Databases and Text Processing Applications

Databases and Text Processing Applications

Editor

Ahmad Zaki Abu Bakar
Roliana Ibrahim
Norhawaniah Zakaria

Penerbit UTM Press
UNIVERSITI TEKNOLOGI MALAYSIA

www.penerbit.utm.my
2008

First Edition 2008

Perpustakaan Negara Malaysia Cataloguing-in-Publication Data

Databases and text processing applications / chief editor Ahmad Zaki b. Abu Bakar ; editors Roliana bt. Ibrahim, Norhawaniah bt. Zakaria.
ISBN 978-983-52-0631-3
1. Text processing (Computer science). 2. Database management.
I. Ahmad Zaki Abu Bakar, 1956-. II. Roliana Ibrahim. III. Norhawaniah Zakaria.
005.74

Editor: **Ahmad Zaki Abu Bakar dan Rakan-rakan**
Pereka Kulit: **Mohd Nazir Md. Basri & Mohd Asmawidin Bidin**

Diatur huruf oleh / *Typeset by*
Fakulti Sains Komputer & Sistem Maklumat

Diterbitkan di Malaysia oleh / *Published in Malaysia by*
PENERBIT
UNIVERSITI TEKNOLOGI MALAYSIA
34 – 38, Jln. Kebudayaan 1,Taman Universiti
81300 Skudai,
Johor Darul Ta'zim, MALAYSIA.
(PENERBIT UTM anggota PERSATUAN PENERBIT BUKU MALAYSIA/ MALAYSIAN BOOK PUBLISHERS ASSOCIATION dengan no. keahlian 9101)

Dicetak di Malaysia oleh / *Printed in Malaysia by*
UNIVISION PRESS SDN. BHD
Lot. 47 & 48, Jalan SR 1/9, Seksyen 9,
Jalan Serdang Raya, Taman Serdang Raya,
43300 Seri Kembangan,
Selangor Darul Ehsan, MALAYSIA.

CONTENTS

PREFACE

This book is a collection of chapters written and peer reviewed by a committee in the field of databases and text processing applications. A database is a structured collection of records or data that is organized and stored in a computer system so that it can easily be accessed, managed, and updated. The contents of a database can be bibliographic, full-text, numeric, and images. To process textual data, we have a myriad of different text processing applications such as information retrieval, text and web mining, information extraction, term recognition, text categorization, text summarization, text understanding, question answering, opinion mining and sentiment analysis, demographic analysis and dialogue systems. Using a text query application or search engine, a user is able to search document collections such as Web sites, digital libraries, or document warehouses. Its importance is growing at a tremendous rate due to the need to easily locate interesting as well as useful content on the Internet over the abundance of multilingual information.

The existence of database technology supported by computational techniques like Artificial Intelligence and statistical methods are becoming indispensable to organizations around the world. Existing data can be used to predict certain scenarios and conditions as well as detect plagiarisms in publication. Besides numerical, time series and text data, new form of data repositories

such as knowledge bases are becoming more popular. Ontology represents the concepts and metadata of objects in the knowledge bases. The emergence of ontology is also to handle the issue of heterogeneity and diversity of data from databases implemented in different platforms.

The race to create a better database and text processing application has been running for a long time since the 1990's after information is treated as a vital commodity with the rise of the information age. In Malaysia interest in databases and text processing have always been high and with economic value. Database management systems are the heart of many critical systems such as electronic banking, electronic commerce and electronic government. Machine translation, text recognition, text retrieval, and text summarizing systems have been developed in Malaysia since the 1980's. Institutions of Higher Learning like Universiti Teknologi Malaysia have also been at the forefront of such efforts. Many successful systems that started as student projects have been further developed into commercial systems and being utilized by industry. Unfortunately, not many knew this fact and the initial work laid in technical reports, thesis and conference papers are not easily accessible to the general public. The strength and wealth of research in this field would be lost if no effort is made to document them for a wider audience.

It is with this spirit that prompted the publication of this book, which comprise a collection of book chapters from various researchers in the field. As a first step and a precursor for other books in the same series, this book could only focus on only a few issues. Nevertheless, it would be an excellent source for students, researchers and practitioners as well as those interested in databases and text processing applications to understand what is the current state-of-the art in Malaysia.

Academics and post-graduate students from the Department of Information Systems, Faculty of Computer Science and Information Systems, UTM, predominantly wrote the book chapters with collaborations from many authors from other universities and agencies. Some of the work came from post-

graduate research while others came from studies carried out at various public and private sector organizations.

There are ten chapters in this book covering various aspects of databases and text processing applications. The book chapters discussed issues, techniques to support users in querying and retrieving information. It also includes chapters on methods and techniques suitable to model data for prediction. It is not possible for the co-editors to do justice and adequately summarize the contributions of the authors of the book chapters. Hence, we will only skim the book chapters and provide a few observations and comments as found below. It would be to your interest and reading pleasure to read the book chapters entirely.

Chapter 1 - Text Summarization by Mohamed Salem Binwahlan, Ladda Suanmali and Naomie Salim

This chapter discussed the evaluation measure and technique for text summarization. Text summarization is the task of rewriting text in short compressed form to represent the original text. The task is normally accomplished by human after deep reading and well understanding the document content, selecting the most important points and paraphrasing them to a concise version. Automatic text summarization is the creation of the summary by a machine. The aim of automatic text summarization is to condense the source text by extracting its most important content that meets a user or application needs. There are several ways in which one can characterize different approaches to text summarization: extractive and abstractive from single document or multi document that, summary type (informative and indicative summary), level of processing (surface level, entities level and discourse level). In this chapter, the architecture of an automatic text summarization system is presented, together with discussions on text summarization techniques that have been done for single document and multi-document summarization. Evaluating a summary is a difficult task because there is no ideal summary for a given document or set of documents. The absence of a standard human or

automatic evaluation metric made it very difficult to compare different systems and to establish a baseline. A famous evaluation measure called ROUGE is discussed in this chapter.

Chapter 2 - Classifying of Biomedical Text Abstracts For Binary And Multi-Class Support Vector Machine Using Balanced And Unbalanced Data by Rozilawati Dollah, Masaki Aono, Mohd Shahizan Othman and Roliana Ibrahim

This chapter discussed issues and techniques in classifying biomedical text abstracts, taking into account the increasing number of published biomedical articles on the Web. Based on this situation, many researchers attempt to improve the performance of classification results for finding relevant articles. Systems for finding relevant documents must be able to identify terms related to the search in the abstracts and must also distinguish between relevant and irrelevant results. Therefore, text classification systems on biomedical literature aim to select articles that match query keywords from large corpora. For this purpose, good training and testing of biomedical text datasets must be collected in order to build and validate the performance of text classification systems. This chapter focuses on the problem of identifying relevant and irrelevant biomedical text abstracts based on binary and multi-class classification using balanced and unbalanced data. In the experiments, 600 paper abstracts of four diