compared to other common TF.IDF text-based classifier.

Another study that shifts from keyword-based representation to key-concepts is proposed by Ajgalik et al, (2013). The advantage of using concepts over simple words is that concepts, apart from words, are unambiguous. This leads to better understanding of key-concepts than keywords. This work try to extracted key-concepts which is a substantial dimensionality reduction of document's feature space. Since it corresponds exactly to some WordNet synset, it can easily retrieve the exact meaning of it. In addition, it knows exact relations to other synsets, like hypernym, hyponym, homonym, meronymc. Ajgalik evaluated the performance of two standard classifiers K-NN and NB classifier.

Another approach was proposed by (Shein & Nyunt, 2010) used domain ontology to select the features and sentiments from the software reviews. This study attempts to enhance the sentiment classification tasks. The approach used domain ontology to extract the related concepts and attributes, while SVM classification algorithm is used for assigning concepts and attributes as positive or negative.

The main purpose of research proposed by (Dollah & Aono, 2011) is to enhance the hierarchical text classification performance by increasing the accuracies of classes in the datasets that are represented with a small number of biomedical text abstracts. The researchers have exploited the ontology hierarchical structure. "Anchor-Flood algorithm" (AFA) is used to search the most related concepts from ontology for text classification which reduce the size of feature set by replacing terms with set of concepts. To evaluate the performance of the approach, dataset from Medline abstracts and the OHSUMED corpus are used. The researchers also presents more experiments using LIBSVM classification algorithm for multi-class classification. By using ontology concepts instead of terms, the dimension of feature set is reduced. Different feature selection methods were used in this study to select relevant features such as Chi and DF. The results indicate that improved the performance of the proposed works is improved.

Another work is proposed by (Xiaoyue & Rujiang, 2009) attempts to use BOC instead of BOW to enhance the accuracy of text classification. The BOC develops classifiers which classify the text semantically using different RDF ontologies for index the text using concepts from ontology. For classification task, SVM algorithm is used. To evaluate this method, OHSUMED, Reuters-21578 and 20 Newsgroups (20NG) collections are used. The results show that the performance of BOC in terms of micro -F1 values and macro-F1 values is better than BOW for all datasets.

A new method to classify the document by finding the optimal concepts from ontology is proposed by (Wang, McKay, Abbass, & Barlow, 2002). In this study to find the related concepts from the structured ontology, hill climbing algorithm is used. K-NN algorithm is used to classify the text. Set of documents selected from journals in the MEDLINE database for creating training document and test document. To evaluate this method title and abstract from this dataset are tested. The dimension of feature set is reduced by using concepts from ontology. The results show the performance of the proposed methods is improved in terms of accuracy.

A novel ontology-based text classification approach of is proposed by (Zhang & Song, 2006). The approach proposes approach to measure the semantic similarity among the different concepts. VSM approach combined with the fuzzy technology to classify the text to set of predefined ontologies. In this sturdy, the classes are set of predefined ontology. This model used K-NN algorithm for classification. The results from this study show that by using small size of keywords by using concepts instead of terms, the accuracy dose not degrades.

A study names "Ontology-Based Feature Weighting for Biomedical Literature Classification" was proposed by (He & Wu, 2006) as a strategy used to calculate the weight of feature to classify the biomedical text by using ontology. From this study the semantic information incorporated and the size of feature vector is reduced by replacing terms with concepts from ontology. To evaluate this work, abstracts from MEDLINE journal s collection is selected. A binary classifier is built for each group of journal. The experiments from this study indicate that the improvement is significantly achieved in terms of accuracy.

A novel approach is proposed by (Shahi, Issac, & Modapothala, 2012) to enhance the accuracy of NB classification algorithm for short "Corporate Sustainability Reports (CSR)" documents. It studied the ontological characteristics of document categories and grouping them under virtual super-categories to narrow down the search for a suitable category. For this study, the "Correlation Feature Selection" (CFS) algorithm is used for feature selection. Best First method is used to find the relevant attributes .Also, "greedy hill climbing" method with a "backtracking facility" to search the space of attributes subsets so that the classes candidate will be reduced therefore the performance is improved.

The study by (Li & Hu, 2009) try to enrich the corpus with semantic information using statistical methods. It proposes a VSM by presenting the semantic similarity between the concepts to VSM. Concepts from ontology are adopted instead of terms as feature. To test the proposed work, Polynomial Bayesian classification algorithm is used. By using "Bayesian Minimum-Error-Rate Decision-making theory" the falling classes are predicted. The ontology used to evaluate this study is Hornet 2000 and corpus from Sougou Lab is selected as training set. Experiment results from the proposed study shows that the performance classification is enhanced.

A new method named "ontology-based method for building text classifier with Bayesian theorem" (ADCS_BO) is proposed by (Chang & Huang, 2008). First ontology for specific domain is constructed using Domain Ontology Module using "theorem of formal concept analysis" and the expert will support this model with set of the synonym. Then the Bayesian classifier is used to classify documents. Collections of documents from "Electronic Theses and Dissertations System" corpus is selected to evaluate this method. The results from this study show that the performance of this study records 80% effectiveness for some classes and for other classes it records 60%.

A new method for weighting the terms and feature selection methods using ontology are proposed by (Khan, Baharudin, & Khan, 2012). "Maximal Frequent Subgraphs" (MFS) algorithm is used to select the association terms from document. To improve the performance of VSM, the concept for each term is selected from ontology. A collection of documents downloaded from Internet are used to evaluate the proposed method. The classifier is used to test this study. The results show that the F1 value is improver for all classes and recoded 85.36%. So that by comparing this work with TF.IDF the improvement reaches 10.93%. The limitation of this study is the time needed for ontology creation and concept extraction.

Table 2.2

Literature summary on reducing dimension

AUTHOR	TECHNIQUES	CLASSIFICATION ALGORITHM	DATASET	STRENGTH	WEAKNESS
Li (2013).	IG	SVM	"Chinese Academy of Computing Sciences".	Accuracy is improved.	The major limitations of the traditional rough sets model in the real applications is the inefficiency in the computation of core and reduct, because all the intensive computational operations are performed in flat files.

Table 2.2 *Continued*

(Xu, 2012).	IG,CHI,ODD.ECE, weight of evidence.	NB and SVM.	From public e-mail.	ODD and WET are very competitive.	An odds ratio does not meaningfully describe a marker's ability to classify subjects.
Nuipian, Meesad, & Boonrawd, (2011).	Chi, IG,GR.	NB, BN, SVM and K-NN.	ACM Digital Library.	This work make a comparison between these feature selections.	Time consuming.
(Haruechaiyasak, Jitkrittum, Sangkeettrakarn, & Damrongrat, 2008).	DF, IG , CHI. NTF.ID	SVM, NB and Decision Tree.	A collection of news articles obtained the Web (Thai texts).	SVM algorithm combined with IG feature section gives the best result.	Information gain has the disadvantage that it prefers attributes with large number of values that split the data into small, pure subsets.

Table 2.2 *Continued*

(Almeida, Yamakami, & Almeida, 2009).	DF, IG, MI, Chi, and "odds ratio".	NB	TREC dataset	IG and Chi statistic were most effective method for reducing the feature size.	Limitation of using the Naïve Bayes classifier is that the real-world data may not always satisfy the independence assumption among attributes.
(Haifeng, Shousheng, & Zhan, 2010).	IG	KNN	Dataset from "Sina and Xinhuanet".	The values of precision and recall value are improved with 8% in average.	Memory and classification time computation are very high.
Xue, et. al (2010)	IG-	k-NN	"570 US patents of pneumatic".	K-NN algorithm or "IKNN" is better than the other classification algorithms.	Memory and classification time computation are very high.

Table 2.2 *Continued*

(Ma, Ofoghi, Watters, & Brown, 2009).	IG	SVM, Decision Tree", "random forest", "multi-layer perceptron	Dataset created from the live emails.	Tthe effective classifier is generated after removing the redundant feature.	Time consuming for training classifier.
Zhang et.al (2009).	VSM IG	Bagging classifier.	"Sogou La (http://www. sogou.com/labs/dl/c.html)".	The recall rate, precision rate and F1 gained through attribute bagging are better. Cost time to train these classifiers.	These works make a comparison between these feature selections.
Bin, et. al (2008).	VSM while TF-IDF and IG.	K-NN	"Global Shortage of Fresh Water".	The accuracy achieve up to 76% accuracy.	Memory and classification time computation are very high.
Islam & Islam (2008).	Tf.idf IG	**Rocchio**' classifier	Text document.	Performs better than other random walk models.	The feedback results from **Rocchio**' are not comparable.

Table 2.2 *Continued*

(Zhang, Xu, & Wu, 2012).	N-gram and Chi.	SVM	Text document.	Higher performance comparing with the traditional SVM algorithm in terms recall.	Time consuming for training classifier.
(Mesleh & Kanaan, 2008).	Ant Colony optimization Chi- Feature.	SVM classifier.	"Online Arabic newspaper archives".	By using different features selection methods the classification performance achieved best recall, precision and F1 values.	The main problem in Ant Colony Optimization algorithm, for a large number of nodes, are very computationally difficult to solve exponential time to convergence Coding is somewhat complicated.

Table 2.2 *Continued*

(Kadhim & Omar, 2012).	CHI, MI, OR and GSS.	"Bayesian Learning" (MGNB), (FB), (MBNB), and (MNB.	Arabic Texts.	FB achieves the best recall value when Chi FSM is applied using 1-gram representation.	The main reasons following that are the use of features number, which in fact fixed for each category and the document dataset does not vary in size.
Mouratis & Kotsiantis, (2009).	Chi	NBM, DNBM	"http:/.. //Hewlett-Packard".	"Discriminative Multinomial Bayesian" Classifier resulted in higher accuracy particularly when calculating the Chi value for attributes with respect to the class.	Their use in practice is often limited due to implementation difficulty, inconsistent prediction performance, or high computational cost.

Table 2.2 *Continued*

(Meena & Chandran, 2009).	Chi	NB	"20Newsgroups ".	Enhanced in terms of accuracy when using Chi selection method.	In NB classifier, real-world data may not always satisfy the independence assumption among attributes.
Kim and Chang (2007).	Chi	NB	Reuters-21578 and 20 Newsgroups.	Enhancing the model by weight assignment is potentially a good strategy to the NB classifier to improve its performance.	In NB classifier, real-world data may not always satisfy the independence assumption among attributes.

Table 2.2 *Continued*

Yan Li & Chen (2012).	DF	SVM	"Fudan University".	Weighted Document Frequency algorithm enhances the performance of text classification in term of accuracy.	Time consuming for training classifier.
(Maleki, 2010).	DF	SVM	Articles sourced from IEEE.	The best values of "micro-averaged F1" and "macro-averaged F1" of SVM classifier by using TFCRF are 0.933and 0.939 for 4000 features respectively but these values for TFRF method are 0.883 and 0.889for2000 features.	Time consuming for training classifier.

Table 2.2 *Continued*

Gang and Jiancang (2009).	DF, Chi, DF+Chi	SVM	Document.	The F1 value of SVM algorithm has risen respectively from 0.826 to 0.935.	Time consuming for training classifier.
Xia et. al (2009).	DF	Employ a binary weighting method.	Document.	Reduce dimension.	The use of binary weights is too limiting and proposes a framework in which partial matching is possible.
Yusof and Hui (2010).	DF	ANN	"The Bloom's cognitive level".	The experiments conducted show that the proposed method enhanced the performance of classification in terms of time.	DF method ignores the term frequency which effects on the ranking of the feature.

Table 2.2 *Continued*

Harrag, et al (2010).	Light-Stemming, DF, TF.IDF and (LSI).	(BPNN)	Arabic text collection.	The results indicate that the proposed method achieved high performance in terms of Macro- Average F1 measure for Arabic text classification.	LSI it is a distributional model, so not an efficient representation, when compared against state-of-the-art methods.
(Hong-we, Jian-fang, & Feng, 2010).	TF.IDF MI	SVM	Documents created manually.	The performance of this method is improved in terms of accuracy.	Time for training classifier.

Table 2.2 *Continued*

Pang et. al (2007).	MI and Back-and-forth maximum-match, shortest-path", omni and maximum-probability.	K-NN and SVM	Corpus for Chinese documents.	Some improvement from the new method in term of precision, recall and F1 comparing.	The probability distribution resulting from the GIS algorithm may lead to poor prediction accuracy.
Pang et. al (2007)	MI, MEB", ME, MI, ME, AMI and MI with CE.	SVM	The corpora used in these experiments selected from set of journals.	The best performance is noticed in ME with CE, and the improvement recorded is 2.78% better than Baseline.	Time for training classifier.

Table 2.2 *Continued*

Warintarawej et al, (2011).	DF and MI syllable frequency.		French Larousse thesaurus.	Syllables play important roles for classification model, and performs considerably better than MI.	It was designed to model very slow changes in one set of sounds over time, though, computational (and data) limitations necessitated simulation of a small subset of the entire language.
Wei, Gao and Wu (2010).	DF, IG, MI, Chi, CE and PCA.	Decision Tree, Bayesian, SVM and K-NN	Reuters-21578corpus.	That MI reduces high dimensionality better that the other feature selection methods.	Only Making a comparison between methods.

Table 2.2 *Continued*

Fu, Chen, Gong, & Bie, (2008).	Chi, IG and GR,	NB,BN, AODE,HNB and CNB.	web pages	The AODE and HNB performance are competitive comparing with the other algorithm.	It make a comparison between the most important features selections, so there is no any new work presented.
Pei et al, (2010).	Chi and MI	SVM	Reuters-21578.	Macro F1 is enhanced from 84.6% to 87.7.	It depends crucially on the probabilistic mode.
Xiaoming and Yan (2013).	MI	NB and SVM	The Reuters-21758.	Named text-based word frequency which performs considerably better.	Mutual information method focuses on the correlation between terms and categories, without considering the connections between terms.

Table 2.2 *Continued*

Bleik, et. al (2013).	UMLS database.	NB, SVM, and k-NN	Medical articles only half of the text content.	Results show that the rich graph representation of documents improves the classification performance significant.	The size of otology is effected in term of space and time where the concepts of the ontology is used.
Ajgalik et al., (2013).	WordNet	K-NN and NB classifier.	Document.	Efficient, concise representation of document content.	WordNet has only a limited number of connections between topically related words.
(Shein & Nyunt, 2010).	Ontology	SVM	Software reviews.	Enhance the sentiment classification tasks.	Time for training classifier.

Table 2.2 *Continued*

(Dollah & Aono, 2011).	(AFA) Chi and DF Ontology.	LIBSVM	Biomedical text abstracts from Medline abstracts and OHSUMED.	The results indicate that improved the performance of the proposed works is improved.	The best running time computational complexity of this algorithm is O (n), and the worst case is O (N2), when the taxonomy is flat.
(Xiaoyue & Rujiang, 2009).	Ontologies	SVM	OHSUMED, Reuters-21578 and 20 Newsgroups (20NG) collections.	The results show that the performance of BOC in terms of micro -F1 values and macro-F1 values is better than BOW for all datasets.	The complexity in terms of space and time depends on the ontology, and this work use the whole ontology will be used as class.

Table 2.2 *Continued*

(Wang, McKay, Abbass, & Barlow, 2002).	Ontology	K-NN	Title and abstract from journals in the MEDLINE.	The results show the performance of the proposed methods is improved in terms of accuracy.	When increasing the number of variables, the number of evaluations increases as well. While goal functions with few variables are feasible, optimizations with about twenty variables are usually impractical.
(Zhang & Song, 2006).	VSM fuzzy, ontology.	K-NN	Abstracts from MEDLINE.	The results from this study show that by using small size of keywords, the accuracy dose not degrades.	Only abstract from document.

Table 2.2 *Continued*

(He & Wu, 2006).	ontology	Binary classifier	Abstracts from MEDLINE journal	The experiments from this study indicate that the improvement is significantly achieved in terms of accuracy.	Training each class need time, using the abstract only to reduce the size of feature so many information are ignored.
(Shahi, Issac, & Modapothala, 2012).	"Correlation Feature”,"greedy hill climbing" ,"backtracking facility".	NB		"Greedy hill climbing" method with a "backtracking facility" to search the space of attributes subsets so that the classes candidate will be reduced therefore the performance is improved.	Correlation Feature Selection Slower than univariate techniques and less scalable than univariate techniques and ignores interaction with the classifier.

Table 2.2 *Continued*

(Li & Hu, 2009).	VSM. Hornet 2000	Polynomial Bayesian	Corpus from Sougou Lab	Experiment results from the proposed study shows that the performance classification is enhanced.	The disadvantage of the naive Bayes classifier is that it assumes that all attributes are conditionally independent given the class, while this often is not a realistic assumption.
(Chang & Huang, 2008).	Ontology, Expert, theorem (FCA)	Bayesian classifier	"Electronic Theses and Dissertations System".	This study records 80% effectiveness for some classes and for other classes it records 60%.	Bayes classifier is that the real-world data may not always satisfy the independence assumption among attributes.

Table 2.2 *Continued*

(Khan, Baharudin, & Khan, 2012).	(MFS) ontology	NB	documents downloaded from Internet	The results show that the F1 value is improver for all classes and recoded 85.36%. So that by comparing this work with TF.IDF the improvement reaches 10.93%.	The limitation of this study is the time needed for ontology creation and concept extraction.;l km .

The previous sections explain different studies on selecting the feature vector for text classification task. “Feature selection methods” try to remove the irrelevant features from text for reducing the feature vector size which enhance the accuracy of classification task and decrease the time complexity for learning algorithms. The score for each feature from feature set is calculated then top feature is selected. Different “feature selection methods” were discussed and explained.

The main idea of feature selection method is to select subset from set of feature from training and testing example from training classifiers. The common disadvantage of these methods is that they ignore the interaction with the classifier and each feature is considered independently thus ignoring feature dependencies, the limitation of each one is discussed. Table (2.2) make a comparison between different works used many different methods to reduce the dimension of training set for enhancing text classification task.

Many works try to use ontology for creating features to reduce the size of feature by replacing the terms with concepts from ontology because of removing the common terms and grouping the terms with same meaning. But these studies used traditional classification algorithms to do classification task. This means the same problem in creating the examples from training and testing phases.

/

2.3 Ontology

With the advancement of computer technology, ontologies have been adopted to represent and organize information in the fields of knowledge representation, library science, IR, natural language processing or Internet search engines (Chandrasekaran et al, 1999). The most concise and widely used definition of ontology as used in computer science applications is “a specification of a conceptualization” (Gruber 1993). The definition can be expanded as a formal representation of a body of knowledge formed by a collection of concepts and their relationships describing a particular domain (Gruber 2009). Ontology provides means for much richer representation of concepts with their relationships. In ontology, concepts are represented by individuals, classes and properties (Lord, 2010).

Individual is a real world object, class represents a set of individuals that belong together according to their common properties and property represents a relationship either between individuals or between individuals and data values. A property restriction is a characteristic of a class, meaning that all individuals of a particular class are required to have certain properties with certain value types. A domain of a property is a set of individuals to which the property is applied. A range of a property is a set of individuals that the property has as its value.

Classes may have subclasses (more specific classes) and super classes (more general classes). Hierarchical relationships in an ontology are also referred to as taxonomical, or vertical relationships, while non-hierarchical ones are sometimes called horizontal relationships. Furthermore, the term hyponym is used when referring to more specific relationships and hypernym – for more general, or “is a”, relationships.

An important feature of ontologies is that they describe knowledge in a way that is readable for machines (computers). This characteristic enables knowledge sharing and reuse, information resources can be communicated between either humans or computer software. For these purposes Web Ontology Language (OWL) (Dean et al. 2004) has been developed. OWL is an XML-based semantic markup language for publishing and sharing ontologies. It was designed to be processed by computer applications, and not meant to be presented to humans.

”A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them” (Genesereth & Nilsson, 1987). Each “knowledge base”, “knowledge-based system”, or “knowledge-level agent” is dedicated to certain conceptualization, implicitly or explicitly. “An ontology is an explicit specification of a conceptualization”. It is rented from philosophy, “an Ontology is a systematic account of Existence”.

The knowledge which represent the knowledge_base program is a set of objects and relations among them. So that, in AI, a set of representational terms is defined the ontology program. The definition in such ontology supports the names of entities in the universe of discourse, with a text that describe the “name means” and “formal axioms” that makes the use of these terms well-formed. Officially, ontology can be defined as a “statement of a logical theory”.

2.3.1 Ontology for Text Classification

"Ontology provides a share understanding of a domain of interest" (Uschold & Gruninger 1996). Over the years, it has emerged as a fundamental for "semantics driven modeling" and is adopted for TC. Traditional approaches of TC are easy to implement. Furthermore, datasets in traditional method needs to be changed whenever the classification label changes and this is an arduous and time consuming efforts (Tong & Koller, 1998). To overcome the limitations of traditional text classification models in terms of missing semantic relations between terms and the effect of using example in training classification algorithms, many researchers attempt to use ontologies.

Ontology in text classification is used to solve many main problems on traditional text classification algorithm, particularly high dimensionally, document representation, and classification methods (Agarwal et al, 2012; Dollah & Aono, 2011).

2.3.2 Applications of Ontology

Many applications used different types of ontologies and the following are some of them:

- “EBI’s Experimental Factor Ontology” (EFO): “The EBI’s Experimental Factor Ontology” is used to represent sample variables from gene expression experimental data. EFO collects classes from different reference ontologies and yelids new classes to add more knowledge to reference ontology classes in order to meet querying use cases (Malone et al, 2010).

- Neuroscience Information Framework Standard ontology (NIFSTD): “The NeuroInformatics Framework – NIF (NIF)”, previously named BIRN, have built the NIFSTD ontology. NIF is a dynamic catalogue neuroscience resources created from Web such as data, tools and materials through any connection between the computer and the Internet’ (Gardner et al, 2008).

- Virtual Life Sciences Library (VIVO): 235Library (VIVO): “Virtual Life Sciences Library” which contain 122 information about “courses”, “genomics services”, “faculty”,” departments”, “undergraduate majors”,” graduate fields” anything related to the “Life Sciences” at Cornell (Devare et al, 2007).

- Crop-pest Ontology: The crop-pest ontology was built to facilitate image retrieval in an image collection taken by a scientist who is working on crops and pests in the University of Florida. The collection contains 291 images that shows three crops (soybean, peanut, and cotton) and related insects that cause damage on them. The scope of the crop-pest ontology covers at least the domain knowledge contained by the image collection.

- Crop-biosecurity Ontology Cataloging. An ontology in the “crop-pest domain” has been built which contains concepts and their relationships on crops, related pests like “insects”, “diseases”, “weeds”, “nematodes”, and “mammals”, and pest management subjects like (“integrated pest management”, “chemical control methods”, “biological control”). The ontology was built to work as a system to classify the publication with in EDIS “Extension Digital Information Source” (Howard et al, 2005).

- Food Nutrition and Agriculture Journal is a bibliographical Metadata Ontology that consists of 14 ‘metadata’ concepts and 1800 instances with 3 languages. Metadata for ontology was created using containing relationships between resource attributes such as title, authors and keyword (Sini et al, 2007).

- Food Safety it is a domain ontology for food safety, animal health, and plant health of 1600 concepts in English. Starting from the AGROVOC Thesaurus and specific terminology from web sites and documents (Maloni & Parkinson, 2010; Salokhe, 2006).

2.3.3 Type of Ontology

Several classifications of ontologies have been presented in the literature (Roussey, et al, 2011). Each of them focused on different dimensions in which ontologies can be classified.

2.3.3.1 Classification Based on “Language Expressivity and Formality”

Depending on the expressivity of an ontology (or, in general, of a knowledge representation language), different kinds of ontology components can be defined (concepts, properties, instances, axioms, etc.) .

”Concepts”, “instances” and “properties” are mentioned by one or more symbols. Symbols are terms that humans can recognize it by reading them. And then all the components of ontology are connected via semantic relations. “Semantic relations” connects only concepts, for example, the “location” relationship shows that “city” concept is localized in another concept named “country”. “Instance” relations make a connection between instances. “Instance relations” are instances of “semantic relations”. Some relations between instances can be contextual and cannot be generalized to all instances of their concept.

2.3.3.2 Classification Based on the “Scope of the Ontology”, or on the "Domain Granularity”

The scope of a “local ontology” is specific than the scope of a “domain ontology”. The main difference between the “core ontology” and “domain ontology” is that the concept of “domain ontology” more specific that the while the other contains “foundational concept” of such a domain. “Foundational ontologies” can be represented as Meta ontologies that show the top level concepts or primitives used to define others ontologies. Finally, “general ontologies” are not devoted to a specific domain so that its concepts can be as general as those of “core reference ontologies”.

In the proposed work ontology for specific domain is used, for computer science domain.

2.3.4 Ontology as Classifier

Many researches try to solve the problem of training examples for classification algorithms by using ontology concepts as training set instead of training example that are created manually. With the ontology, there is no need to use traditional text classification algorithm, instead depend on the structure of the ontology to do classification. These researchers replace the terms with set of concepts from ontology then using some technique to find the related class for classification the text. Many studies are represented to classify text using ontology.

A novel text classification and ranking method using ontology is presented in a study titled "Ontology-Based Automatic Classification and Ranking for Web Documents" by Fang, et al., (2007). Documents are represented by a set of terms, while different ontologies are used as classes. In this method, the similarity between the ontology and the document is calculate using "Earth Mover's Distance", while WordNet is used to calculate the similarity between the terms. From this method the document could be classified to a number of different ontologies. For each class the document will be ranked depending on similarity score. To evaluate this method, collection of document downloaded from "http://dmoz.org/" website. The result from this work shows that the accuracy is not high.

"DCSO" is a method proposed by (Chang, 2007) for building a domain ontology and document classification automatically. There are three distinct features of the DCSO. There are three distinct features of the DCSO. First is its automatic construction of the ontology using the theorem of formal concept analysis (FCA). Secondly, the DCSO generates proposition of an XML knowledge-based schema for document storage and quick search and lastly, the utilization of the hierarchy's property of ontology offering the accuracy of document classification. Five hundred and twenty five documents in the area of information management are retrieved from the "Electronic Theses and Dissertations System http: //etds.ncl.edu.tw/ the abs/index.jsp". The behavior of the accuracy for classification with DCSO is well and the searching time for DCSO steadily. Limitation of FCA, it have high execution time and high computational cost which make it infeasible to extract the concepts for large context.

A novel text classification method using ontology is presented in the study of (Fang, Guo, & Niu, 2010). Firstly, weighted terms represent the document and set of different ontologies used as classes. To calculate the similarity between the concepts from ontology and each weighted term from each document, the "Normalized Google Distance" (NGD) is used. Swoogle is used for ontology searching where the ontologies are represented as classes. Experimental results show that the performance of the proposed work is more efficient.

Another work proposed by (Song et al., 2005) proposed a new ontology-based text classification method. In this study, the similarity between the new document and documents already classified using ontology is presented. To evaluate this method, dataset is downloaded from "Yahoo Economy news" collection. The results from this work show that meanings and relationships of document give more accurate results. By using the ontology text class show that by using of ontology to conceptually express the meaning of relationships contained in Web documents and the author suggested an automated document classification method that uses the ontology.

In the study name "Classification of RSS feed news items using ontology" by (Agarwal et al., 2012) proposed an approach which uses "weighted Concept Frequency-Inverse Document Frequency" (CF-IDF) with domain Ontology, for classification of RSS feed News Items. There is no trained classifier required while ontology itself acts as a classifier. The researchers designed ontology based on news industry standards. Evaluation of experimental results reveals that proposed approach gives better classification results.

"Ontology-supported webpage classifier" (Onto Classifier) is proposed by (Lee, Yang, & Hsu, 2008). Onto-Classifier allows users to input some related documents. The system can deal and study the contents of document to extract and calculate the term frequency to classify the document of related scholars. In the proposed work, ontology was combined with text created from web page crawler. The experiments produced a more accurate TC since the meanings. The precision of classification is 67%.

The work by (Manuja & Garg, 2014) propose a self-governed ontology-based approach to classify documents purely in the relevant context of user query combine with SVM classifier. Two branches of the system start working in parallel: one being the collection of relevant documents through a focused web crawler and second being the build-up of seed ontology for experiment. Two parts in this system working at the same time. Firstly, relevant documents are collected through web crawler and then seed ontology is built. The system is applied from scratch there is no manually organized seed ontology being used which is quite encouraging. This study made a comparision with other frameworks and show that a better usefulness of the framework in terms of self- governed learning system. The evaluated parameters are accuracy 86%.

Calvier, Planti´e, Dray, & Ranwez, (2013) defined the semantic description of a class as a vector of concepts from the domain ontology. The proposed work is presented in two steps. During the first step, the system exploits documents indexed by an expert in order to identify the differences between documents of each category. Both documents and their indexes are given to the system during this step. The second step consists provision of a classes to index a new document. PubMed is a wide source of biomedical articles manually indexed using MeSH concepts. In order to avoid font problems and character identification, this study focused on the abstract of the articles which are available in a simple text version.

The work titled "Automatic Topic Identification Using Ontology Hierarchy" by (Tiun, Abdullah, & Kong, 2001) proposes a method of using ontology hierarchy in automatic topic identification. The keywords which are extracted from a given text will be mapped onto their corresponding concepts in the ontology. By optimizing the corresponding concepts, it will pick a single node among the concepts nodes which it believes is the topic of the target text. It extends the ontology by enriching each of its concepts with new concepts using the external linguistics knowledge-based WordNet. The work is interested in extracting out information from the web document based on HTML tag. The node concept can be in a form of one word or more. The result on classification accuracy increased up to 36.5%.

The study name "An ontology-based text processing approach for simplifying ambiguity of requirement specifications" by (Polpinij, 2009) aims to solve the ambiguity problem by proposing a new method through different approaches which are based on ontology. First one is text classification and the other is text filtering. Text classification enhance the classification because of using ontology which give the a shared understanding of the domain of interest while text filters are used to weight abstract requirements in documents. Then, the ontology- based text models is executed with a probabilistic machine learning algorithm NB. Moreover, the researchers used datasets of "20 newsgroups" and 5000 web pages gathered from the

WWW. After testing by F-measure, the empirical results demonstrate that the purposed method may help to provide more effectiveness for simplifying and handling ambiguous in requirements specifications.

Another study by Ha-Thuc & Renders, (2011) takes advantage of the ontological knowledge in the process of text classification. By exploiting the hierarchical to construct a context-aware query for each class, the query is submitted to a web search engine to get relevant documents in that class. Then the researchers propose a hierarchical topic model to extract multinomial distribution over words for each class. The hierarchical topic model takes the relationships amongst classes defined in the hierarchy to exclude noise, identifying really relevant parts in training documents, and to estimate class language models from these relevant parts only. The researchers also propose novel classification algorithm using information propagated both top-down and bottom up when making decisions. The evaluation set consists of a collection of 1130 news items, crawled on the web sites of 4 news agencies "CNN, Reuters, France24 and DW-World". The resulting in terms of the standard and hierarchy-based F-1 measures was 41:3% and 67%.

Another study by (Brank, Mladenić, & Grobelnik, 2010) deals with the problem of classifying textual documents into a topical hierarchy of classes. The approach constructs the coding matrix gradually, one column at a time, with each new column being defined in such a way that the new binary classifier attempts to rectify the most common mistakes of the ensemble of binary classifiers built up to that point. The study also presents systematic experiments on a small dataset which demonstrate that good coding matrices with a small number of columns exist, but are rare. Experiments have shown that SVM can lead to good and accurate models in many problem domains, including text classification where it is now one of the state-of-the-art methods. The Mean Average Precision (MAP) values for k-NN framework and hierarchical SVM framework are 42.40 and 29.94 respectively.

Noh, Seo, Choi, Choi, & Jung, (2003) presented the automated Web page classifier based on adaptive ontology. To extract significant, notable features from a set of terms, the weights of the terms are computed. Then the information gains of the features are calculated for ranking their consequences. In the experiments, the researchers tested OUT Web page classifier and measured its performance in terms of classification accuracy. To implement the automated Web page classifier, set of training tuples are compiled into terms-classification rules using C4.5 NB classification algorithm, CN2, and back propagation learning algorithms. The resulting accuracy of the classification was 95.2%.

The work titled "A New Method for Knowledge and Information Management Domain Ontology Graph Model" by (Liu, He, Lim, & Wang, 2013) described a comprehensive and innovative ontology-based system framework called Knowledge Seeker. The approach adopts a chi-square based statistical learning method to extract and formalize knowledge from a document corpus in the form of the "Domain Ontology Graph" (DOG). In this study experiments are carried out to evaluate the performance and the effectiveness of the proposed method of ontology graph modeling and learning, and the ontological operation. The high performance of the ontology-graph-based text classification method reveals that the ontology graph learning method is highly effective and has successfully generated a set of small sized DOGs that were able to represent domain knowledge. Classification accuracy with 92.3% in f-measure compared with other methods is 86.8% in f-measure for the term-frequency–inverse-document-frequency approach.

Table 2.3

Literature summary on reducing dimension ontology as classifier for text classification task

AUTHOR	TECHNIQUES	CLASSIFICATION ALGORITHM	DATASET	STRENGTH	WEAKNESS
(Fang, Guo, Wang, & Yang, 2007).	"Earth Mover's Distance", WordNet.	Different ontologies.	From "http://dmoz.org/" website.	EMD Naturally extends the notion of a distance between single elements to that of a distance between sets, or distributions, of elements.	The result from this work shows that the accuracy is not high.
(Chang, 2007).	Formal concept analysis.	The hierarchy's property of ontology.	The hierarchy's property of ontology.	The utilization of the hierarchy's property of ontology offering the accuracy of document classification.	Limitation of FCA if have high computational cost.

Table 2.3 *Continued*

(Fang, Guo, & Niu, 2010).	"Normalized Google Distance" (NGD).	Swoogle	Text corpus in question.	Experimental results indicate that the performance of the proposed method is more efficient.	NGD distance did not take into account the context in which the words co-occur.
(Song et al., 2005).	similarity	Ontology	"Yahoo Economy news".	The results from this work show that meanings and relationships of document give more accurate results.	Time need to calculate similarity.
(Agarwal et al., 2012).	" (CF-IDF)	With domain Ontology.	RSS feed News.	Evaluation of experimental results reveals that proposed approach gives better classification results.	For short document.

Table 2.3 *Continued*

(Lee, Yang, & Hsu, 2008).	some related documents.	Ontology Tf.idf.	Web page.	The precision of classification is 67%.	Time consuming, where each ontology is set of document.
(Manuja & Garg, 2014).	ontology	With SVM classifier tf.idf.	Documents through a focused web crawler.	The evaluated parameters are accuracy 86%.	In this work tf.idf used to calculate the weight of the features.
Calvier, Planti´e, Dray, & Ranwez, (2013).	Expert MeSH	Exploits documents indexed by an expert in order to identify the differences between documents	PubMed is a wide source of biomedical articles “abstract”.	The efficiency of the system depends strongly on the index quality.	Time consuming in create the feature of each category because it is manually.

Table 2.3 *Continued*

(Tiun, Abdullah, & Kong, 2001).	WordNet	The sense-tagger system	HTML tag.	Classification accuracy increased up to 36.5.	The sense-tagger system which resulted the unsuccessful mapping between the keywords and the concepts.
(Polpinij, 2009).	ontology	NB	"20 newsgroups" and 5000 web pages.	The empirical results demonstrate that the purposed method may help to provide more effectiveness for simplifying and handling ambiguous in requirements specifications.	The limitations of the rule-based taggers are that they are non-automatic, costly and time-consuming.

Table 2.3 *Continued*

Ha-Thuc & Renders, (2011).	Ontological information propagated both top-down and bottom up	Multinomial distribution over words	News items	The resulting in terms of the standard and hierarchy-based F-1 measures was 41:3% and 67%.	The main difficulty is that fitting the model requires evaluating probabilities given by multidimensional normal integrals,
(Brank, Mladenić, & Grobelnik, 2010).	coding matrix gradually	SVM for k-NN framework.	Small dataset	That SVM can lead to good and accurate models in many problem domains, Time complexity, space complexity.	Time complexity, space complexity.

Table 2.3 *Continued*

Noh, Seo, Choi, Choi, & Jung, (2003).	weights of the terms information gains.	C4.5 NB classification algorithm, CN2, and back propagation learning algorithms.	OUT Web page	The resulting accuracy of the classification was 95.2%.	Time consuming for large space.
(Liu, He, Lim, & Wang, 2013).	a chi-square	Innovative ontology-based	Graph modeling and learning.	Classification accuracy with 92.3% in f-measure.	A word which occurs frequently in many categories will have a low Chi.

From these works, the main thing is that by using ontology as classifier the classification performance is improved. Some works used ontology as class for topic classification while other used concept as class. For the studies used ontology as class, the main problem is that the time and space complexity is very high. Because there is need to process all the concepts from these ontologies. Especially when there is need to classify the document to different ontologies for topic classification there is need to process all concepts from all different ontologies. More problem is that these works depends on calculating the frequency of concepts to make decision and ignoring the effect of semantic relations between these concepts. Ontology can describes the semantic relation between concepts in specific domain. So that there is need to study the effect of these relations between concepts which represent the document. Another problem is that all there studies treat the different document representation same. There are many different types of document needed to be classified and are different in size and structures. By using concept frequency for classification task, feature vector for all these different document will be created in the same way. Especially for structured document there is important thing is that the way in writing these type of document should be used as main factor. Table (2.3) show different works classified the text document using ontology as classifier.

2.4 Summary

Numerous approaches have been applied to classify the documents. The approaches presented in this section suffer from dealing with missing the relations between the terms which effect on the information. Another important problem in text classification is the high dimensionality of training set which effect on the performance of the text classification approach. Ontology is considered as an approach that can be used to solve the problem of semantics between terms according to its structure. The concept of ontology can be replaced instead of training set for classification the text. Furthermore, integrating the document structure for creating feature vector with ontology (concepts and relations) is an approach that can be studied to optimize the feature creation and dimension reduction for text classification.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter presents the framework and methodology for this thesis to develop an algorithm to classify text into set of classes using ontology concepts and semantic relation. Section 3.1 discusses the research framework. Section 3.2 gives the dataset development, while section 3.3 presents the proposed methodology used to create feature for the texts to be classified, and then Section 3.4 present the enhanced algorithm to solve the dimensionality problem caused by training set. After that Section 3.5 present the enhanced algorithm to classify this document to set of classes. Finally, Section 3.6 presents the evaluation of this study. Finally Section 3.7 presents the summary of this chapter.

3.1 Research Framework

The research framework starts with the data set development. After that, the enhanced text classification algorithms are proposed for classifying the text into a number of classes which solve the problem of high dimensionality. The third phase deals with the proposed new enhanced technique to calculate the important class to be classified. Figure 3.1 depicts the phases of the research framework.

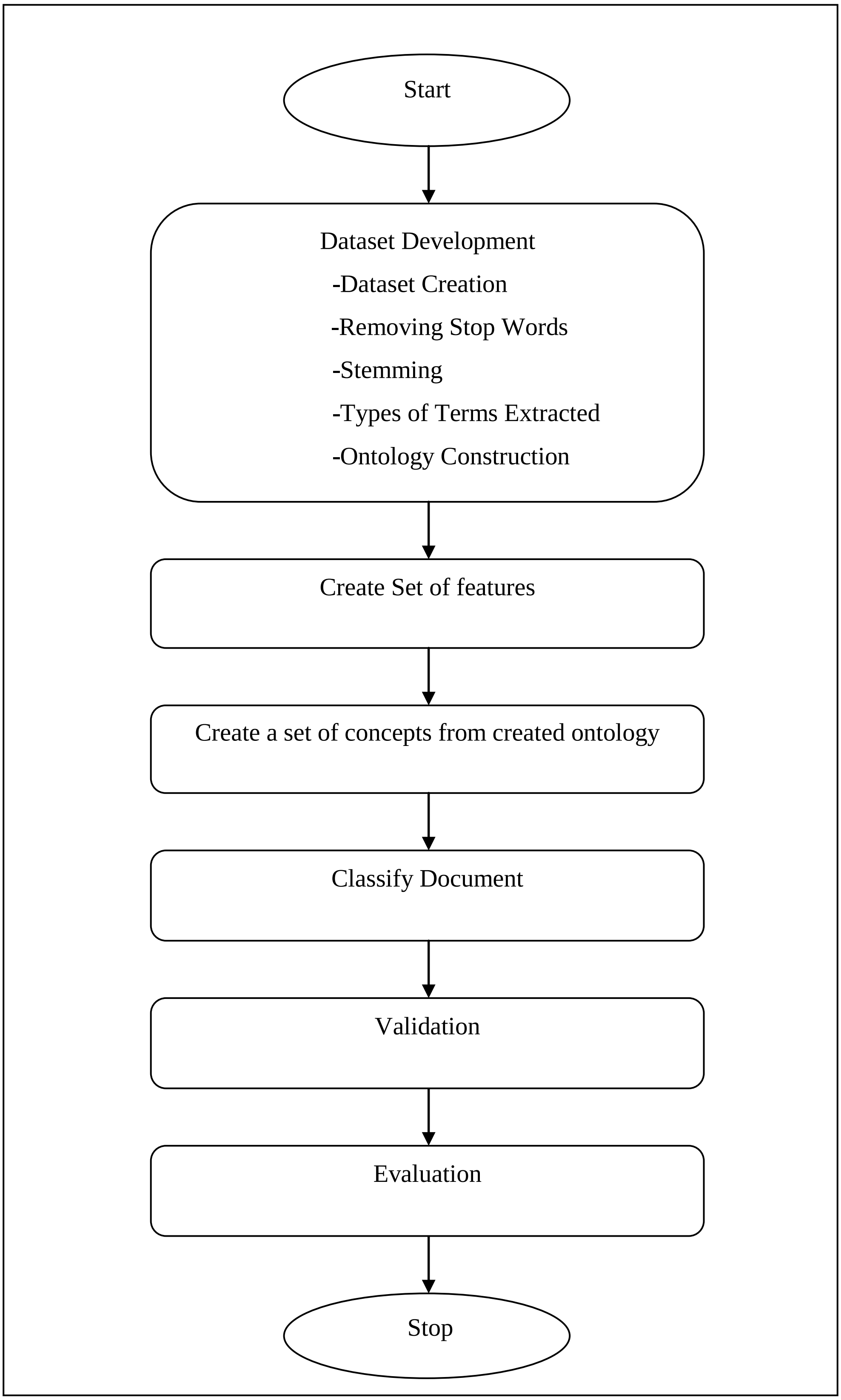


Figure 3.1. Proposed Research Architecture

Each phase of the framework has its methodology. The research methodology is the route used to solve the research problem. It may be understood as a science of study on how research is carried out scientifically. The research methodology relates to the logic behind the methods used in the context of the research study and explains the used of one particular method or technique rather than another in order to evaluate the research results. The research methodology is divided into a retrospective, perspective, experimental and non-experimental studies (Kumar, 2011).

This thesis proposes an enhanced text classification of algorithms based on combined ontology with document representation. The proposed algorithms are required to be evaluated with other approaches. In order to do this evaluation, conducting experiment using dataset is needed. Therefore, an experimental methodology approach is adopted in this thesis.

3.2 Dataset Development

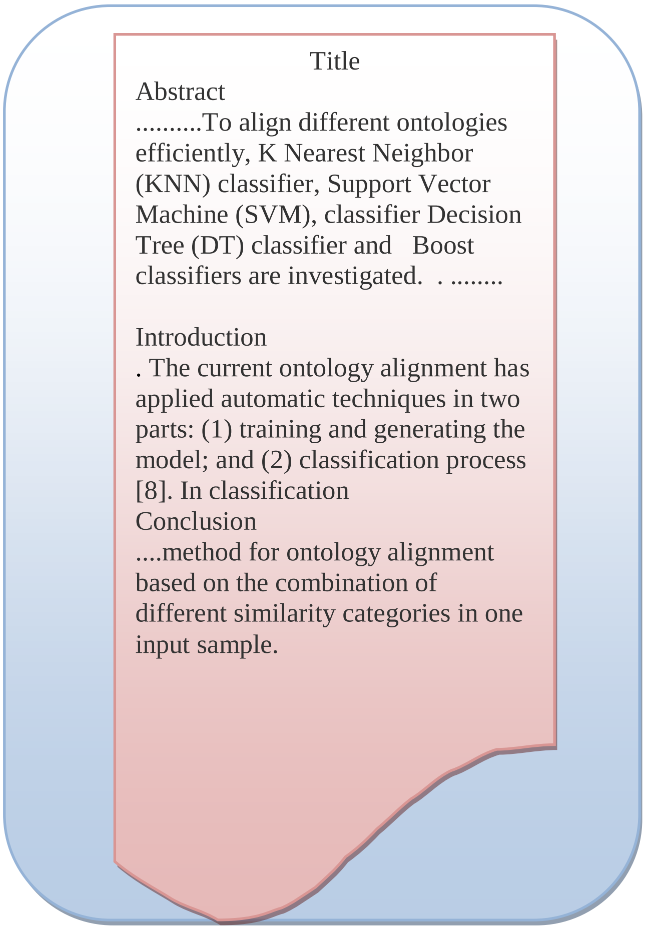
For text classification algorithm, first the datasets are created for evaluation in term of precision, recall, f- measure, and accuracy. Then these datasets will be preprocessed to have a pure data for more accurate classification results. Preprocessing involves removal of unnecessary terms, group word with the same root, and defining the type of terms to be extracted.

3.2.1 Dataset Creation

Many different types of dataset are used in text classification such as web pages, newspaper, scientific paper, books. This proposed work deals with the scientific paper document. Several studies have been conducted on TC using scientific papers (Dollah & Aono, 2011) and (Nuipian et al, 2011) used datasets with abstracts while (Zhanguo et al, 2011) used the whole document.

The scientific publications follow a structured format of writing but are stored in unstructured file format. The structured writing format can be helpful in IR. A scientific document is divided into sections. This component of the scientific publication can be valuable in generating semantically enriched context-aware metadata (Ahmed, Khan, Latif, Masood, & Elberrichi, 2008). Semantic structure that represents the relationship between the concepts and the component, helps in improving search preciseness and minimizes the information loss (Ahmed, Khan, Latif, Masood, & Elberrichi, 2008). Figure 4.5 presents the generic structure of a scientific publication.

The DT_TREE model by (Rizvi & Wang, 2010) downloads different datasets from the Internet. Each dataset downloaded by DT_TREE consist of a collection of 150 documents. Each dataset is comprised of 3 different groups where the size of each group is 50 documents. The main part of the proposed work is the structure of document to be classified. So there is a need to download document which contain all the information from this document. There is no standard dataset from whole scientific paper .So that the models used this type of document download their dataset from internet.



Title

Abstract

..........To align different ontologies efficiently, K Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM), classifier Decision Tree (DT) classifier and Boost classifiers are investigated. . ...

Introduction

. The current ontology alignment has applied automatic techniques in two parts: (1) training and generating the model; and (2) classification process [8]. In classification

Conclusion

....method for ontology alignment based on the combination of different similarity categories in one input sample.

Figure 3.2. Scientific paper structures

The first step in this thesis is creating dataset. A collection of documents that represents a dataset from five different digital libraries are downloaded. The size of each dataset is 150 scientific paper documents divided into three groups depending on three different queries. By selecting the top 50 document after representing the query to the library search engine, the dataset is created. Each group from dataset is decomposed into positive set and negative set. Maximum Normalized term frequency is used to calculate the weight of each term in the query for the document in the dataset (Manning, Raghavan & Schütze, 2008). After find the weight of these terms, threshold value is used to make a decision if the document in the positive where the weight is greater than or equal threshold or negatives if the weight values less than the threshold value. Where the threshold value is 0.1.

150 documents are retrieved from each dataset, where each dataset contained three different groups from different queries. The length of queries is short and the size is between 2 to 3 keywords. For each query 50 documents are collected. First query is “Support Vector machine” or “ontology”, second query is “Classification” or “Support Vector machine” the last query is “Machine Learning”, “classification” or “clustering “. For the first group, the number of classes is two. For the second group, the number of classes is two. Finally for the last group, the number of classes is three. Table 3.1 shows the query used to create dataset. First column represents the name of query to create. Second column represent the number of document download for each query. And the last column represents the number of classes for each dataset.

Query 1:” classify the document into two different classes Support Vector machine or ontology this mean ta document could be classified into one class or two different classes Thus

Query 1= [Support Vector machine ∨ ontology ∨ (Support Vector machine ∧ ontology)].

Query 2:" classify the document into two different classes Classification or Support Vector machine. This mean the document could be classified to one class or two different classes Thus

Query2= Classification ∨ Support Vector machine ∨ (Classification ∧ Support Vector machine)].

Query 3:" classify the document into three different classes of machine learning, classification or Clustering. This means that a document could be classified into one class, two or three different classes".

Thus

Query 3= [machine learning ∨ classification ∨ Clustering) ∨ (machine learning ∧ classification) ∨ (machine learning ∧ clustering) ∨ (classification∧ clustering) ∨ (machine learning ∧classification ∧ clustering)].

Table 3.1

Query for creating dataset and its classes

Query	**Number of Document**	**Number of Classes**
"Support Vector machine" or "ontology"	150	2
"Classification" or "Support Vector machine"	150	2
"Machine Learning","classification"and "clustering "	150	3

The First dataset is downloaded from the IEEE digital library. The IEEE Explore is a scholarly research database that indexes, abstracts, and provides full-text for articles and papers on computer science, electrical engineering and electronics. The database mainly covers material from "the Institute of Electrical and Electronics Engineers" (IEEE) and the Institution of Engineering and Technology. The IEEE Explore database contains over two million records.

The Second dataset is downloaded from "the Association for Computing Machinery" (ACM) digital library. ACM is the largest educational and scientific computing society, and its database delivers periodical scientific publication particularly in the field of engineering, computing and science.

The Third dataset is downloaded from the CiteSeerx digital library. The CiteSeerx is digital library that provides access to thousands of literature in the field of computer and information science. CiteSeerx also provide resources such as algorithms, data, metadata, services, techniques, and software that can be used to promote other digital libraries. CiteSeerx has developed new methods and algorithms to index PostScript and PDF research articles on the Web.

The Fourth dataset was obtained from the World Scientific (WS) database. The WS publishes approximately 500 new titles annually in various fields. The final dataset is obtained from Google Scholar which provides a search of scholarly literature across many disciplines and sources, including theses, books, abstracts and articles.

Table 3.2

Datasets for the proposed work

	Query 1			**Query 2**			**Query 3**		
DATASET	Number of Sections	Year	Positive group-Negative group	Number of Sections	Year	Positive group-Negative group	Number of Sections	Year	Positive group-Negative group
IEEE	4-12	2006-2013	(45-5) (42-8)	5-10	2004-2013	(40-10) (44-4) (9-41)	4-10	2005-2013	(43-7) (37-13)
ACM	5-10	2003-2013	(37-13) (25-25)	5-10	2004-2013	(44-6) (30-20) (15-35)	4-13	2000-2013	(40-10) (34-16)
CiteSeerx	4-10	1997-2013	(34-16) (41-9)	5-12	1990-2012	(39-11) (40-10) (32-18)	5-12	1995-2002`	(48-2) (35-15)

Table 3.2 *Continued*

World Scientific	5-13	2005-2014	(34-16) (36-14)	5-10	2001-2013	(37-13) (32-18) (23-27)	5-10	2002-2013	(39-11) (6-44)
Google scholar	2002-2012	2002-2012	(47-3) (38-12)	5-11	2000-2011	(43-7) (35-15) (24-26)	5-11	2000-2011	(44-6) (30-20)

Table 3.2 present the characteristic of the dataset used in the experiments for the work. First column represents the name of digital library where the dataset was loaded. Second column represents the minimum and maximum number of sections for all documents the dataset. The third column represents the years of publication. And the last column represents the number of document in positive and negative group depending on the criteria represented in section 3.2.

3.2.2 Removing Stop Words

Stop words or stop lists are lists of words that are removed prior to or after the processing of text relaying on their level of usefulness in a given context. In text classification one of the important steps is to eliminate the stop word from the text. The removal of stop words improves information retrieval and searching by ignoring words that usually appear in every document and thus are not helpful in distinguishing documents from each other. Removing stop words reduces the index size, number of distinct words in the index, and therefore saves space and time. Examples of some stop words in English include "the', "and", "a", "of". Some search engines contain a single multilingual stop list while others contain a stop list for each language.

In this study set of stop-words is downloaded from the website "http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop" (Genkin, Madigan, & Lewis, 2007).

3.2.3 Stemming

Other important step in text classifications is stemming. Stemmers are basic elements in query systems, indexing, web search engines and information retrieval systems (IRS). It is the process of reducing words to their roots or stem in order for the text processing to index and recognize them as the same word. For example, the words dies, and died would be recognized as one word die. In the field of text mining, stemming is used to group semantically related words to reduce the size of the dictionary (feature reduction). Stemming can be viewed as a recall-enhancing device or a precision-enhancing device.

Porter Stemmer is an algorithm proposed to remove common morphological and inflextual endings from English words. Mainly used in information retrieval systems, the stemmer works for normalization process. Developed in Java, C and Perl, the Portal Stemmer is widely adopted in many applications today "http://tartarus.org/martin/PorterStemmer/".

3.2.4 Types of Terms Extracted

This step extracts terms from each document for classification. The terms are further classified into two sub-categories namely the key word, and the key phrase. A key word is a single word, while a key phrase (multi word) is a means a set of separate words that build a phrase. The recognition of key phrase allow the user to focus on accurate topics because they are more precise and more specific to a particular scientific domain than key word. The key phrase identification makes it possible to either index texts with a high degree of precision or to guide the user in his information search. Phrases are more complete than words in syntactic and semantic structures. Key-phrases have clearer meanings than hot terms and thus are better representations for topics, for example, “nature language processing” vs. “nature”,” language” “processing”. Key-phrase is consists of several related terms or words. Many works used N-gram to extract key phrases from document.

In the proposed work. Single term is easy to extract because it is any sequence of letters while the Key-phrase extraction is more complex because of the sequence of steps applied to find the key phrase. After text preprocessing, the pure text from the input document and word sequences is obtained. N-gram based evaluation metrics is used for automatic key phrase extraction where N=3.

3.2.5 Ontology Construction

In the proposed work, Wikipedia is used as a source to create the ontology.

3.3 Create Set of Features

Using the ontology created from Wikipedia, the feature set that represents the document will be a set of concepts from ontology instead of terms. Therefore there is a need to map the term from a document to the ontology concepts. The mapping is by corresponding the keywords from document with concepts from ontology. The algorithm will filter all terms which do not have any word match to the concepts of ontology. This thesis studies the effect of document structure in creating the feature of the document. Two algorithms were proposed for the creation of features depending on ontology concepts and document representation. First algorithm Concept Feature Vector (CFV) used the concept frequency with respect to the structure of the document. Where the document is decomposed into sections and each sections will be used as distinct document. For each section normalized concept frequency is calculated for each concept. This is followed by obtaining the total the weights of the concepts from their sections. The second algorithm is the Structure Feature Vector (SFV) that deals with the semantic relation between concepts with respect to the structure of document. The SFV decompose a document into number of sections and each section will be treated as document. Section 4.3 in chapter four explains these two algorithms in detail.

3.4 Creating Set of Concepts from Ontology

The Ontology Based Text Classification (OBTC) algorithm is proposed to solve the problem of high dimensionality. In OBTC, the concept of the ontology will be used as a class. For each class select related concept set from ontology which have semantic relation with this concept to calculate the similarity. The similarity measure is then calculated between the features vector which represent the document and the related concept set created from ontology for specific class by using conditional probability. Section 4.4 in chapter four explain this algorithm in detail with examples.

3.5 Classify Document

To classify the text document into set of classes, feature vector creation algorithm is combined with a text classification algorithm. The combined enhanced algorithm addresses the semantic relation between terms from the structured documents. It also solves the problem of high dimensionality caused by training sets. This is achieved by using concepts of ontology which affects the performance of the classification.

Two classification algorithms are proposed in this work. First algorithm Concept Feature Vector for Text Classification (CFV_TC) and the second one is Structure Feature Vector for Text Classification (SFV_TC). The proposed algorithms are tested on a dataset created from different digital libraries which contains set of documents dealing with computer science. Section 4.5 in chapter four explain these two algorithms in detail with examples.

A comparison with works RSS (Agarwal et al, 2012) and (Dollah & Aono, 2011) in term of precision ,recall, F-measure and accuracy will be performed.

Net-Beans IDE and Net-Beans Platform were used to implement the proposed algorithms. Protégé application was used to create ontology for the computer science domain. Experiments were performed on an Intel® Core ™ 2 Due CPU T5750, running at 2.00 GHz with 4.00 GB RAM and 32-bit operating system.

3.6 Validation

Two TC models are used to make a comparison. The first one is RSS classification model (Agarwal et al., 2012) which try to classify the text into set of related classes using ontology. The main part is to use concepts from ontology as classes. This work try to calculate the concept frequency inverse document frequency (CF.IDF) for each concepts from document. And add more weight for the concepts if this concept appear in title. No training set is created in this work. And no training phase need to train the model. RSS model is chosen for validation because there is need to find the limitation of the most known technique used to calculate the weight of concepts in text classification approach (CF.IDF). Furthermore, each concept is defined as class and no training set is defined from examples.

The other model used SVM classification algorithm combined with ontology to classify the document (Dollah & Aono, 2011). The concept weight is calculated using tf.idf, replacing each terms with its related concepts from ontology then Chi Square feature selection used to reduce the dimension of training set. After that make a classification.

3.7 Evaluation Measures

Text classification is typically evaluated using performance measures from information retrieval. Common metrics for text categorization evaluation include recall, precision, accuracy and error rate and F-measure.

The metrics for binary-decisions are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

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

$$\text{F-measure} = \frac{2 * \text{TP} + \text{TN}}{\text{TP} * \text{FN}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{N}}$$

Where, TP is true Positive, FP is False Positive, TN is True Negative and FN is False Negative, respectively.

In classification, the simplest method for calculating an aggregate score across categories is to average the scores of all binary tasks. The resulted scores are called macro-averaged recall, precision, accuracy (Yang & Joachims, 2008).

Figure 3.3 shows the Confusion Matrix for Text Classification Evaluation.

		Actual class (expectation)	
		Correct Category/ Relevant	Wrong Category/ not relevant
Predicated Class "Observation"	Correct category/ Relevant	TP "True positive" correct result	FP "False positive" unexpected result
	Wrong category/ not relevant	FN "False negative" Missing result	TN "True negative" absence of result

Figure 3.3. Confusion Matrix for Text Classification Evaluation (Classification of RSS feed news items using ontology)

3.8 Summary

This chapter presents the methodology of the study. The study involved 6 phases: dataset development, Create Set of features, Create a set of concepts from created ontology, Classify Document, Validation and the last one is Evaluation. First clarifies the framework of the proposed work and the datasets that are used in this study. This is followed by the preprocess steps in the data set to make pure data from these documents. This is followed by an explanation of the enhanced algorithm for feature vector creation to create sets of suitable features which can describe the document. This chapter then presents the explanation of the enhanced text classification algorithm to which solve the high dimensionality problem caused by the training set. It also presented the enhanced ontology based classification algorithm to classify the text to set of classes. The equations for calculating the precision, recall, f-measure, and accuracy are explained in detail. Finally, the summary of this chapter is presented in the last section.

CHAPTER FOUR
ENHANCED ONTOLOGY BASED TEXT CLASSIFICATION ALGORITHM FOR SCIENTIFIC PAPER

4.1 Introduction

In this thesis two algorithms are proposed based on ontology structure, concepts and semantic relations which connect them, while the features created from structured document. These algorithms are proposed in order to obtain enhanced precision, recall and accuracy for text classification. These two algorithms can handle the problem of high dimensionality which effect on text classification efficiency by filtering all terms that can degrade the precision of text classification accuracy.

The direction of these algorithms is to find the feature of the document to be classified using the concepts from the ontology with respect to the structure of this document in calculating the weight of the features. The other direction is to solve the high dimensionality problem. Each concept from the ontology will be treated as a class. Furthermore, the set of concept is created from ontology, where each concept from this set has semantic relation with the class.

Figure 4.1 presents the generic architecture for the proposed text classification algorithm. As shown in Figure 4.2, the proposed framework starts with dataset cleaning by removing stop words and stemming, then creating feature vector by using enhanced algorithms and pass it to the classification algorithm using ontology.

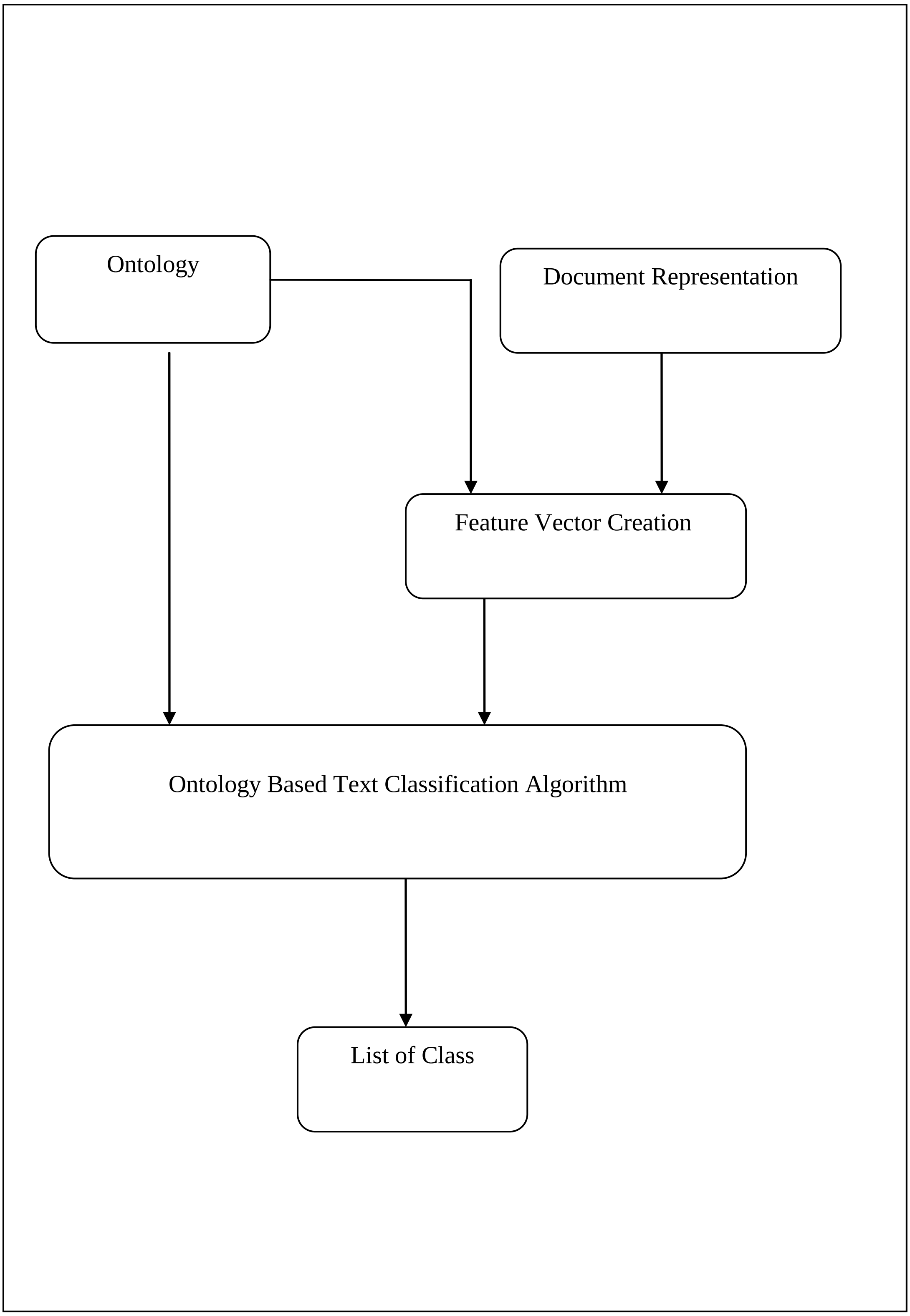


Figure 4.1. General Architecture of the proposed work

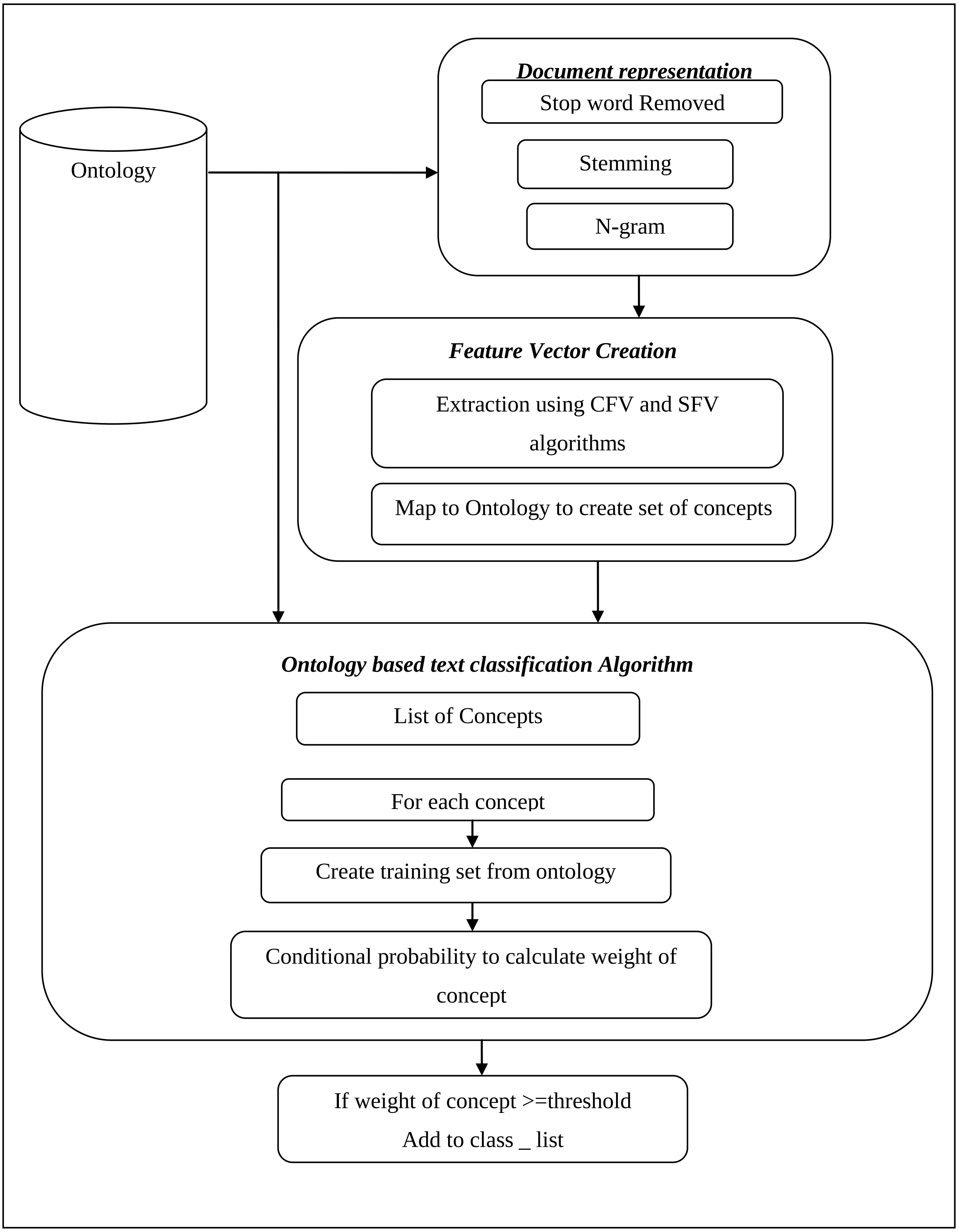

Ontology
Document representation
Stop word Removed
Stemming
N-gram
Feature Vector Creation
Extraction using CFV and SFV algorithms
Map to Ontology to create set of concepts
Ontology based text classification Algorithm
List of Concepts
For each concept
Create training set from ontology
Conditional probability to calculate weight of concept
If weight of concept >=threshold
Add to class _ list

Figure 4.2. The proposed Text Classification framework

4.2 Ontology Structure

The ontology structure proposed in this study involves a collection of concepts as shown in Figure 4.3. Each concept has its semantic relation that connects it to other concepts. Particularly, the 'Is-a' relationship connects two concepts if there is any semantic relationship between them. Generally, the ontology used in the proposed model is light, represented as a graph. Based on this, each concept is presented as a set of related concepts (input and output). This structure is used in the extraction and classification parts. Figure 4.4 shows the representation of ontology in RDF form.

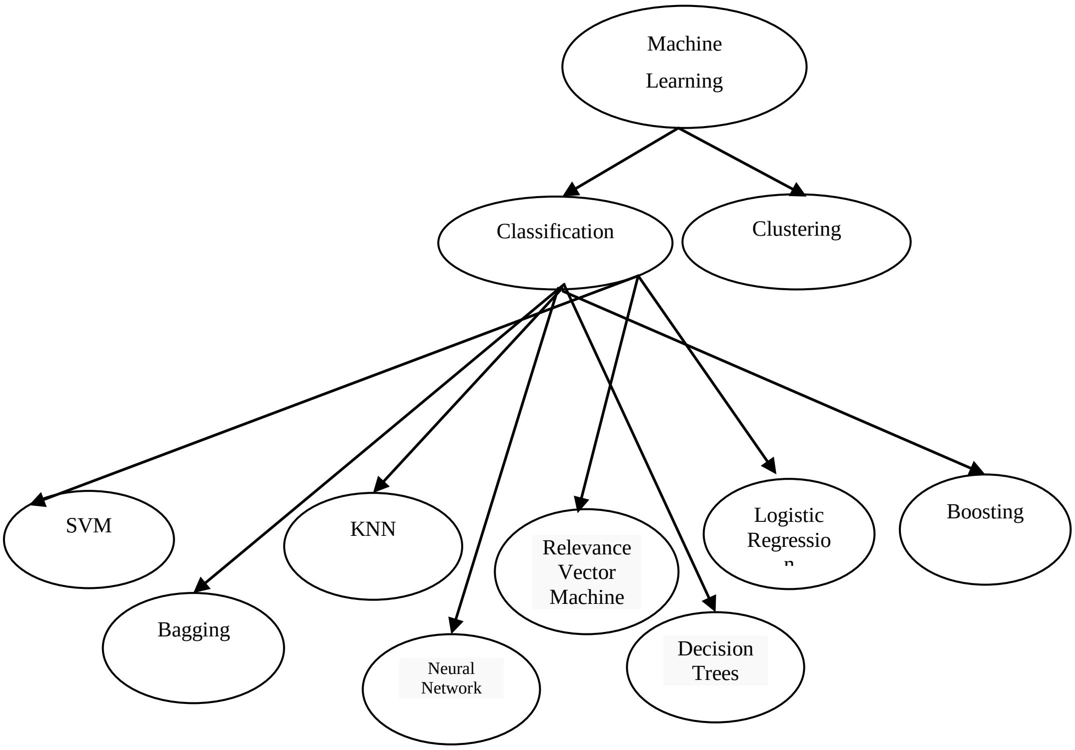


Figure 4.3. Ontology for Computer Science Domain (Classification concept)

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
 .
 .
 .
 rdf:about="http://somewhere/D:/classification">
    <vcard:FN>decision tree</vcard:FN>
    <vcard:FN>supervise learn</vcard:FN>
    <vcard:FN>rocchio algorithm</vcard:FN>
    <vcard:FN>neural network</vcard:FN>
    <vcard:FN>nearest nigh</vcard:FN>
    <vcard:FN> naiv baye t</vcard:FN>
    <vcard:FN>relevance vector machine</vcard:FN>
    <vcard:FN>boostin</vcard:FN>
    <vcard:FN>support vector machine</vcard:FN>
    <vcard:FN> Logistic Regression </vcard:FN>
    <vcard:FN>machin learn</vcard:FN>
    <vcard:FN>bagging</vcard:FN>
  </rdf:Description>

  .
  .
  .
  .
</rdf:RDF>
```

Figure 4.4. Ontology for classification concept from Computer Science ontology Science Domain (RDF)

4.3 Feature Vector Creation Algorithm for Text Classification

For solving the semantic problem in text classification task, this research automatically extracts structured information from a scientific paper to create feature vector in order to help with more accuracy in terms of precision and recall. This structured information is represented based on the sections of the scientific paper where the terms are extracted from. These section are “abstract”, “introduction” and “conclusion”. These terms should be at the beginning of statement.

4.3.1 Proposed Concept Feature Vector (CFV)

The Concept of Feature Vector (CFV) is an enhanced ontology based text classification algorithm which create a set of features from the document for classification. The input is a scientific paper and the output of this algorithm is a set of weighted concepts. These concepts will be used in the classification phase as classes.

Based on the general representation of the scientific papers which is a collection of sections, the following steps are followed to extract information for classification phase:

- *Let D_sec (Doc) = {s1, s2…..} as the representation of document Doc according to section, si represents a section si in document Doc.*

- *Let Doc = {t1, t2…..tn} as the representation of document Doc, ti represents a term in document Doc.*

- *Let O = {c1, c2,………..cm} as the representation of ontology O, cj represents a concept cj in ontology O.*

For each term from document Doc, search the ontology O to find its match concept, to create sets of concepts which represent Doc.

• *Let C = {c1,c2,………..cm} as set of candidate concepts which represent the document Doc,*

Where ti=cj and ti belongs to Document Doc, cj belongs to ontology O.

According to the first proposed study, the weight is given to the concept which is treated in the next step as a class:

For each candidate concept ci, from the concept list C, Normalized Concept Frequency NCF (ci,S_k) is calculated, as shown in equation 4.1 below:

$$NFC(C_i, S_k) = \frac{CF_{i,k}}{Max(CF_k)} \qquad (4.1)$$

Where, $CF_{i,k}$ is the frequency of concept ci in section sk. Max (CF_k) is the max term frequency at section sk.

To calculate the weight of the concept for the document Doc, the summation of all weights for this concept, derived from different section will be calculated.

The coefficient should be given to each section. According to the proposed work the coefficient is the same for all sections because this study focus on distribution of concepts in different sections.

By using equation 4.2, the final weight of each the candidate concept is calculated from:

$$W(C_i) = \sum_{k=1}^{n} NFC(c_i, s_k) * Coef \tag{4.2}$$

Where n is the number of sections representing the document, k is the section number, and Coef is equal to 0.333 for each section. According to the proposed work the document is decomposed into three section, so that the threshold value is 0.333 for each section Figure 4.5 shows the proposed CFV algorithm.

```
The proposed Concept Feature Vector Creation algorithm CFV
Input document D, Ontology O
Output set of concepts, set of weights
Begin
    Preprocess Document D to extract set of terms T
    Stop word removing
    Stemming
    N-gram
    Create set of single and compound terms Using N-gram
    For each term t in T
        Map to Ontology O and create set of concepts C
    End
    Decompose the document into sections
    For each Concept $c_i$
        For each section $sec_j$
            Calculate the Normalized Concept Frequency of concept
            $c_i$ using equation 4.1
        End
        Calculate the weight of concept ci using equation 4.2
    End
End
```

Figure 4.5. The proposed Concept Feature vector (CFV) Algorithm

This algorithm will be used to calculate the weight of all concepts extracted from document for classification phase.

Furthermore, document Doc is classified by algorithm CFV. The first step in the CFV algorithm is to create set of terms presented from line (1) to line (4). Preprocess the terms extracted using stemming rules to find root of the terms and remove the stop word from list predefined. In line (6), line (7) ad line (8) the terms mapped to ontology to create set of concepts. The document is decomposed into sections at line (9) to calculate the weight of each concept with respect to the sections. Iteratively calculate the weight each concept. Starting from line (10) to line (15). for each concept from line (11) check section one by one and calculate the weight in term of section by check each section separately using Normalized terms frequency in line (12) example 4 (and (5).

Example 1: Algorithm CFV

let's take the following Document Doc (Doc shown in Figure 4.6).

Abstract= [To align different ontologies efficiently, K Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier and AdaBoost classifiers are investigated].

Introduction= [The current ontology alignment has applied automatic techniques in two parts: (1) training and generating the model and (2) the classification process [8].]

Conclusion= [method for ontology alignment based on the combination of different similarity categories in one input sample.]

This document is represented as set of sections. "Find sections from this document" first remove stop word and stem these terms using rules after that N-gram to find single and compound terms:

Remove stop word

Abstract=[align , ontologies efficiently, K Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier, AdaBoost classifiers ,investigated.]

Introduction= [current ontology alignment, applied automatic techniques, training, generating, model, classification process.]

Conclusion= [method, ontology alignment based, combination, similarity categories, input sample.]

Stemming

This step is to find the root of word using set of defined rules to group the word with the same root.

Abstract= [align, effici ontlogi, nearest neighbor classifi, support vector machin classifi, decision tree classifi, adaboost classifi, investig]

Introduction= [curr ontlog align, appli automat techniqu, train, gener, model, classif process]

Conclusion=[method, ontlog align base, combin, similar categori, input sampl]

N-gram

Abstract=[align, effici, ontlogi, effici ontlogi, nearest, neighbor, nearest neighbor, classifi , neighbor classifi , nearest neighbor classifi, support, vector, support vector, machine, vector machine, support vector machin, classifi, machin classifi, vector machin classifi, support vector machin classifi,, decision, tree, decision tree, classifi, tree classifi, decision tree classifi, adaboost, classifi, adaboost classifi, investig]

Introduction= [curr, ontlog, curr ontlog, align, curr ontlog align, ontlog align, appli, automat, appli, automat, techniqu, appli automat techniqu, automat techniqu, train, gener, model, classif, process, classif process]

Conclusion= [method, ontlog, align, ontlog align, base, ontlog align base, align base, combin, similar, category, similar categori, input, sampl, input sampl]

Let's take the following Ontology O (Ontology shown in Figure 4.3) and set of concepts

"Find all concepts in Ontology O":

Ontology=["machine learning, classification, clustering, Nearest Neighbor, Support Vector Machine, Neural Network, Bagging, Relevance Vector Machine, Regression"s].

Thus after stemming the ontology is:

ontology=[machin learn, classifi, clust, Nearest Neighbor, Support Vector Machin, Neur Network, Bag, Releva Vector Machin, Regres].

Map to ontology concepts

Abstract= [ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi, adaboost classifi]

Introduction= [ontlog, classif]

Conclusion= [ontlog, category which equal"category"]

The set of concepts from document doc is:

Document_concept= [ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi, adaboost classifi, ontlog, classif, ontlog, category which equal"category"]

Let's take the "classification" as class and calculate its weight after stemm "classification"

After stemming

"classification"="classifi"

For abstract NFC (classi) =4/4=1 where the frequency of classifi is 1 and t/he max frequency is for classifier and equal 4.

For introduction NFC (classi) = 1/1 where the frequency of classifi is 1 and t/he max frequency is equal 1

For conclusion NFC (classi) =1/1 where the frequency of classifi is 1 and t/he max frequency is for classification (categories) and equal 1.

"Calculate the summation of all weight from different sections"

In abstract NCF (classifi) =4/4

In introduction NCF (classifi) =1/1

In conclusion NCF (classifi) =1/1

*The final weight (classifi)=4/40*0.333+1/10.333+1/1*0.333 =1*

4.3.2 Proposed Structure Feature Vector (SFV)

In the first part of our work, we have proposed an algorithm (CFV) which selects the concept as feature to represent the document depending on the frequency of the concept with respect to the structure of the document. In the second part, we enhance the performance by using the semantic relation between concepts on the document by proposing a new algorithm. A new algorithm called Structure Feature Vector (SFV) tries to create sets of feature from the document to be classified.

In the first step, as shown in the first classification algorithm, the document is filter from stop-word and stems the terms of its root using porter stemmer. Then all single and compound terms from documents are extracted using N-gram where N=3.

Based on the general components of the scientific papers, the following rules are followed:

• *Let Doc = {*sec_1, sec_2*.....} as the representation of document Doc,* sec_i *represents a section i in document Doc.*

Where, the document is decomposed into a specific number of sections.

Each section is a collection of terms representing the information in this section.

• *Let Doc_Sec (secj) = {t1, t2.....} be as the representation of document Doc, where ti represents a term in document Doc at section secj.*

The ontology as presented in the previous algorithm is a collection of concepts for a specific domain.

• *Let O = {c1, c2,…cj……..cm} as the representation of ontology O, where cj represents a concept in ontology O.*

The terms from each section to its matched concept from ontology create sets of concepts to each section. Mapping the terms to the concept is the finding of equal concept.

• *Let Doc_Sec_Con = {c1, c2…..ck} as the representation of document Doc, Where ti=ck and ti represents a term in document Doc at section Secj, ck concept from ontology O.*

The output of this algorithm is a number of sections where each section is set of concepts extracted from this section.

```
The proposed Structure Feature Vector SFV algorithm
Input document D, Ontology O
Output set of concepts per sections
Begin
1   Preprocess Document D to extract set of terms T
2   Stop word removing
3   Stemming
4   N-gram
5   Decompose the document into sections
6   For each section $sec_j$
7       Create set of single and compound terms
8       For each term t in T
9           Map to Ontology O and create set of concepts C
10      End
11  End
End
```

Figure 4.6. The proposed Structure Feature Vector SFV algorithm

Figure 4.6 shows the proposed Structure Feature Vector algorithm (SFV). Furthermore, document Doc is classified by algorithm SFV. Thus, first set of terms extracted from line (1) to line (4). Try to preprocess the terms extracted using stemming rules to find root of the terms and remove the stop word from list predefined. In line (5) decompose the document into number o sections. From line

(6), for each section from the document created in line (7) create set of term. From line (8) to line (9) the terms mapped ontology to create set of concepts. Instead of calculating the weight of these concepts as in algorithm (CFV) there is no need to do such calculation only set of concept are created.

Example 2: Algorithm SFV

let's take the following Document Doc (Doc shown in Figure 4.6).

Abstract= [To align different ontologies efficiently, K Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier and AdaBoost classifiers are investigated.]

Introduction= [The current ontology alignment has applied automatic techniques in two parts: (1) training and generating the model and (2) the classification process [8].]

Conclusion= [method for ontology alignment based on the combination of different similarity categories in one input sample.]

This document is represented as set of sections. "Find sections from this document" first remove stop word and stem these terms using rules after that N-gram to find single and compound terms:

Remove stop word

Abstract=[align , ontologies efficiently, K Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier, AdaBoost classifiers ,investigated.]

Introduction= [current ontology alignment, applied automatic techniques, training, generating, model, classification process.]

Conclusion= [method, ontology alignment based, combination, similarity categories, input sample.]

Stemming

Abstract= [align, effici ontlogi, nearest neighbor classifi, support vector machin classifi, decision tree classifi, adaboost classifi, investig]

Introduction= [curr ontlog align, appli automat techniqu, train, gener, model, classif process]

Conclusion= [method, ontlog align base, combin, similar categori, input sampl]

N-gram

Abstract=[align, effici, ontlogi, effici ontlogi, nearest, neighbor, nearest neighbor, classifi , neighbor classifi , nearest neighbor classifi, support, vector, support vector, machine, vector machine, support vector machin, classifi, machin classifi, vector *machin classifi, support vector machin classifi,, decision, tree, decision tree, classifi, tree classifi, decision tree classifi, adaboost, classifi, adaboost classifi, investig]*

Introduction= [curr, ontlog, curr ontlog, align, curr ontlog align, ontlog align, appli, automat, appli, automat, techniqu, appli automat techniqu, automat techniqu, train, gener, model, classif, process, classif process]

Conclusion= [method, ontlog, align, ontlog align, base, ontlog align base, align base, combin, similar, category, similar categori, input, sampl, input sampl]

Let's take the following Ontology O (Ontology shown in Figure 4.3) and set of concepts

"Find all concepts in Ontology O":

Ontology=["machine learning, classification, clustering, Nearest Neighbor, Support Vector Machine, Neural Network, Bagging, Relevance Vector Machine, Regression"].

Thus after stemming the ontology is:

ontology=["machin learn, classifi, clust, Nearest Neighbor, Support Vector Machin, Neur Network, Bag, Releva Vector Machin, Regres"].

Map to ontology concepts

Abstract= [ontlogi, nearest neighbor, support vector machin, decision tree, adaboost classifi]

Introduction= [ontlog, classif]

Conclusion= [ontlog, category which equal"category"]

4.4 Ontology Based Text Classification Algorithm (OBTC)

An ontology based classification algorithm is proposed to classify the document into set of classes where each class is a concept from ontology for direct access. For each candidate class (concept), related concept set from ontology is created where these concepts should have a relation with the candidate class. So that only ontology concepts will be used to classify the document and no need to create training set for classification task. This algorithm start after extracting a set of candidate concepts from the feature creation algorithms, the classification algorithm will test each concept sequentially to make a decision if this concept is greater than threshold. According to the proposed work the document is decomposed into three section, so that the threshold value is 0.333 for each section.

The Ontology-Based Text Classification algorithm proposed in the second part is to find the weight of the candidate concept using similarity measure between related concept set for the candidate concept and set of feature extracted from document. The input is a set of features from the document needed to be classified and the ontology concept set. The output is a set of concepts as classes. The algorithm starts with creating a related set of concepts from ontology for the each candidate concept, where this related concepts contains all concepts connected directly to the candidate concept. The ontology used in this study is undirected graph. This study suggests that each concept of ontology for specific domain is treated as a class for direct access and not topic classification, so the weight is calculated for the concept (class) itself.

For Text classification the set of rules will be used are as follows:

• Let R = {r (c_1, c_2),…. r (c_i, c_j) …..r(c_1, c_m) } represents the sets of relations where c_i, c_j are concepts from ontology O and have semantic relation r. where c_i is a concept extracted from document Doc., as shown in equation 4.3 below:

Concept-tree $(c_i) = \{c_1,..c_j,...c_n)$ (4.3)

Where r (c_i, c_j) in R, and c_i is a concept extracted from document Doc.

Concept-tree is the concept set created for a concept (class) c_i from ontology.

After creating the related concept set, the classification phase starts, where a comparison is made between a set of concepts from input document to be classified with a related concept set for each candidate concept.

Conditional probability is used to calculate the value of relatedness between the document and this candidate concept. The best method used to calculate the similarity between two samples is similarity measures. Many different measures are used in this area. Lin's (1998) suggest a similarity measure which tries to find the semantic relation between two concepts from word-net using conditional probability. This measure gives a good result in finding the most related concepts in query expansion measure as compared with the other techniques, as shown in equation 4.4 below:

$$SIM(c_i, c_j) = \frac{2\log p(lso(c_i, c_j))}{\log p(c_i) + \log p(c_j)} \tag{4.4}$$

Where log p (lso (c_i,c_j)) is the least common concepts between the two concept c_i,c_j to calculate the similarity between them, p (c_i) is the weight of concept c_i.

So, we used this calculation in the proposed study to find the similarity between the document concepts and related concept set for each concept from the candidate set, as shown in equation 4.5 below:

$$SIM(c_i, d) = \frac{2\log p((c_i, d))}{\log p(c_i) + \log p(d)} \tag{4.5}$$

Where: $P\ (c_i,\ d) = (c_i \cap d)$ (4.6)

$$P\ (c_i) = |c_i| \tag{4.7}$$

$$P\ (d) = |d| \tag{4.8}$$

Where P (c_i, d) is the common concept between documents d and related concept set of concept c_i. P (c_i) is the number of concepts that construct the related concept set of concept c_i. P (d) is the number of concepts which represent the features of document d.

The output of this algorithm is the weight value of the candidate concept (class).

Figure 4.7 shows the proposed OBTC algorithm.

```
The proposed Ontology Based Text Classification Algorithm (OBTC)
Input Set of set of concepts C, feature set Feature_set, Ontology O
Output set of Class_List
For each Concept $C_{candidate\_class}$ from concept set C
    Generate training set from ontology TRAINING_SET:
    If there is Semantic relation between $C_{candidate\_class}$ and $C_{concept}$
        ADD ( TRAINING_SET,$C_{concept}$) where $C_{concepts}$ belong to Ontology O
    END
    Find similarity measure between Feature_set and TRAINING_SET  Sem_Measure using equation 4.5
End
If (Sem_Measure >=threshold)
    Add (Class_List, $C_{candidate\_class}$)
End
End
```

Figure 4.7. The proposed text classification Ontology Based Text Classification Algorithm

This algorithm try to create related concepts set from ontology and calculate similarity measure. The input are set of concepts C and ontology O. From line (1) to line (5) steps used to create related concept set from ontology for each concept from C each concept is checked one by one check the concept. Creating new set at line (4) which contain all concepts from ontology have direct connection with candidate concept. At line (6) similarity measure is calculated between the related concept set created from previous lines with feature from document.

Example 3: Algorithm OBTC

let's take the following Document Doc (Doc shown in Figure 4.6).

For each Concept C $_{candidate\_class}$ from concept set C

Let's take the "classification" as class and calculate its weight

after stemm "classification"

class ="classifi"

Let's take the following Ontology O (Ontology shown in Figure 4.3) and set of concepts

"Find all concepts in Ontology O":

Ontology= [machine learning, classification, clustering, Nearest Neighbor, Support Vector Machine, Neural Network].

Thus after stemming the ontology is:

Ontology =[machin learn, classifi, clust, Nearest Neighbor, Support Vector Machin, Neur Network].

Generate training set from ontology TRAINING_SET:

If Semantic relation between C candidate_class and Cconcept

TRAINING_SET(classifi)= [machin learn, classifi, Nearest Neighbor, Support Vector Machin, Neur Network, Bag, Releva Vector Machin, Regres].

Find similarity measure between Feature_set and TRAINING_SET Sem_Measure using equation

$$SIM(c_i, d) = \frac{2 \log p((c_i, d))}{\log p(c_i) + \log p(d)}$$

Feature set is set of concepts extracted from document

let's take the following Document Doc (Doc shown in Figure 4.6).

Set of concepts is

Feature_set = [ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi , adaboost classifi, ontlog, classif,ontlog, category which equal"category"]

*SIM=2*log (length (Feature_set) ∩ (TRAINING_SET))/ log (length (Feature_set)) + log (length (TRAINING_SET))*

*SIM=2*log (4)/log (6) + log (7)*

4.5 Combine Feature Vector Creation Algorithm with Text Classification Algorithm

Two different algorithms are proposed to classify the documents by combing feature creation vector algorithms and text classification algorithm. First algorithm is CFV_TC developed by combining CFV algorithm for feature creation and OBTC for text classification, while the second algorithm is SFV_TC which combines SFV algorithm for feature creation and OBTC for classification. The next section will explain each proposed algorithm with its steps.

4.5.1 Proposed Concept Feature Vector for Text Classification CFV_TC Algorithm

The first proposed algorithm aims to classify the text into set of concepts based on using CFV algorithm to select the feature set which represents the document to be classified. Then, using a text classification algorithm we proposed OBTC for classifying this document into set of classes.

As explained in the CFV algorithm, the input is document and ontology concepts, while the output are sets of concepts. According to this proposed algorithm the output is a feature which describes the document to be classified with set of weight. The output from the first step is then input to the proposed text classification algorithm OBTC, where the input are feature set, ontology concepts and the output is a set of classes.

```
For each concept $c_i$ from feature set C (d) represents document D
 If w ($c_i$) > threshold
          Create related concept set semantic_set ($c_i$)
```

Where w (c_i) is the weight of the concept c_i calculated from CFV algorithm.

The similarity measure between the related concepts set created from OBTC and a set of feature selected from a document is calculated for each candidate concept.
For each concept c_i from feature set C represents document D.

Create related concept set semantic_set (c_i) using equation (4.3)
If the weight of the concept is greater than threshold then call text classification algorithm OBTC.

By using the equation 4.9, find the common concepts between the related concept set and document concept C (d):

$$Common\_concept\ (c_i,\ d) = |\ C\ (d) \cap semantic\_set\ (c_i)\ | \tag{4.9}$$

Calculate the Similarit_measure between concept c_i and C (d) feature set by using equation 4.10:

Similarit_measure $(c_i, d) = 2 * |common\_concept(c_i, d)| / |C(d)| + |semantic\_set(c_i)|)$ (4.10)

Where |common_concept (c_i,d)| is the size of common_concept (c_i).

|C (d) | is the size of the feature set of document D.

|semantic_set (c_i)| is the size of the related concept set of concept c_i|.

The final weight which is the similarity measure compared with a threshold value. If this value greater than this threshold, then the concept will be selected as a class.

For each c_i belong to C (d):

Set_Class(d) =$\{c_1,c_2,\ldots\ldots,c_i\ldots c_n\}$ (4.11)

Where Similarit_measure (c_i, d) > = threshold.

Figure 4.8 shows the proposed CFV_TC algorithm.

```
The proposed CFV_TC algorithm
Input document D, Ontology O, set of concepts
Output set of class_set
Begin
CFV algorithm (D, O, C, W)
2       For each ci in C
3         If W ($c_i$)>threshold
4           Call classification algorithm OBTC ($c_i$,O, S)
5           Calculate the weight of each concept ci in S W_concept
                Using equation (4.10)
6           If W_concept >threshold
7               Add_class_list (ci)
8           End
9        End
10End
End
```

Figure 4.8. The proposed CFV_TC algorithm

First, line (1) call algorithm CFV which create set of concepts as feature from defined , then for each concept from this set iteratively do steps from line (2) to line (9). These steps try to check if the concept is related enough to the document or not depending on predefined threshold. In line (4) another algorithm is called (OBTC), to create related concept set from ontology for each concept. Then calculate the similarity measure between these created set and concept set from document at line (5). If the value of similarity measure between the feature vector and set of concepts is greater that predefined threshold this concept will be added to the class set else ignore this concept.

Example 4: Algorithm CFV_TC

let's take the following Document Doc (Doc shown in Figure 4.6).

Let's take the following Ontology O (Ontology shown in Figure 4.3) and set of concepts

"Find all concepts in Ontology O":

Ontology=["machine learning, classification, clustering, Nearest Neighbor, Support Vector Machine, Neural Network, Bagging, Relevance Vector Machine, Regression"].

Thus after stemming the ontology is:

ontology=[machin learn, classifi, clust, Nearest Neighbor, Support Vector Machin, Neur Network, Bag, Releva Vector Machin, Regres
Call CFV algorithm (D, O, C, W)

The output from CFV algorithm is set of concepts extracted from document and weight for each concept.

Document_concept=[ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi , adaboost classifi, ontlog, classif,ontlog, category which equal"category"]

For each ci in C
Let's take the "classification" as class and calculate its weight after stemm "classification"

After stemming

“classification”=”classifi”

The weight of class (“classifi”)=1

If W (c_i)>threshold
Weight(classifi)>0.333
Where threshold =0.333

Call classification algorithm OBTC (c_i,O, S)

The input for OBTC algorithm are set of concepts from document Feature_set and TRAINING_SET for class as presented in appendix 3 and the output is weight for the class

Calculate the weight of each concept ci in S W_concept Using equation (4.10)

Similarit_measure (c_i, d) =2 |common_concept (c_i, d)|/|C (d)| +| semantic_set*

*SIM=2*log(length(Feature_set)∩(TRAINING_SET))/log(length(Feature_set))+ log(length (TRAINING_SET))*

*SIM=2*log(4)/log(6) + log(7)*

4.5.2 Proposed Structure Feature Vector _Text Classification (SFV_TC) Algorithm

The second proposed text classification algorithm is SFV_TC. This algorithm depends on two steps similar as CFV_TC algorithm. First, create a set of features to represent the document to be classified using SFV, and then classify the document using OBTC algorithm.

To create the feature vector, the input is document and ontology concept set. The output of this algorithm will then be passed to the next algorithm OBTC. For the classification part of the OBTC text classification algorithm used, where the input is a set of feature of the document from the first part and ontology concept set while the output is a set of classes.

In SFV algorithm, each section will be treated as the document, so that the similarity measure will be calculated for each section alone. To find the weight of each concept, the summation of all these values will be calculated.

So that for each concept c_i in feature set C (d) of document d:
Create related concept set semantic_set (c_i) using equation (4.3)
For each section Sec_j belongs to document d where

$$Sec_j = |\{c_1,c_2,\ldots c_n\}| \tag{4.12}$$

Finding the common concepts between the semantic_set (c_i) and sec_j:

$$Common\_concept\ (c_i,sec_j) = |Sec_j \cap semantic\_set\ (c_j) \tag{4.13}$$

Calculating the similarity measure between c_i and Sec_j

Similarit_measure (ci, Sec_j) =2*|Common_concept (c_i, secj)|/Sec_j| +| semantic_set (c_i)| (4.14)

Where | Common_concept (c_i, secj)| is the size of Common_concept between c_i and sec_j.

|Sec_j| is the size of section Sec_j from document d.

| semantic_set (c_i)| is the size of related concept set of concept c_i.

To calculate the final weight of each concept, the summation of all weights from all sections will be calculated.

Weight (c_i) =∑ ((Similarit_measure (c_i, Sec_j)) *Coef) (4.15)

Where the coef is 0.333

If the weight of concept weight c_i is greater than the threshold, then added to the class set.

Class_set={c,c2,…cn}

Where Weight (c_i) >=threshold.

Figure 4.9 Show the proposed SFV_TC algorithm.

```
The proposed SFV_TC algorithm
Input document D, Ontology O
Output set of concepts as classes_set
Begin
1 Call feature creation algorithm SFV algorithm (D, O, C, S)
2       For each $c_i$ in C
3         For each $sec_j$
4             Call classification algorithm OBTC ($c_i$,,O,S, W)
5             Calculate the weight of each concept ci in C using equation
                (4.14)
6         End
7       Calculate the weight of each concept W_class$c_i$ C using equation
         (4.15)
8       If W_classi >threshold
9           Add_class_list($c_i$)
10      End
11    End
End
```

Figure 4.9. The proposed SFV_TC algorithm

First, line (1) call algorithm SFV which create set of concepts as feature from defined, Document. This algorithm instead of calculating the weight of each concept, just decompose the document to set of section and each section will be separated document and create another set C contain all concept from the whole document. In line (2) each concept from C candidate concept do the line (2) iteratively do the steps from line (3) to line (10). In line 3 check all the sections one by one. In line (4) OBTC algorithm is called to create the related concepts set from ontology for

Concept candidate concept .in line (5) calculate the similarity between the related concepts and set of concepts at this section. Then in line (7) find the summation of all these values from different section.

Example 5: Algorithm (SFV_TC)

let’s take the following Document Doc (Doc shown in Figure 4.6).

Let’s take the following Ontology O (Ontology shown in Figure 4.3) and set of concepts

“Find all concepts in Ontology O”:

Ontology=[“machine learning, classification, clustering, Nearest Neighbor, Support Vector Machine, Neural Network, Bagging, Relevance Vector Machine, Regression”].

Thus after stemming the ontology is:

ontology=[machin learn, classifi, clust, Nearest Neighbor, Support Vector Machin, Neur Network, Bag, Releva Vector Machin, Regres].

Document-concept is set of concepts extracted from ontology

Document_concept=[ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi , adaboost classifi, ontlog, classif,ontlog, category which equal”category”]

Call SFV algorithm (D, O, C, W)
The output from SFV algorithm is number of sections and each section is set of concepts.

Abstract= [ontlogi, nearest neighbor, support vector machin, decision tree, adaboost classifi]

Introduction= [ontlog, classif]

Conclusion= [ontlog, category which equal"category"]

For each ci in document_concept
Let's take the "classification" as class and calculate its weight after stemm "classification"

After stemming

"classification"="classifi"

The weight of class ("classifi") =1

For each sec$_j$

Call classification algorithm OBTC (c$_i$*,,O,S, W)*
Calculate the weight of each concept ci in C using equation

Similarit_measure (ci, Sec$_j$) =2*|Common_concept (c$_i$, secj)|/Sec$_j$| +| semantic_set

For abstract section

Abstract= [ontlogi, nearest neighbor, classifi, support vector machin, classifi, decision tree, classifi, adaboost classifi]

*SIM=2*log (Abstract)∩(TRAINING_SET))/log (length (Abstract)) +log (length (TRAINING_SET))*

*=2*log (5)/log (6) +log (9)*

For introduction section

Introduction= [ontlog, classif]

*SIM=2*log (Introduction) ∩ (TRAINING_SET))/log (length (Introduction)) + log (length (TRAINING_SET))*

*=2*log (1)/log (2) +log (9)*

For conclusion section

Conclusion= [ontlog, category which equal"category"]

*SIM=2*log (Conclusion) ∩ (TRAINING_SET))/log (length (Conclusion)) + log (length (TRAINING_SET))*

*=2*log (1)/log (2) +log (9)*

4.6 Summary

This chapter presents the algorithms that used to classify documents to set of related classes using ontology. First the ontology which is used as a backbone in the proposed work is described. After that, the first algorithm of feature vector creation with the Equations and Figures are clarified. This is followed by the second enhanced algorithm of structure feature vector creation and related Equations with Figures. After that, an enhanced text classification algorithm with the related Figures and Equations is explained in detail. Last, two enhanced algorithms for classifying the text which combines the two feature vector creation proposed and text classification algorithm are also presented in detail with its equation and figures. Finally, the summary of the chapter is presented.

CHAPTER FIVE
EXPERIMENTAL RESULTS

The proposed algorithms were tested on five different datasets downloaded from different digital libraries and repositories.

The performance of the proposed algorithms is evaluated by comparing it with RSS Classification (Agarwal et al., 2012) algorithm which try to classify the text to set of classes where each class is a concept from ontology; this will lead to, direct access and compared with work presented by (Dollah & Aono, 2011) that classifies the document using SVM classifier and ontology.

Three different queries of classes are used in evaluation these works as presented in chapter three.

5.1 Result and Analysis

In this section the results from three different algorithms is presented and analyzed in detail. Tables (5.1 - 5.10) summarizes the performance statistics for the three algorithms, namely Concept Feature Vector for Text Classification (CFV_TC) algorithm Structure Feature Vector for Text Classification (SFV_TC) algorithm and the RSS classification algorithm. The threshold value in the proposed work is 0.333. The number of sections are three in the proposed work (abstract, conclusion, other). Therefore, the weight of each section is equal 0.333.

A comparisons are done among the different downloaded dataset presented in this works. This comparison to find the effect of using different dataset downloaded from different digital libraries with different size as presented in table (3.2). The type of data on digital libraries play an important role in creating dataset. This mean that if the digital library is large so the number of related document using query submitted are large, while if the size of digital library is small the number of related documents are small. Therefore, for large size digital library the precision and recall are high. For Google Scholar digital library 160 million documents as of May 2014, for ieee digital library the 3,861,202 (December 2014), for CiteSeerx has over 4 million documents with nearly 4 million unique authors and 80 million citations, for World Scientific publishes published 120 journals in various fields and 500 titles new titles a year and finally for ACM digital library the number of records are 2,351,822.

The evaluation of these works are depicted in Table 5.1. And Figure 5.1 present the classification evaluation of the first proposed algorithm CFV_TC in terms of precision, recall, F-measure and accuracy to show the result from different dataset using CFV_TC algorithm.

Table 5.1

Evaluation of CFV_TC

Dataset	**Precision**	**Recall**	**F-measure**	**Accuracy**
IEEE	0.8660	0.2383	0.8449	0.7828
Google Scholar	0.9356	0.3935	0.8125	0.7571
CiteSeerx	0.9418	0.3980	0.7589	0.7742
ACM	0.9372	0.4687	0.8082	0.7685
World Scientific	0.7080	0.4995	0.5661	0.7828

From Table 5.1 the data set obtained from the CiteSeerx digital library shows the best value in terms of precision when compared with the other datasets. The worst one is the World Scientific dataset. The other datasets are approximated. In terms of recall, the best one is evident from the World Scientific. This means the number of missing document is not very high. In terms of precision and recall World Scientific still poses the least value.

The F-measure which is the relation between the precision and recall is best seen in IEEE; this is because the precision and recall for this data set are better compared to the others.

The accuracy which is the relation between the numbers of documents classified correctly with the size of dataset is best seen in IEEE and World Scientific.

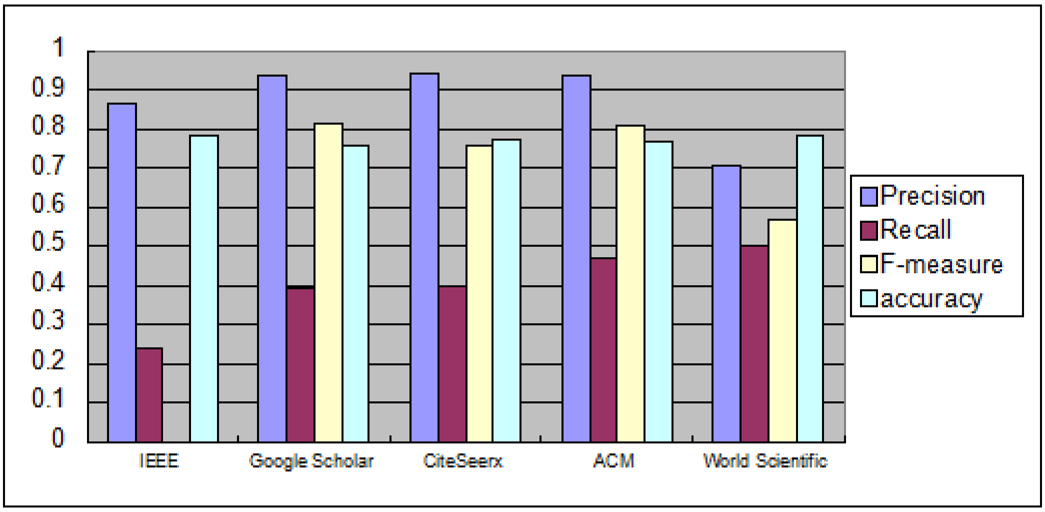


Figure 5.1. The evaluation of the first proposed algorithm CFV_TC

From the Figure above, it is clear that IEEE has the best value in terms of precision while in Citseerx, the precision and recall are varies. The World Scientific dataset has the lowest value in terms of both precision and recall as well as in f-measure. In terms of accuracy the results are close for all datasets. In terms of accuracy, the best value is evident from the IEEE and World Scientific.

Results of evaluating these classification are depicted Table 5.2 and Figure 5.2 present the classification evaluation in terms of precision, recall, F-measure and accuracy for the SFV_TC algorithm to show the results from different dataset using SFV_TC algorithm.

Table 5.2

Evaluation of SFV_TC

Dataset	Precision	Recall	F-measure	Accuracy
IEEE	0.8821	0.9128	0.8972	0.8542
Google Scholar	0.8962	0.8886	0.8924	0.8228
CiteSeerx	0.9477	0.7627	0.8452	0.8200
ACM	0.9158	0.8670	0.8907	0.8428
World Scientific	0.8420	0.5285	0.6494	0.8542

From the table 5.2, the data set from digital library CiteSeerx shows the best value in terms of precision compared to the other datasets, and the worst one is the World Scientific dataset.

In terms of recall, the best one is the IEEE. This means the number of missing document is not very high. The F-measure which is the relation between the precision and recall is seen in IEEE, CiteSeerx and ACM; this is because the precision and recall for these datasets are good compared to with the others. The World Scientific is the lowest value in terms of both precision and recall as well as in f-measure.

In terms of accuracy, which is the relation between the numbers of documents classified correctly with the size of dataset, the best values are the IEEE and World Scientific.

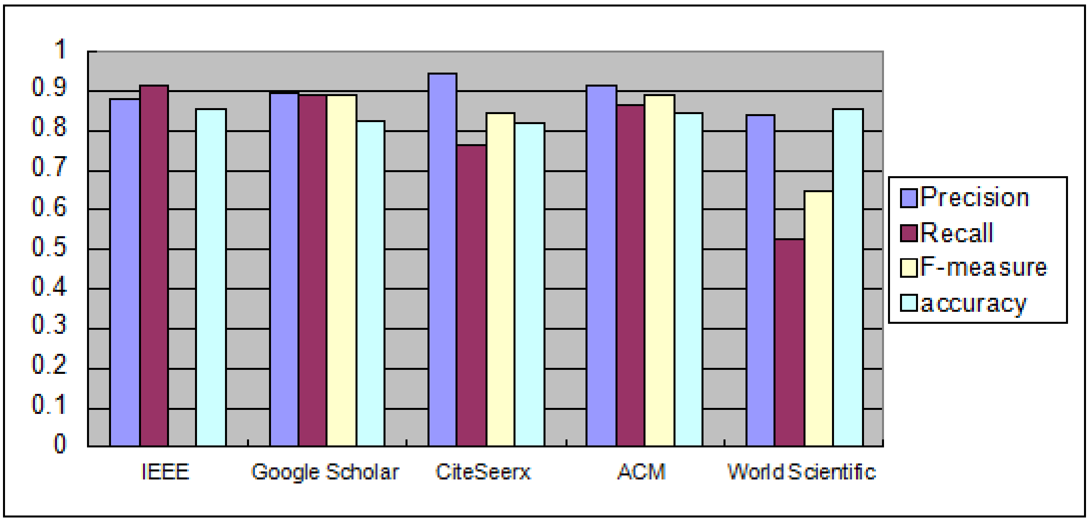


Figure 5.2. The evaluation of the second proposed algorithm SVF_TC

The result from figure 5.2 shows that the best recall is for IEEE. The best values in term of precision is recorded from Citeseerx dataset. The best dataset in terms of precision, recall and F-measure is the GOOGLE dataset. The lowest value in term of precision, recall and f-measure is a World Scientific dataset. In terms of accuracy, the best value is evident from the IEEE and World Scientific.

Results of evaluating these classifications are depicted Table 5.3 and Figure 5.3 present the classification evaluation in terms of precision, recall and F-measure of the RSS Classification model in 2012.

Table 5.3

Evaluation of RSS

Dataset	Precision	Recall	F-measure	Accuracy
IEEE	1	0.5904	0.7424	0.7114
Google Scholar	1	0.5292	0.6921	0.6228
CiteSeerx	1	0.6089	0.7569	0.7400
ACM	1	0.5820	0.7358	0.6942
World Scientific	0.9257	0.3338	0.4907	0.7857

From table 5.3, four datasets have the top values in terms of precision. Only World Scientific records a weak value. This is due to splitting the dataset into related and not related document groups. When creating the dataset the document is distributed into two group related and not related. The criteria used to makes decision is tf.idf. This means same method as CF.IDF. From this we can see the precision is very high.

In terms of recall all datasets have very weak results and the best one is for Citeseerx dataset. And the worst one is the World Scientific dataset. The value f threshold in proposed work is different, so that this threshold effect on the result. In terms of F-measure, IEEE and CiteSeerx are very convergent, and the best compared to the other datasets. The f-measure values depends on the values of precision and recall so that the best result in precision and recall will give the best result in f-measure.

In terms of the best values in term of accuracy is recorded from World Scientific dataset. The accuracy depends on the number of related document classified correctly and the number of document from not related group document.

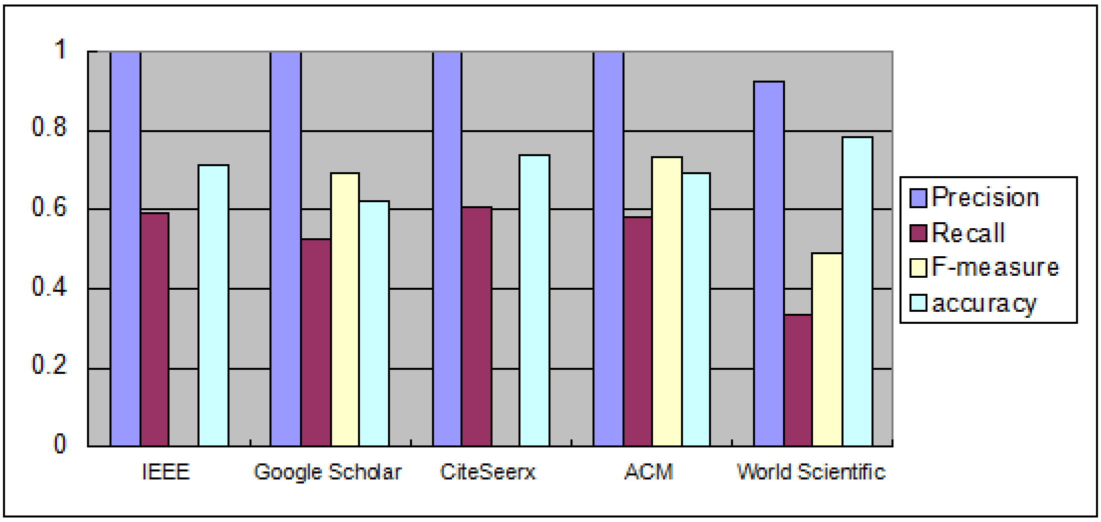


Figure 5.3. The evaluation of the RSS classification algorithm

The result from Figure 5.3 shows that the precision values for four dataset are 1. Only one has low value which is World Scientific. IEEE, ACM and CiteSeerx are the best and their results are closed in terms of recall and f-measure .While GOOGLE is the best in terms of precision, it has low value in terms of recall as compared to the other three datasets. The Precision, recall and f-measure for World Scientific dataset are not good.

The results from tables 5.1, 5.2 and 5.3 show that the dataset created from different digital libraries effect on the result. When the size of digital library is large then it means the related document for specific query is efficient. While when the size of digital library is small for example World Scientific, the document retrieved is not the same as the large one. For example to create dataset for query expansion and ontology, the document retrieved from IEEE is more efficient while from World Scientific is small number comparing with the large one. It effect on precision and recall because this document contain little number of information about query submitted form example frequency =1 or =2. And because it is small size so, this mean will be in related according to the criteria in creating dataset to related group and non-related.

In term of accuracy it is efficient because the number of document is related in correct classification and non-related in correct group is very good because the number of no related document is high. While the other digital has high precision or high recall which effect on the accuracy.

The comparison between all these algorithms (CFV_TC, SFV_TC, and RSS) is presented in tables (5.4, 5.5, and 5.6). Table (5.4) and Figure 5.4 present the comparison in terms of precision, Table 5.5 and Figure 5.5 presents comparisons in terms of recall between these algorithms and Table 5.6 and Figure 5.6 show the comparisons in terms of f-measure.

Results of evaluating these classification are depicted Table 5.4 and Figure 5.4.

Table 5.4

The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of precision

Dataset	**RSS**	**CFV_TC**	**SFV_TC**
IEEE	1	0.9174	0.8821
Google Scholar	1	0.9667	0.8962
CiteSeerx	1	0.9924	0.9477
ACM	1	0.9342	0.9158
World Scientific	0.9257	0.8505	0.8420

From table 5.4 the precision from RSS is the best value compared with the CFV_TC algorithm and SFV_TC algorithms. For SFV_TC all the results are less than the other algorithms. The precision calculates the number of documents which is classified into specific class correctly. If there are additional number of documents classified into this class and it is not related, the precision value will be degraded. The proposed study depends on the relation between the concepts present in the document from ontology therefore it utilized from all relations between the concepts and did not depend on the concepts of frequency. For example "machine learning" is related to "classification" and "clustering". If there is a document containing much information about the two classes, “classification" and "clustering", semantically will be classified to the "machine learning" class which has relation with the other two. This means this work expand the classification semantically by adding more related classes.

More, the other two algorithms CF.IDF and CFV depend on frequency of concepts in different assumption and the criteria for building a group of related and not related is the same. The result therefore will be divergence.

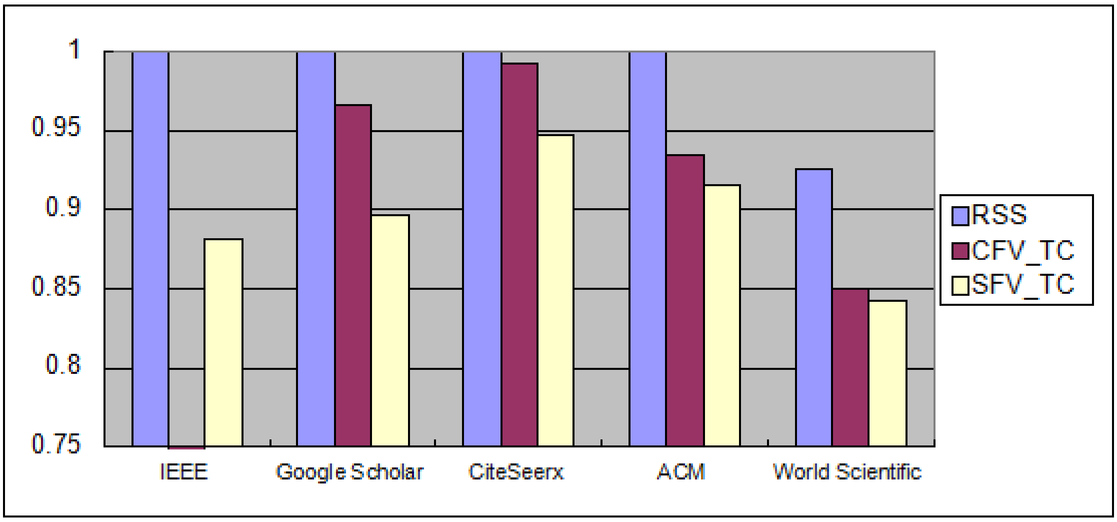


Figure 5.4. The comparison between RSS, CFV_TC, and SF_TC algorithm in terms of precision

The result from Figure 5.4 shows that the best precision in terms of the algorithm is for RSS. For the SFV_TC all the resulted values are lower than the other algorithms. Dataset obtained from CiteSeerx performs the best, while the remaining other repositories namely Google, ACM and IEEE perform considerably well. World Scientific however, scores low values for all datasets and algorithms.

Precision evaluating means findings the number of unrelated document set which are categorized into some classes. According to RSS and CFV_TC both algorithms depends on the concept frequency and used different assumption. And this is the same techniques used to make the decision in splitting the dataset into two parts related and not related. As a consequence, the result of RSS and CFV_TC are convergent because RSS used CF.IDF to calculate the weight of each concept and CFV_TC used Normalized concept frequency with respect to the section to calculate the weight of each concept from document. The SFV_TC depends on the semantic relation between the concepts with respect to the structure of the document. No term frequency used in this algorithm, so that there is a difference in result compared with the other algorithm and with the dataset. Another thing is that the not related ry 41a is not high value. In addition, the concept frequency ignores the semantic relation between the concepts represented by the document. Results of evaluating these classification are depicted Table 5.5 and Figure5.5.

Table 5.5

The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of recall

Dataset	**RSS**	**CFV_TC**	**SFV_TC**
IEEE	0.5904	0.7830	0.9128
Google Scholar	0.5292	0.7008	0.8886
CiteSeerx	0.6089	0.6144	0.7627
ACM	0.5820	0.7121	0.8670
World Scientific	0.3338	0.4242	0.5285

As shown in Table 5.5, the recall value recorded from RSS is lower than CFV_TC. Google, ACM and IEEE all have low recall values, but the recall value for CiteSeerx is high Recall refers to the number of missing documents in classification. When comparing RSS recall values with SFV-TC, it was observed that the values generated by SFV_TC are higher. Similar pattern is observed when SFV_CV is compared against CFV_TC.

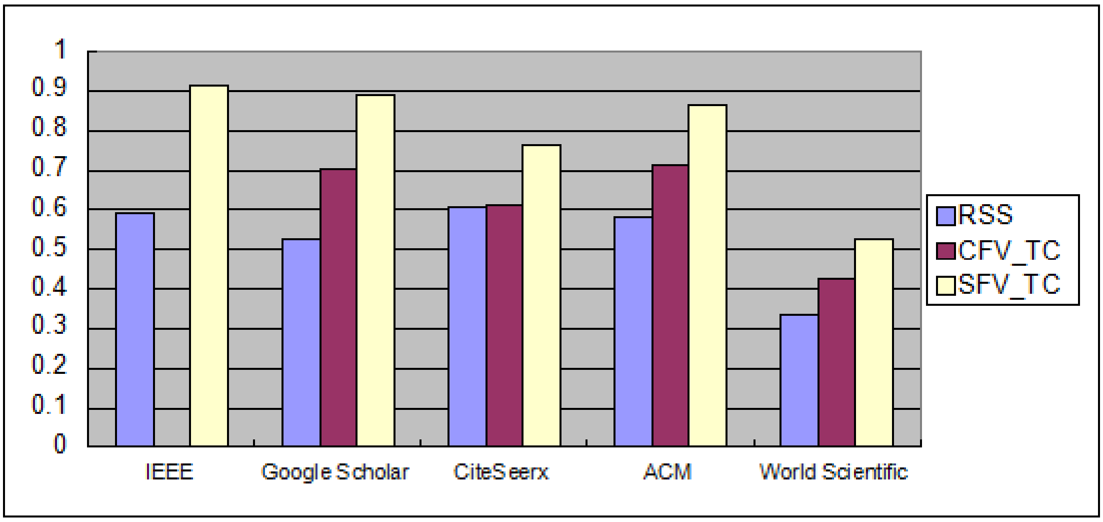


Figure 5.5. The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of recall

The result from Figure 5.5 shows that the best recall in terms of the algorithm is for SFV_TC compared to other two algorithms. The CFV_TC give the highest result compared with RSS. Only one dataset has higher results from RSS comparing with CFV_TC.

In terms of dataset, there is no dataset that give the highest result in all algorithms. Results of evaluating these classification are depicted Table 5.6 and Figure 5.6.

Table 5.6

The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of F_measure

Dataset	RSS	CFV_TC	SFV_TC
IEEE	0.7424	0.8449	0.8972
Google Scholar	0.6921	0.8125	0.8924
CiteSeerx	0.7569	0.7589	0.8452
ACM	0.7358	0.8089	0.8907
World Scientific	0.4907	0.5661	0.6494

F-measure is used to find the relationship between the precision and recall. In Table 5.6, it is shown that F-measure values from RSS is lower than CFV-TC for all datasets, except CiteSeerX. The CiteSeerx dataset has high recall values when is compared with CFV_TC. This is due to CiteSeerx possessing equal value in precision and records the lowest values for recall.

When comparing RSS f-measure values with SFV_TC, all resulted values from SFV_TC is higher than RSS algorithm. The comparison between SFV_TC with CFV_TC shows all values have highest f-measure values resulted from CFV_TC.

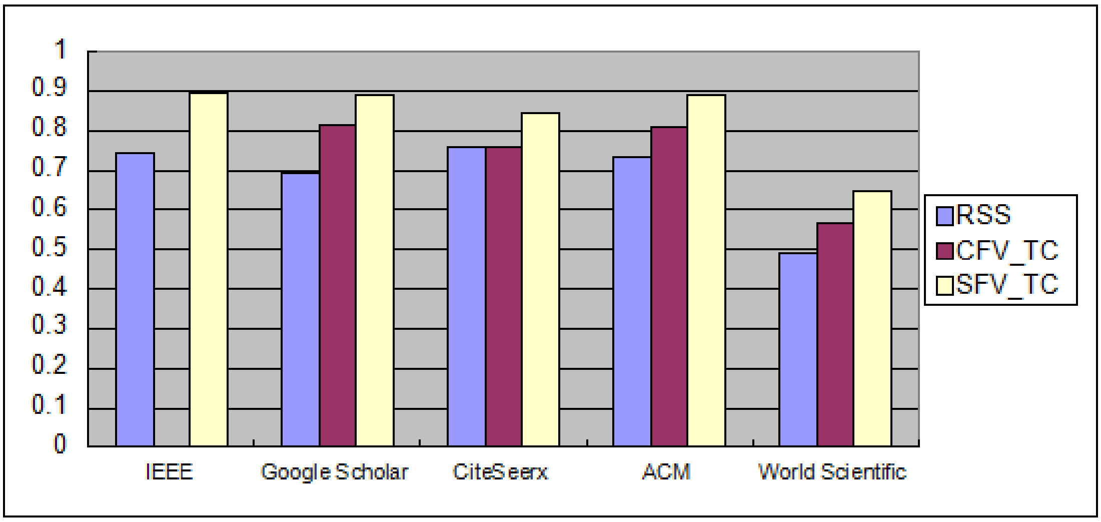


Figure 5.6. The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of F_measure

The result from Figure 5.6 shows that the best f-measure in terms of the algorithm is for SFV_TC compared to the other two algorithms. The CFV_TC gives the highest result comparing with RSS. Only one dataset has higher results from RSS comparing with CFV_TC.

In terms of dataset, no dataset obtained the highest result in all algorithms. Results of evaluating these classification are depicted Table 5.7 and Figure 5.7.

Table 5.7

The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of Accuracy

Dataset	RSS	CFV_TC	SFV_TC
IEEE	0.7114	0.7828	0.8542
Google Scholar	0.6228	0.7571	0.8228
CiteSeerx	0.7400	0.7742	0.8200
ACM	0.6942	0.7685	0.8428
World Scientific	0.7857	0.7828	0.8542

Accuracy is used to find the relationship between the number of set of document classified correctly (negative and positive) over the size of the dataset in Table 5.7, it is shown that Accuracy values from RSS is lower than CFV-TC and SFV for all datasets. Only World Scientific dataset has higher results from RSS comparing with CFV_TC.

When comparing CFV_TC accuracy values with SFV_TC, all resulted values from SFV_TC is higher than CFV_TC algorithm. If the document classified correctly the accuracy is efficient.

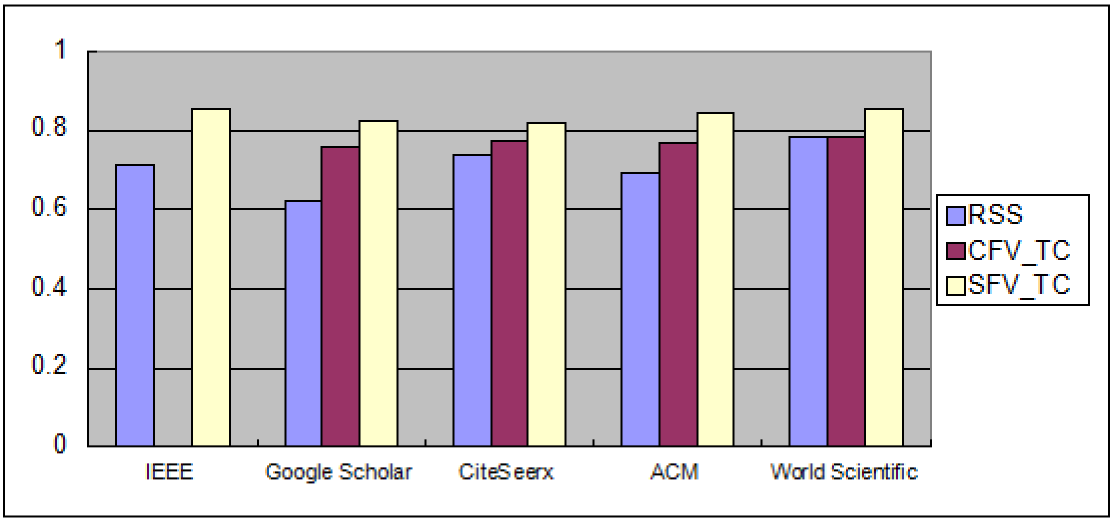


Figure 5.7. The comparison between RSS, CFV_TC, and SFV_TC algorithm in terms of Accuracy

The result from Figure 5.7 shows that the best accuracy in terms of the algorithm is for SFV_TC compared to the other two algorithms. The CFV_TC gives the highest result comparing with RSS. Only one dataset has higher results from RSS comparing with CFV_TC. In terms of dataset, IEEE and World Scientific have high accuracy values.

The second work is to compare the proposed work with another one try to reduce the dimension of training set using feature selection methods and using traditional classification algorithm. The work proposed by (Dollah & Aono, 2011) which classify the document using SVM classification algorithm. Ontology used to present the feature set instead of terms so that the dimension is reduced. Set of examples are created to train the classifier. Chi square used to reduce the dimension of the training set. Chi square is rank feature used to rank the feature values according the probability of feature distribute on different classes.

ML-SVM used to do this test on all dataset created before.

Results of evaluating these classification are depicted. Table 5.8 and Figure 5.8 present the classification evaluation in terms of precision, recall, F-measure and accuracy of the SVM Classification model in (Dollah & Aono, 2011).

Table 5.8

Evaluation of SVM

Dataset	Precision	Recall	F-measure	Accuracy
IEEE	0.8660	0.2382	0.3736	0.5730
Google Scholar	0.9356	0.3934	0.5539	0.6539
CiteSeerx	0.9418	0.3979	0.5595	0.8246
ACM	0.9372	0.4687	0.6248	0.6928
World Scientific	0.7080	0.4995	0.5857	0.8515

From table 5.8, three datasets have the top values in terms of precision. The World Scientific records a weak value. In terms of recall all datasets have very weak results and the best one is for World Scientific dataset. And the worst one is the IEEE dataset. In terms of F-measure, ACM is the best compared to the other datasets. In terms of the best values in term of accuracy is recorded from World Scientific dataset. By comparing the result from SVM classification algorithm and the RSS which used CFIDF method to classify the document, the results are different. SVM depend on examples created as training set and find the relation between the training set and classes while in RSS there is no training set just used the concept frequency to classify the document. The CFIDF used concept frequency to represent the document and it is the same method used to create the dataset and decide the case of each document related or not related while in SVM the criteria depend on CHi square use to represent the feature of the training and testing set.

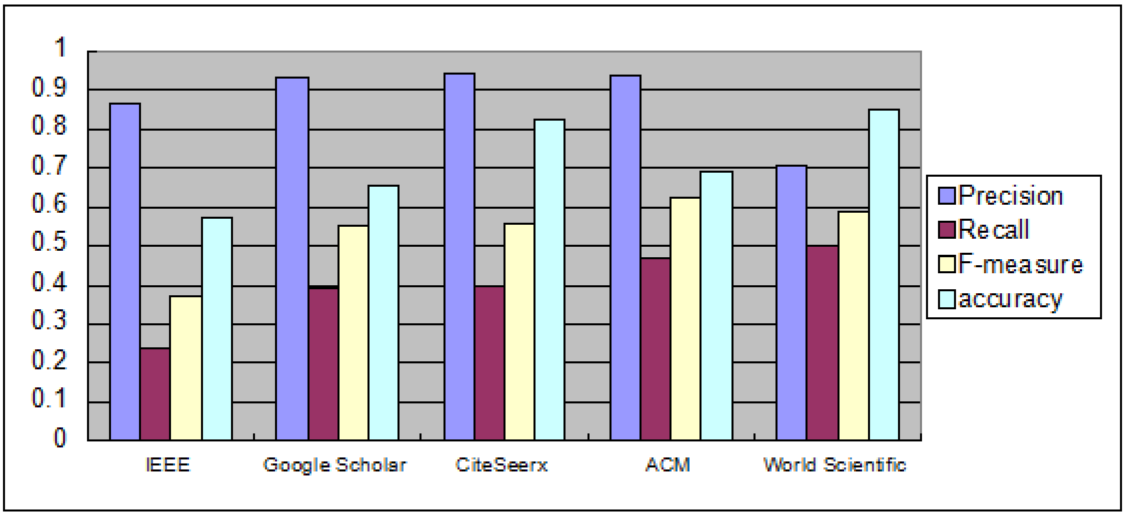


Figure 5.8. Results of the SVM classification

The result from Figure 5.8 shows that the precision values for four dataset are 1. Only one has low value which is World Scientific. IEEE, ACM and CiteSeerx are the best and their results are closed in terms of recall and f-measure .While GOOGLE is the best in terms of precision, it has low value in terms of recall as compared to the other three datasets. The Precision, recall and f-measure for World Scientific dataset are not good.

The comparison between all these algorithms (CFV_TC, SFV_TC, and SVM) is presented in tables (5.9, 5.10, 5.11 and 5.12). Table (5.9) and Figure 5.9 present the comparison in terms of precision, Table 5.10 and Figure 5.10 presents comparisons in terms of recall between these algorithms and Table 5.11 and Figure 5.11 show the comparisons in terms of f-measure and Table 5.12 and Figure 5.12 show the comparisons in terms of accuracy.

Results of evaluating these classification are depicted Table 5.9 and Figure 5.9 show the comparison between SVM, CFV_TC, and SFV_TC algorithm in term of precision.

Table 5.9

The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of precision

Dataset	**SVM**	**CFV_TC**	**SFV_TC**
IEEE	0.8660	0.9174	0.8821
Google Scholar	0.9356	0.9667	0.8962
CiteSeerx	0.9418	0.9924	0.9477
ACM	0.9372	0.9342	0.9158
World Scientific	0.7080	0.8505	0.8420

From table 5.9 the precision from the proposed algorithms CFV_TC algorithm is the best value compared with the SVM. For SFV_TC some dataset are best compared with SVM.

More, the other two algorithms CF.IDF and CFV depend on frequency of concepts in different assumption and the criteria for building a group of related and not related is the same. The result therefore will be divergence.

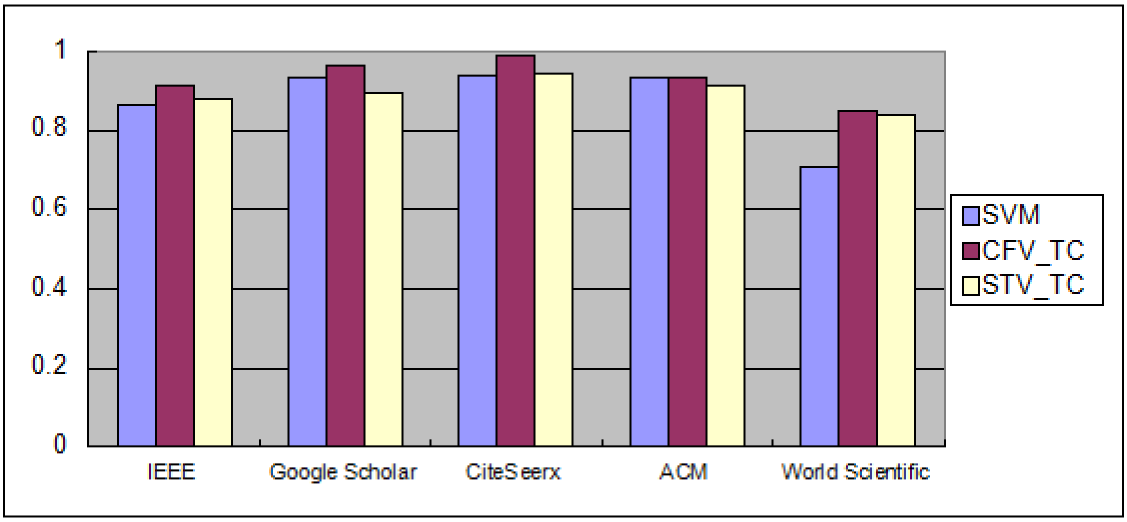


Figure 5.9. The comparison between SVM, CFV_TC, and SF_TC algorithm in terms of precision

The result from Figure 5.10 shows that the best precision in terms of the algorithm is for RSS. For the SFV_TC all the resulted values are lower than the other algorithms. Dataset obtained from CiteSeerx performs the best, while the remaining other repositories namely Google, ACM and IEEE perform considerably well. World Scientific however, scores low values for all datasets and algorithms.

Precision evaluating means findings the number of unrelated document set which are categorized into same classes. According to RSS and CFV_TC both algorithms depends on the concept frequency and used different assumption. And this is the same techniques used to make the decision in splitting the dataset into two parts related and not related. As a consequence, the result of RSS and CFV_TC are convergent.

The SFV_TC depends on the semantic relation between the concepts with respect to the structure of the document. No term frequently used in this algorithm, so that there is a difference in result compared with the other algorithm and with the dataset. Another thing is that the not related document is not high value. In addition, the concept frequency ignores the semantic relation between the concepts represented by the document.

Results of evaluating these classification are depicted Table 10.5 and Figure 5.10.

Table 5.10

The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of recall

Dataset	**SVM**	**CFV_TC**	**SFV_TC**
IEEE	0.2382	0.7830	0.9128
Google Scholar	0.3934	0.7008	0.8886
CiteSeerx	0.3979	0.6144	0.7627
ACM	0.4687	0.7121	0.8670
World Scientific	0.4995	0.4242	0.5285

As shown in Table 5.10, the recall value recorded from SVM is lower than CFV_TC. Google, CiteSeerx and IEEE all have low recall values, but the recall value for CiteSeerx is high.Recall refers to the number of missing documents in classification. When comparing SVM recall values with SFV-TC, it was observed that the values generated by SFV_TC are higher. Similar pattern is observed when SFV_CV is compared against CFV_TC.

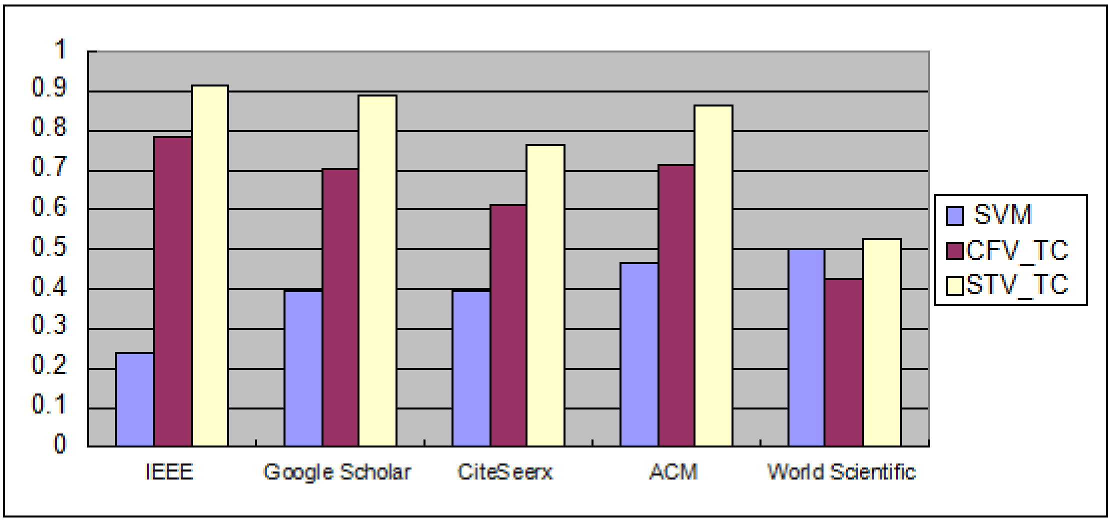


Figure 5.10. The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of recall

The result from Figure 5.10 shows that the best recall in terms of the algorithm is for SFV_TC compared to other two algorithms. The CFV_TC give the highest result compared with RSS. Only one dataset has higher results from RSS comparing with CFV_TC. Results of evaluating these classification are depicted Table 5.11 and Figure 5.11.

Table 5.11

The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of F_measure

Dataset	**SVM**	**CFV_TC**	**SFV_TC**
IEEE	0.3736	0.8449	0.8972
Google Scholar	0.5539	0.8125	0.8924
CiteSeerx	0.5590	0.7589	0.8452
ACM	0.6248	0.8082	0.8907
World Scientific	0.5857	0.5661	0.6494

In Table 5.11, it is shown that F-measure values from SVM is lower than CFV-TC for all datasets, except World Scientific.

When comparing SVM f-measure values with SFV_TC, all resulted values from SFV_TC is higher than SVM algorithm.

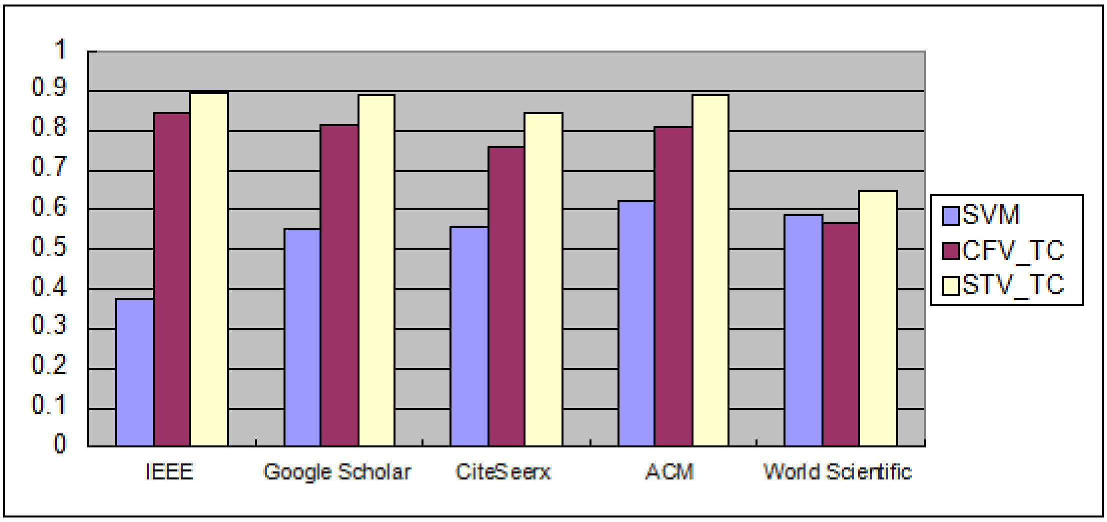


Figure 5.11. The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of F_measure

The result from Figure 5.11 shows that the best f-measure in terms of the algorithm is for SFV_TC compared to the other two algorithms. The CFV_TC gives the highest result comparing with SVM. Only one dataset has higher results from RSS comparing with CFV_TC. Results of evaluating these classifications are depicted Table 5.12 and Figure 5.12.

Table 5.12

The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of Accuracy

Dataset	**SVM**	**CFV_TC**	**SFV_TC**
IEEE	0.5730	0.7828	0.8542
Google Scholar	0.6539	0.7571	0.8228
CiteSeerx	0.8246	0.7742	0.8200
ACM	0.6928	0.7685	0.8428
World Scientific	0.8158	0.7828	0.8542

In Table 5.12, it is shown that Accuracy values from SVM is lower than CFV-TC and SFV for all datasets. Only World Scientific dataset has higher results from SVM comparing with CFV_TC.

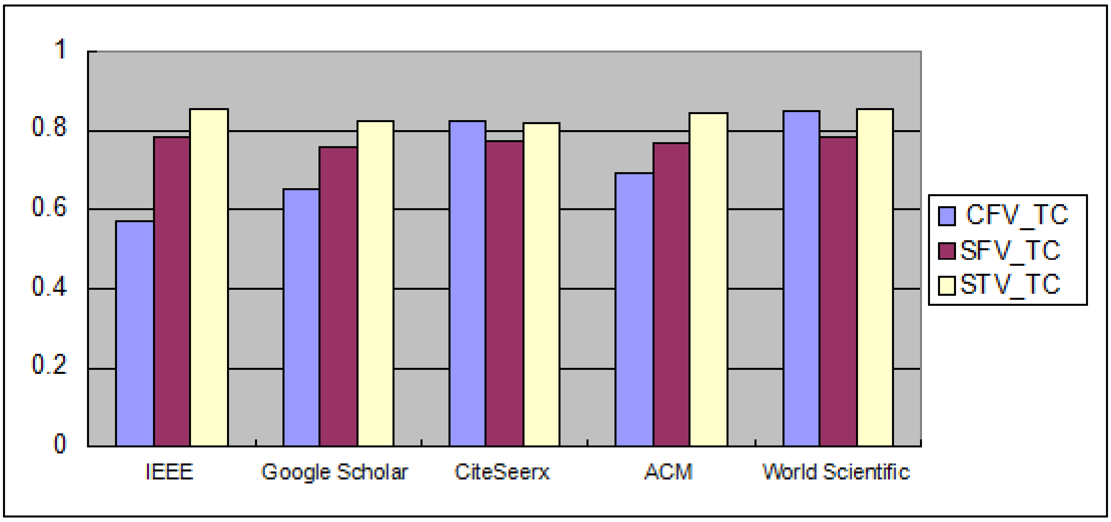


Figure 5.12. The comparison between SVM, CFV_TC, and SFV_TC algorithm in terms of Accuracy

The result from Figure 5.13 shows that the best accuracy in terms of the algorithm is for SFV_TC compared to the other two algorithms. The CFV_TC gives the highest result comparing with SVM. Only one dataset has higher results from RSS comparing with CFV_TC.

From these experiment, the result show that the dataset effects on the result from all these different work. To create dataset, different queries are submitted to library to select set of document. Select the top 50 document from the resulted document. To make a decision if the document is related to the query or not, tf.idf was used to calculate the weight of each query term. Tf.idf depend on the size of the corpus by calculating the term frequency then find the number of document contain this term. If the library is large size the result is more related to the query submitted. While if the library is small size the number of related document is little. For more detail if the frequency of terms from query is little it will be selected in dataset.

Another part is to solve the semantic problem in text classification. Many works try to use concept frequency inverse document frequency CF.IDF techniques. This work depends on frequency of each concept from document in training set and testing set. This technique ignore many important property of document. For structured document CF.IDF ignore the distribution of terms in document. From these experiments the results shows that the text classification depending on document structure give a good result comparing with concept frequency. In term of precision, the results shows that concept frequency is more accurate. In fact, the proposed work depends on the semantic relation between concepts from ontology and document. So that it many document will be classified even it has low frequency in document. This is because depends on number of related concepts not on the frequency. This can be used to expand the search semantically. In term of recall the result shows that the proposed works enhanced the concept frequency in text classification depending on the structure of document. Document and recall and accuracy. In term of precision it is clear that there are many concept related semantically will be as class.

The result from these experiments show that the similarity measure can give comparable result in text classification task. In traditional text classification algorithm, for example SVM classification algorithm, the algorithm depends on the training set created to train the classifier which mean the results depends on the training set. In similarity measure proposed in this works, the similarity between the set of concept have a semantic relation with feature set from the document to be classified.

CHi square selection method combined with ontology reduced the dimension of the training set in the experiment. This method try to replace the terms from feature set with set of concept and then select the most important feature from training set. In fact this method rank the feature according to distribution on different classes. Therefore, it depends on the training set created for this classifier for specific corpus. Even using ontology by combining it with traditional classification algorithm, in this experiment SVM, still depends on the training set which effect on the results.

Table 5.13 shows the comparison evaluation between DT_TREE, CFV_TC, and SFV_TC algorithm in term of feature size.

Table 5.13

The Comparison between DT_TREE, CFV_TC, and SFV_TC Algorithm in terms of Feature Size

Dataset	**DT_TREE**	**CFV-TC**	**SFV-TC**
IEEE	112	6	18
GOOGLE	107	6	18
CiteSeerx	101	6	18
ACM	112	6	18
World Scientific	112	6	18

The feature size of the training set is calculated in this part of this chapter (Rizvi & Wang, 2010) try to reduce the size of the training set to cluster the document. The main idea is to calculate the score between two documents by finding a measure between them. The problem of this similarity is the high dimensionality of data. The dataset was scientific paper and it depends on the structure of the document which is a collection of sections. The author tries to segment the document to reduce the size of the training set to make a clustering. This work used term to represent the vector of features to make a calculation. The result of this work reaches until 100 least feature set. And by comparing with the proposed work on thesis the size of training set it at most 10 for CFV_TC and 30 for SFV_TC for all datasets. Because we already depend on the concepts of ontology to represent the training set, there is no redundant or irrelevant data on these set.

5.2 Summary

In this chapter, a tradeoff between the precision and recall of text classification by utilizing the properties of ontologies which include the concepts and semantic relations between these concepts. Also, this study investigates the effect of the document to get more precise results. The result found that the recall and accuracy are improved by using the structure of document combined with semantic relations between concepts from the ontology. While the results in terms of precision are low compared with frequency, but the difference is not too much. For the number of number document classified wrongly, in fact it is related semantically. And we can say this work can help in exploring the related document implicitly to expand the text classification. The proposed work making a tradeoff between the precision and recall which is calculated using f-measure. In terms of feature size, it is reduced compared with the previous work.

CHAPTER SIX
CONCLUSION AND FUTURE WORK

This chapter is dedicated to summarize the thesis achievements as well as to outline future guidelines in the on ontology for text classification research field. A summary of the thesis contributions is presented in Section 6.1. Section 6.2 offers some suggestions and future directions.

6.1 Contributions

The massive growth of digital libraries imposes management and organization on these libraries to ease the retrieving and browsing operations. Thus, text classification methods are used to classify text in these libraries. The main contributions of this thesis as specified in Chapter 1 are as follows:

The first contribution of this thesis is related to the enhanced algorithm which extract set of features represent the document for classification task. This enhancement is achieved by focusing on the structure of the document to be classified. The first contribution in this research is to overcome the semantic problem in text classification method. The second contribution revolves around the dimensionality reduction which effect on the classification performance. The main cause of high dimension of data is the training set for classification task. Therefore, another algorithm is proposed to classify the document to set of classes by using only the set of concepts from ontology. The similarity between the feature vector from document and set of concept from ontology to classify the document into set of concepts is calculated by using Similarity measure.

This method shows the performance is enhanced in term of accuracy, f-measure comparing with other method using training set for classifying text using traditional classification algorithm enhance by using ontology because this algorithm depends creating training set. Furthermore, two algorithms proposed in this research. The first algorithm depends on the assumption that a concepts appeared on different sections from structured document is most important than the other that appears on only one section. While the second algorithm depends on the assumption that the concept from different sections with its related set of concept is more important than the concepts on one section even it has high frequency.

In section 4.4, Ontology Based Text Classification Algorithm (OBTC) is presented in detail on page 146. For the last part which deal with classification process, the first algorithm Concept Feature Vector_text Classification (CFV_TC) was presented in section (4.5.1) on page 151. While the second algorithm Structure Feature Vector_ text Classification (SFV_TC) was presented in section (4.5.2) on page 154.

6.2 Future Works

This section concentrates on the future research recommendations based on this research. These recommendations can be outlined below:

i. In this research, the ontology concepts used to enhance text the classification approach semantically. The main algorithms were evaluated in term of precision, recall and accuracy. The recall values needs to be enhanced by applying other types of classification approaches i.e. SVM, K-NN to make sure that the proposed approaches are effective in terms of recall.

ii. Other methods can be used to enhance the text classification approaches through enrich the ontology. More concepts could be added to the seed ontology created manuall by using the proposed extraction algorithms.

iii. The work in this thesis can be enhanced by using application which can decompose the scientific document to different number of sections depending on the content of the document.

iv. The proposed work could also be extended by adding more classes semantically. Ontology is a set of concepts, each concept is connected to other concept if there is any semantic relation between them. So that, by using these semantic relation the classification can be expanded to more related concept from ontology.

References

Achananuparp, P., Zhou, X., Hu, X., & Zhang, X. (2008). Semantic representation in text classification using topic signature mapping. *Paper presented in IEEE International Joint Conference on Neural Networks IJCNN*, 1034 – 1040.

Agarwal, S., Singhal, A., & Bedi, P. (2012). Classification of RSS feed news items using ontology. *Paper presented in IEEE 14th Intelligent Systems Design and Applications (ISDA)*, 491 – 496.

Aggarwal, C., C., Zhai, & ChengXiang. (2012). Mining Text Data. *Chapter six of the book XII, 524 p*, 182.

Ahmed, N., Khan, S., Latif, K., Masood, A., & Elberrichi, Z. (2008). Extracting semantic annotations and their correlation with document components. *Paper presented in IEEE 4th International Conference on Merging Technologies, ICET* 32 – 37.

Ajgalik, M., Barla, M., & Bielikova, M. (2013). From Ambiguous Words to Key-Concept Extraction. *Paper presented in IEEE 24th International Workshop on Database and Expert Systems Applications (DEXA)*, 63 - 67.

Almeida, T. A., Yamakami, A., & Almeida, J. (2009). Evaluation of Approaches for Dimensionality Reduction Applied with Naive Bayes Anti-Spam Filters. *Paper presented in IEEE International Conference on Machine Learning and Applications, ICMLA*, 517-522.

Aytug, H., Boylu, F., & Koehler, G. J. (2006). Learning in the Presence of Self-Interested Agents. *Paper presented in IEEE International Conference of the 39th Annual Hawaii, Vol. 7*, 1-7.

Basu, T., & Murthy, C. A. (2012). Effective Text Classification by a Supervised Feature Selection Approach. *Paper presented in IEEE 12th International Conference on Data Mining Workshops (ICDMW), 918 –* 925.

Bhatia, N., & Vandana, S. (2010). Survey of Nearest Neighbor Techniques. *(IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 2,* 302-305.

Bin, L., Jun, L., Min, Y., J., & Ming, Z. Q. (2008). Automated Essay Scoring Using the KNN Algorithm. *Paper presented in IEEE International Conference on Computer Science and Software Engineering, Vol. 1,* 735 - 738.

Bleik, S., Mishra, M., Huan, J., & Song, M. (2013). Text Categorization of Biomedical Data Sets Using Graph Kernels and a Controlled Vocabulary. *Paper presented in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 10 (Issue 5),* 1211 - 1217.

Brank, J., Grobelnik, M., Milic-Frayling, N., & Mladenic, D. (2002). Interaction of feature selection methods and linear classification models. *In Workshop on Text Learning held at ICML*.

Brank, J., Mladenić, D., & Grobelnik, M. (2010). Large-scale Hierarchical Text Classification Using SVM and Coding Matrices. *In: Large-Scale Hierarchical Classification Workshop of ECIR, 28 – 31 March, Milton Keynes, UK.*

Calvier, F. c.-E., Planti´e, M., Dray, G. e., & Ranwez, S. (2013). Ontology Based Machine Learning for Semantic Multiclass Classification. *Author manuscript,* *published in "TOTH: Terminologie & Ontologie: Théories et Applications 2013, Chambéry: France*.

Calvo, R. A., Lee, J.-M., & Li, X. (2006). Managing content with automatic document classification. *Journal of Digital Information*, 1-15.

Celik, K., & Gungor, T. (2013). A comprehensive analysis of using semantic information in text categorization. *Paper presented in IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 1 – 5.

Chang, Y.-H. (2007). Automatically Constructing a Domain Ontology for Document Classification. *Paper presented in IEEE International Conference on Machine Learning and Cybernetics, Vol. 4*, 1942 - 1947.

Chang, Y.-H., & Huang, H.-Y. (2008). An Automatic Document Classifier System based on Naíve Bayes Classifier and Ontology. *paper presented in IEEE International Conference on Machine Learning and Cybernetics, Vol. 6*, 3144 - 3149.

Che, C., & Teng, H. (2009). Document representation combining concepts and words in Chinese text categorization. *Paper presented in IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 1-5.

Chirawichitchai, N., Sanguansat, P., & Meesad, P. (2009). A Comparative Study on Feature Weight in Thai Document Categorization Framework, 257-266.

Cunhua, L., Yun, H., & Zhaoman, Z. (2010). An event ontology construction approach to web crime mining. *Paper presented in IEEE Seventh*

International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol. 5, 2441 - 2445.

Cunningham, P. & Delany, S. J. (2007). K-Nearest Neighbour Classifiers. *Technical Report UCD-CSI-2007-4.*

Debole, F., & Sebastiani, F. (2003). Supervised Term Weighting for Automated Text Categorization. *Proceedings of the 2003 ACM symposium on Applied computing,* 784-788.

Deisy, C., Gowr, M., Baskar, S., Kalaiarasi, S. M. A., & Ramraj, N. (2010). A Novel Term Weighting Scheme Midf for Text Categorization. *Journal of Engineering Science and Technology, Vol. 5,* 94 - 107.

Devare, M., Rikert, J., C., Caruso, B., Lowe, B., Chiang, K., & McCue, J. (2007). Connecting People, Creating a Virtual Life Sciences Community. *D-Lib Magazine, ISSN 1082- 9873, Vol. 13, No. 7/8.*

Dhillon, I. S., Mallela, S., & Kumar, R. (2003). A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning 1265-1287.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *In First International Workshop on Multiple Classifier Systems 2000, Cagliari, Italy, Vol. 1857 of Lecture Notes in Computer Science, Springer,*1–15.

Dollah, R. B., & Aono, M. (2011). Ontology based Approach for Classifying Biomedical Text Abstracts. *International Journal of Data Engineering (IJDE), Vol. 2 (Issue 1),* 1-15.

Elberrichi, Z., Amel, B., & Malika, T. (2012). Medical Documents Classification Based on the Domain Ontology MeSH. *The International Arab Journal of e-Technology, Vol. 2, No. 4,* 210-215.

Erenel, Z., Altincay, H. & Varoglu, E. (2011). Explicit Use of Term Occurrence Probabilities for Term Weighting in Text Categorization. *Journal of information science and engineering 27,* 819-834.

Fang, J., Guo, L., Wang, X., & Yang, N. (2007). Ontology-Based Automatic Classification and Ranking for Web Documents. *Paper presented in IEEE Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD, Vol. 3,* 627 - 631.

Fang, J., Guo, L., & Niu, Y. (2010). Documents Classification by Using Ontology Reasoning and Similarity Measure. *Paper presented in IEEE Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol. 4,* 1535 - 1539.

FAO Ontology Portal Prototype Fishery (2004). Development of Multilingual Domain Ontologies Fishery Ontology.

Fu, Z., Chen, C., Gong, Y., & Bie, R. (2008). A Comparison Study: Web Pages Categorization with Bayesian Classifiers. *Paper presented in 10th IEEE International Conference on High Performance Computing and Communications, HPCC '08,* 789 - 794.

Gang, X., & Jiancang, X. (2009). Performance Analysis of Chinese Webpage Categorizing Algorithm Based on Support Vector Machines (SVM). *Paper* *presented in IEEE Fifth International Conference on Information Assurance and Security, IAS '09, Vol. 1*, 231 - 235.

Gardner, D., Akil, H., Ascoli, G., A., Bowden, D., A., Bug, W., Duncan, E., Donohue, David, H., Goldberg, Grafstein, B., Grethe, J., S., Gupta, A., Halavi, M., Kennedy, D., N., Marenco, L., Martone, M., E., Miller, P., L., Müller, H., L., Robert, A., Shepherd, G., M., Sternberg, P., W., Essen, D., C., V., & Williams, R., W. (2008). The Neuroscience Information Framework: A Data and Knowledge Environment for Neuroscience. *Published in final edited form as: Neuroinformatics, Vol. 6 (3)*, 149–160.

Garrido, A. L., Gomez, O., Ilarri, S., & Mena, E. (2011). NASS: News Annotation Semantic System. *Paper presented in 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 904 - 905.

Genkin, A., Madigan, D., & Lewis, D. D. (2007). Large-Scale Bayesian Logistic Regression for Text Categorization. *American Statistical Association and the American Society for Quality Vol. 49, No. 3*, 291-304.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *ACM Digital Library Knowledge Acquisition - Special issue: Current issues in Knowledge Modeling Vol. 5*, 199 – 220.

Guhan, T., & Selvarajan, S. (2014). A Survey on the Suitability of ANN Based Classification Algorithms for Multidimensional Data Classification. *International Journal of Computer Science information and Engineering Technologies*, 1-5.

Ha-Thuc, V., & Renders, J.-M. (2011). Large-Scale Hierarchical Text Classification without Labelled Data. Proceedings *of the fourth ACM international conference on Web search and data mining, WSDM '11,* 685-694.

Haifeng, L., Shousheng, L., & Zhan, S. (2010). An improved KNN text categorization on skew sort condition. *Paper presented in IEEE International Conference on Computer Application and System Modeling (ICCASM) Vol. 7,* 182-186.

Halloran, J. (2009). Classification: Naive Bayes vs Logistic Regression. *University of Hawaii at Manoa EE 645, Fall. 2009,* 1-24.

Han, J., Kamber, M., & Pei, J. (2013). Data Mining: Concepts and Techniques. *Book, Chapter 9, Classification: Advanced Methods, University of Illinois at Urbana-Champaign & Simon Fraser University.*

Harrag, F., El-Qawasmah, E., & Al-Salman, A. M. S. (2010). Comparing Dimension Reduction Techniques for Arabic Text Classification Using BPNN Algorithm. *Paper presented in IEEE First International Conference on Integrated Intelligent Computing (ICIIC),* 6-11.

Harrag, F., El-Qawasmeh, E., & Pichappan, P. (2009). Improving Arabic text categorization using decision trees. *Paper presented in IEEE First International Conference on Networked Digital Technologies, NDT '09*, 110-115.

Haruechaiyasak, C., Jitkrittum, W., Sangkeettrakarn, C., & Damrongrat, C. (2008). Implementing News Article Category Browsing Based on Text

Categorization Technique. *Paper presented in IEEE International Conference on Web Intelligence and Intelligent Agent Technology WI-IAT '08*, 143 - 146.

He, D., & Wu, X. (2006). Ontology-Based Feature Weighting for Biomedical Literature Classification. *Paper presented in IEEE International Conference on Information Reuse and Integration,* 280-285.

He, J., Tan, A. H., & Tan, C. L. (2000). A comparative study on Chinese text categorization methods. *In PRICA 2000 Workshop on Text and Web Mining, Melbourne, Australia*, 24–35.

Hong-wei, Z., Jian-fang, C., & Feng, S.-q. (2010). An improved text feature selection method based on key words. *paper presented in IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Vol. 2,* 293 - 297.

Hong, K. (2014). Improving the Estimation of Word Importance for News Multi-Document Summarization. *Technical Reports (CIS). Paper 989*.

Howard, B., W., Soonho, K., & Hagan, D. (2005). A Crop-Pest Ontology for Extension Publication. *5th Conference of the European Federation for Information Technology in Agriculture.*

Hui, D., & Siqing, Y. (2010). An improved feature weighting algorithm for Chinese text classification. *paper presented in IEEE International Conference on Computer Application and System Modeling (ICCASM), Vol. 6*, 433-436.

Hur, A., B. & Weston, J. (2010). A User's Guide to Support Vector Machines. *Department of Computer Science, Colorado State University, DOI: 10.1007/978-1-60327-241-4_13 Source: PubMed.*

Huth, J., Brogan M., Dancik B., Kommedahl T., Nadziejka D., Robinson P., & Swanson W. (1994). *Scientific format and style: The CBE manual for authors, editors, and publishers. Cambridge: Cambridge University Press*. p. 825.

Islam, M. R., & Islam, M. R. (2008). An effective term weighting method using random walk model for text classification. *Paper presented in IEEE 11th International Conference on Computer and Information Technology, ICCIT,* 411 - 414.

Jiang, H., Li, P., Hu, X., & Wang, S. (2009). An improved method of term weighting for text classification *paper presented in IEEE International Conference on Intelligent Computing and Intelligent Systems ICIS* 294 - 298

Jin, Y., Xiong, W., & Wang, C. (2010). Feature selection for Chinese Text Categorization based on improved particle swarm optimization. *Paper presented in IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE),* 1 – 6.

Joachims, T. (1998). Text categorization with support vector machines. *In European Conference on Machine Learning (ECML).*

Kadhim, M. H., & Omar, N. (2012). Automatic Arabic Text Categorization using Bayesian learning. *Paper presented in 7th IEEE International Conference on Computing and Convergence Technology (ICCCT),* 415 - 419.

Kaur, H., & Jyoti, K. (2013). Design and Implementation of Hybrid Algorithm for e-news Classification. *International journal of computers and technology, Vol. 12, No. 1,* 3178-3186.

Kehagias, A., Petridis, V., Kaburlasos, V. G., & Fragkou, P. (2001). A Comparison of Word- and Sense-based Text Categorization Using Several Classification Algorithms *Journal of Intelligent Information Systems Vol. 21*, 227-247.

Khan, A., Baharudin, B., & Khan, K. (2010). Semantic based features selection and weighting method for text classification. *Paper presented in IEEE International Symposium in Information Technology (ITSim), Vol. 2,* 850 – 855.

Khan, A., Baharudin, B., & Khan, K. (2012). Efficient Feature Selection and Domain Relevance Term Weighting Method for Document Classification. *paper presented in IEEE Second International Conference on Computer Engineering and Applications (ICCEA), Vol. 2*, 398 - 403.

Kim, H. J., & Chang, J. (2007). Integrating Incremental Feature Weighting into Naive Bayes Text Classifier. *Paper presented in IEEE International Conference on Machine Learning and Cybernetics, Vol. 2,* 1137 – 1143.

Kruse, R., Rosner, D., & Nakhaeizadeh, G. (2001). Enhancing Text Classification to Improve Information Filtering. *Dissertation zur Erlangung des akademis chen Grades, Promotions colloquium: Magdeburg, den 07. December 2001.*

Korada, N., K., Kumar, N., S., P., Deekshitulu, Y., V., N., H. (2012). Implementation of Naive Bayesian Classifier and Ada-Boost Algorithm Using Maize Expert

System. *International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.3*, 63-75.

Kumar, R. (2011). Research methodology: A step-by-step guide for beginners (3rd). *Thousand Oaks, CA: Sage Publications Inc.*

Lee, D.-l., Yang, S.-Y., & Hsu, C.-L. (2008). Ontology-supported webpage classifier for scholar's webpages in ubiquitous information environment. *Paper presented in First IEEE International Conference on Ubi-Media Computing*, 523 - 528.

Li, J. (2013). An approach to Meta feature selection. *Paper presented in IEEE 26th Annual Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1 – 4.

Li, X., & Liu, B. (2003). Learning to Classify Texts Using Positive and Unlabeled Data *In: Proceedings of the 19th international joint conference on artificial intelligence*.

Li, Y., & Chen, C. (2012). Research on the feature selection techniques used in text classification. *Paper presented in IEEE 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 725 – 729.

Li, Y., & Hu, D. (2009). Study on the Classification of Mixed Text Based on Conceptual Vector Space Model and Bayes. *Paper presented in IEEE International Conference on Asian Language Processing, IALP '09*, 269 - 272.

Liu, J. N. K., He, Y.-L., Lim, E. H. Y., & Wang, X.-Z. (2013). A New Method for Knowledge and Information Management Domain Ontology Graph Model.

Paper presented in IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 43 (Issue 1), 115-127.

Liu, Z., & Yang, J. (2011). A Feature Selection Simultaneously Based on Intra-category and Extra-Category for Text Categorization. *Paper presented in IEEE International Conference of Intelligent Human-Machine Systems and Cybernetics (IHMSC), Vol. 2*, 178-181.

Lord, L., (2010). Components of an Ontology. An Ontology Tutorial, Computing Science at Newcastle University.

Lu, Z., Shi, H., Zhang, Q., & Yuan, C. (2009). Automatic Chinese text categorization system based on mutual information. *Paper presented in IEEE International Conference on Mechatronics and Automation, ICMA,* 4986 - 4990

Luo, X., Ohyama, W., Wakabayashi, T., & Kimura, F. (2011). A Study on Automatic Chinese Text Classification. *Paper presented in IEEE International Conference on Document Analysis and Recognition (ICDAR)*, 920 - 924.

Luong, H. P., Gauch, S., & Wang, Q. (2009). Ontology-Based Focused Crawling. Paper *presented in IEEE International Conference on Information, Process, and Knowledge Management, eKNOW '09*, 123 - 128.

Ma, L., Ofoghi, B., Watters, P., & Brown, S. (2009). Detecting Phishing Emails Using Hybrid Features. *Paper presented in IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, UIC-ATC '09.* , 493 – 497.

Malone, J. & Parinson, H. (2010). Refrence and application Ontologies. *European Bioinformatics Institute, Cambridge, CB10 1SD, UK.*

Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., & Parkinson, H. (2010). Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics.15; 26 (8) 10.1093*, 1112-1118.

Manning, C., D., Raghavan, P. & Schütze, H. (2008). Introduction to information retrieval. *Book ISBN: 0521865719. Cambridge University Press.*

Mathy, F. (2010). Assessing Conceptual Complexity and Compressibility Using Information Gain and Mutual Information. *Tutorials in Quantitative Methods for Psychology*, Vol. 6 (1), 16-30.

Maleki, M. (2010). Utilizing Category Relevancy Factor for text categorization. *Paper presented in IEEE 2nd International Conference on Software Engineering and Data Mining (SEDM)*, 334 - 339.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. *ISBN: 9780521865715*.

Manuja, M., & Garg, D. (2014). Intelligent text classification system based on self-administered ontology. *Turkish Journal of Electrical Engineering & Computer Sciences*.

Meena, M. J., & Chandran, K. R. (2009). Classifying Text with Statistically Selected Features to Closely Related Classes. *Paper presented in IEEE International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom '09*, 297- 301.

Mesleh, A. M., & Kanaan, G. (2008). Support vector machine text classification system: Using Ant Colony Optimization based feature subset selection. *Paper* *presented in IEEE International Conference on Computer Engineering & Systems, ICCES*, 143 - 148.

Mohaqeqi, M., Soltanpoor, R., & Shakery, A. (2009). Improving the Classification of Unknown Documents by Concept Graph. *Paper presented in IEEE 14th International CSI Computer Conference (CSICC'09), 259 - 264.*

Mohsenzadeh, M. , Mohaqeqi, M. , Soltanpoor, R. (2010). A New Approach for Better Document Retrieval and Classification Performance Using Supervised WSD and Concept Graph. *Paper presented in IEEE First International Conference on Integrated Intelligent Computing (ICIIC)*, 32 - 38.

Moschitti, A., & Basili, R. (2004). Complex Linguistic Features for Text Classification. *A comprehensive study Lecture Notes in Computer Science, Vol. 2997*, 181-196.

Mouratis, T., & Kotsiantis, S. (2009). Increasing the Accuracy of Discriminative of Multinomial Bayesian Classifier in Text Classification. *Paper presented in Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT '09*, 1246 – 1251.

Mousavi, H., Gao, S., & Zaniolo, C. (2013). Discovering attribute and entity synonyms for knowledge integration and semantic web search. *CSD/UCLA. Los Angeles, computer science Department technical report*, 1-12.

Munteanu, D. (2007). A Quick Survey of Text Categorization Algorithms. *The Annals of "Dunarea de Jos" University of Galati Fascicle ISSN 1221-454X*, 35-42.

Negoita, Marcia, G. H. (2004). Knowledge-Based Intelligent Information and Engineering Systems. *8th International Conference, KES 2004, Wellington, New Zealand,* September 20–25, 2004.

Nguyen, G. S., Gao, X., & Andreae, P. (2011). Phoneme Based Representation for Vietnamese Web Page Classification. *Paper presented in IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 1,* 15 – 22.

Nguyen, T. T., Chang, K., & Hui, S. C. (2011). Supervised term weighting for sentiment analysis. *Paper presented in IEEE International Conference on Intelligence and Security Informatics (ISI),* 89 – 94.

Noh, S., Seo, H., Choi, J., Choi, K., & Jung, G. (2003). Classifying Web pages using adaptive ontology. *Paper presented in IEEE International Conference on Systems, Man and Cybernetics Vol. 3*, 2144 - 2149

Nuipian, V., Meesad, P., & Boonrawd, P. (2011). A comparison between keywords and key-phrases in text categorization using feature section technique. *Paper presented in IEEE 9th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering),* 156 – 160.

Pang, X.-L., Feng, Y.-Q., & Jiang, W. (2007). An Improved Document Classification Approach with Maximum Entropy and Entropy Feature Selection. *Paper presented in IEEE International Conference on Machine Learning and Cybernetics, Vol. 7,* 3911 - 3915.

Patra, A., & Singh, D. (2013). A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification

Algorithms. *International Journal of Computer Applications, Vol. 75, No.7,* 14-18.

Pawar, P. Y., & Gawande, S. H. (2012). A Comparative Study on Different Types of Approaches to Text Categorization. *Paper presented in IEEE International Journal of Machine Learning and Computing, Vol. 2, No. 4*, 295-301.

Pei, Z., Zhou, Y., Liu, L., Wang, L., Lu, Y., & Kong, Y. (2010). A mutual information and information entropy pair based feature selection method in text classification. *Paper presented in IEEE International Conference on Computer Application and System Modeling (ICCASM), Vol. 6*, 258-261.

Ping, Y., Zhou, Y. j., Yang, Y. X., & Peng, W. p. (2010). A novel term weighting scheme with distributional coefficient for text classification with support vector machine. *Paper presented in IEEE on Youth Conference International Information Computing and Telecommunications (YC-ICT),* 182-185.

Polpinij, J. (2009). An ontology-based text processing approach for simplifying ambiguity of requirement specifications. *Paper presented in IEEE Asia-Pacific Services Computing Conference, APSCC,* 219 - 226.

Poyraz, M., Ganiz, M.C.,Akyokus, S. ; & Gorener, B. (2012). Exploiting Turkish Wikipedia as a semantic resource for text classification. *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA),* 1-5.

Pote, R., M., & Akarti, S. P. (2014). Study of multiclass classification for imbalanced biomedical data. *International Journal of Application or Innovation in Engineering & Management (IJAIEM). Vol. 3, Issue 9*, 158-162.

Qin, T., Liu, T.-Y., Zhang, X.-D., Wang, D.-S., & Li, H. (2008). Global Ranking Using Continuous Conditional Random Fields. *Institution Microsoft Research Tech Report Number MSR-TR-156,* 1-8.

Rafi, M., Hassan, S., & Shaikh, M. S. (2012). Content-based Text Categorization using Wikitology. 1-9.

Raghunathan, P. (2003). Fast semi-automatic generation of ontologies and their exploitation. *Technical Report, Kansas State University*.

Rizvi, S. R. A., & Wang, S. X. (2010). DT-Tree: A Semantic Representation of Scientific Papers. *Paper presented in IEEE 10th International Conference on Computer and Information Technology (CIT),* 1280 - 1284.

Roussey, C., Pinet, F., Kang, M., A. & Corcho, O. (2011). An introduction to ontologies and ontology engineering. *Ontologies in Urban Development Projects Advanced Information and Knowledge Processing Vol. 1, 2011, 9-38.*

Rujiang, B., & Junhua, L. (2009a). Improving Documents Classification with Semantic Features. *Paper presented in IEEE Second International Symposium on Electronic Commerce and Security, ISECS '09, Vol. 1,* 640 - 643.

Rujiang, B., & Junhua, L. (2009b). A Novel Conception Based Texts Classification Method. *Paper presented in IEEE International e-Conference on Advanced Science and Technology, AST '09,* 30 - 34.

Sajgal´ık, M. a., Barla, M., & Bielikov´a, M. a. (2013). From ambiguous words to key-concept extraction. *Paper presented in IEEE 24th International Workshop on Database and Expert Systems Applications*, 63-67.

Salkohe, G. (2006). Examples of ontology applications. *Seventh Agricultural Ontology Service Workshop Bangalore, India.*

Salton, G. (1971). The SMART retrieval system. *Experiments in Automatic Document Processing, Prentice-Hall, Upper Saddle River, NJ.*

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM, Vol. 18, No. 11*, 613-620.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, Vol. 34, No. 1*, 1–47.

Shahi, A. M., Issac, B., & Modapothala, J. R. (2012). Enhanced intelligent text categorization using concise keyword analysis. *Paper presented in IEEE International Conference on Innovation Management and Technology Research (ICIMTR)*, 574 - 579.

Sharma, A., & Kuh, A. (2008). Class document frequency as a learned feature for text categorization. *Paper presented in IEEE World Congress on Computational Language Processing and Knowledge Engineering*, 2988 – 2993.

Shimodaira, H., (2014). Text Classification using Naïve Bayes. *Paper presented in Learning and Data Note*, 7, 1-9.

Singh, U., Goyal, V., & Rani, A. (2014). Disambiguating Hindi Words Using N-Gram Smoothing. *An International Journal of Engineering Sciences, Issue June 2014, Vol. 10, ISSN: 2229-6913*, 1-4.

Sini, M., Salokhe, G., Pardy, C., Albert, J., Keizer, J., & Katz, S. (2007). Ontology-based Navigation of Bibliographic Metadata: Example from the Food, Nutrition and Agriculture Journal. *Food and Agriculture Organization of the United Nations, Rome, Italy.*

Shein, K. P. P., & Nyunt, T. T. S. (2010). Sentiment Classification Based on Ontology and SVM Classifier. *Paper presented in IEEE Second International Conference on Communication Software and Networks, ICCSN '10,* 169 – 172.

Song, M.-H., Lim, S.-Y., Kang, D.-J., & Lee, S.-J. (2005). Automatic classification of Web pages based on the concept of domain ontology. *Paper presented in IEEE 12th Asia-Pacific Software Engineering Conference, APSEC '05.*

Soucy, P., & Mineau, G. W. (2005). Beyond TFIDF Weighting for Text Categorization in the Vector Space Model. *In Proceedings of the 19th International Joint Conference on Artificial Intelligence IJCAI*, 1130-1135.

Tiun, S., Abdullah, R., & Kong, T. E. (2001). Automatic Topic Identification Using Ontology Hierarchy. *Paper published in Springer Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, Vol. 2004,* 444 – 453.

Turney, P. D., & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research,* 141-188.

Tong, S., Koller, D. (1998). Support Vector Machine Active Learning with Applications to Text Classification. *The Seventeenth International Conference on Machine Learning (ICML-00), Stanford, California*287-295.

Uchyigit, G. (2012). Experimental evaluation of feature selection methods for text classification. *Paper presented in IEEE 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 1294 - 1298.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review AIAI-TR 191, Vol. 11, No 2*, 1-63.

Varela, P., N. (2012). Sentiment Analysis. *Dissertation submitted for obtaining the degree of Master in Electrical and Computer Engineering*.

Verleysen, M., Rossi, F., & François, D. (2009). Advances in Feature Selection with Mutual Information *Similarity-Based Clustering*, 52-69.

Wang, B. B., McKay, R. I., Abbass, H. A., & Barlow, M. (2002). Learning text classifier using the domain concept hierarchy. *Paper presented in IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions Vol. 2*, 1230-1234.

Wang, D., & Jiang, L. (2007). An improved attribute selection measure for decision tree induction. *Paper presented in IEEE Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD Vol. 4,* 654 – 658.

Wang, L., Liu, Z.-t., Wang, Y., Sun, R., & Liu, H. F. (2009). Event Feature and Personality-Event - Ontology Based for Classifying Chinese Web Pages.

Paper presented in IEEE Second International Workshop on Computer Science and Engineering, WCSE '09 Vol. 2, 555 - 557.

Wang, N., Wang, P., & Zhang, B. (2010). An improved TF-IDF weights function based on information theory. *Paper presented in IEEE International Conference on Computer and Communication Technologies in Agriculture Engineering (CCTAE), Vol. 3*, 439 – 441.

Warintarawej, P., Laurent, A., Pompidor, P., Cassanas, A., & Laurent, B. (2011). Classifying Words: A Syllables-Based Model. *Paper presented in IEEE 22nd International Workshop on Database and Expert Systems Applications (DEXA),* 208 - 212.

Wei, G., Gao, X., & Wu, S. (2010). Study of text classification methods for data sets with huge features. *Paper presented in IEEE 2nd International Conference on Industrial and Information Systems (IIS), Vol. 1,* 433 - 436.

Wei, Z., Guo-He, F., & Zheng, N. (2011). An Improved KNN Text Classification Algorithm Based on Clustering. *Paper presented in IEEE International Conference on Internet Technology and Applications (iTAP)*, 1-5.

Wen, J., & Li, Z. (2007). Semantic Smoothing the Multinomial Naive Bayes for Biomedical Literature Classification. *Paper presented in IEEE International Conference on Granular Computing, GRC*, 648.

Wibowo, A., Handojo, A., & Halim, A. (2011). Application of Topic Based Vector Space Model with WorldNet. *Paper presented in IEEE International Conference of Uncertainty Reasoning and Knowledge Engineering (URKE), Vol. 1*, 133-136.

Wu, G., & Liu, K. (2009). Research on Text Classification Algorithm by Combining Statistical and Ontology Methods. *Paper presented in IEEE Computational Intelligence and Software Engineering CiSE*, 1-4.

Xi, L., Hang, D., & Mingwen, W. (2012). An Effective Feature Selection Tool for Text Classification. *Paper presented in IEEE Fourth International Conference on Multimedia Information Networking and Security (MINES)*, 254 - 257.

Xia, F., Jicun, T., & Zhihui, L. (2009). A Text Categorization Method Based on Local Document Frequency. *Paper presented in IEE Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD '09, Vol. 7*, 468 – 471.

Xia, T., Chai, Y., & Wang, T. (2012). Improving SVM on web content classification by document formulation. *Paper presented in IEEE International Conference on Computer Science & Education (ICCSE)*, 110 - 113.

Xiao, S., Shi, Z., Liu, K., & Lv, X. (2010). A kind of Vector Space Representation Model based on Semantic in the field of English Standard Information 2010. *Paper presented in IEEE International Conference of Computational Intelligence and Security (CIS)*, 582 - 585.

Xiaoming, D., & Yan, T. (2013). Improved mutual information method for text feature selection. *Paper presented in IEEE 8th International Conference on Computer Science & Education (ICCSE),* 163 - 166.

Xiaoyue, W., & Rujiang, B. (2009). Applying RDF Ontologies to Improve Text Classification. *Paper presented in IEEE International Conference on*

Computational Intelligence and Natural Computing, CINC '09, Vol. 2, 118 - 121.

Xu, Y. (2012). A comparative study on feature selection in Chinese Spam Filtering. *Paper presented in IEEE 6th International Conference on Application of Information and Communication Technologies (AICT),* 1-6.

Xue, C., Qiu, Q.-Y., Feng, P.-E., & Yao, Z.-N. (2010). An automatic classification method for patents. *Paper presented in IEEE Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) Conference on, Vol. 4*, 1497 – 1501.

Yan, J., Zhang, B., Liu, N., Yan, S., Cheng, Q., Fan, W., Yang, Q., Xi, W., and Chen, Z. (2006). Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing. *IEEE transactions on knowledge and data engineering, vol. 18, No. 2*, 1-14.

Yang, Y., Joachims, T. (2008). Text Categorization. *Scholarpedia*, 3(5):4242.

Yang, J., & Liu, Z. (2011). A feature selection based on deviation from feature centroid for text categorization. *Paper presented in IEEE International Conference of Intelligent Control and Information Processing (ICICIP), Vol. 1*, 180 – 184.

Yang, M., & Chen, H. (2012). Partially Supervised Learning for Radical Opinion Identification in Hate Group Web Forums. *Paper presented in IEEE International Conference of Intelligence and Security Informatics (ISI),* 96-101.

Yang, Y., & Pederson, J. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning ICML '97, 412*-420.

Yuan, M., Ouyang, Y. X., & Xiong, Z. (2013). A Text Categorization Method using Extended Vector Space Model by Frequent Term Sets. *Journal of Information Science and Engineering*, 99-114.

Yunhe, W., Yuan, G., & Chao, X. (2013). Manifold Learning Method for Large Scale Dataset Based on Gradient Descent. *Article published by Atlantis Press ISSN: 1951-6851,* 1187-1194.

Yusof, N., & Hui, C. J. (2010). Determination of Bloom's cognitive level of question items using artificial neural network. *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, 866-870.

Zhan, Y., & Chen, H. (2012a). Feature extended short text categorization based on theme ontology. *Paper presented in IEEE 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 702 - 705.

Zhan, Y., & Chen, H. (2012b). Feature extended short text categorization based on theme ontology. *Paper presented in IEEE Fuzzy Systems and Knowledge Discovery (FSKD),* 702 – 705.

Zhang, B., Xu, M., & Wu, M. (2012). Research on web filtering technology based on the dual feature selection. *Paper presented in 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 675 - 679.

Zhang, G. P. (2000). Neural Networks for Classification: A Survey. *Paper presented in IEEE Transactions on Systems, man, and cybernetics, Vol. 30, No. 4,* 451-462.

Zhang, H., & Song, H.-t. (2006). Fuzzy Related Classification Approach Based on Semantic Measurement for Web Document. *Paper presented in Sixth IEEE International Conference on Data Mining Workshops, ICDM Workshops,* 615 - 619.

Zhang, W., Yoshida, T., & Tang, X. (2008). TFIDF, LSI and Multi-word in Information Retrieval and Text Categorization. *Paper presented in IEEE International Conference on Systems, Man and Cybernetics SMC,* 108 – 113.

Zhang, X., Zhou, M., Dong, L., & Ye, N. (2009). Design of Chinese Text Categorization Classifier Based on Attribute Bagging. *Paper presented in IEEE International Conference on Business Intelligence and Financial Engineering, BIFE '09,* 201 - 204.

Zhang, W., Yoshida, T., Tang, X. & Ho, T. B. (2009). Improving effectiveness of mutual information for substantively multiword expression extraction. *Journal of Expert Systems with Applications 36*, 10919–10930.

Zhanga, W., Yoshidaa, T., & Tangb, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, Vol. 21,* 879-886.

Zhanguo, M., Jing, F., Liang, C., Xiangyi, H., & Yanqin, S. (2011). An Improved Approach to Terms Weighting in Text Classification. *Paper presented in*

IEEE International Conference on Computer and Management (CAMAN), 1 – 4.

Zhanguo, M., Jing, F., Xiangyi, H., Yanqin, S., & Liang, C. (2011). Improved Terms Weighting Algorithm of Text. *Paper presented in IEEE International Conference of Network Computing and Information Security (NCIS), Vol 2*, 367 – 370.

Zhu, D., & Xiao, J. (2011). A Variety of tf-idf Term Weighting Strategy in Document Categorization. *Paper presented in IEEE Seventh International Conference on Semantics Knowledge and Grid (SKG),* 83 – 90.

Zuo, J., Wan, M., & Ye, H. (2011). Text Classification Model Based on Markov Network Distance. *Journal of Computational Information Systems, Vol. 7: 9,* 3368-3375.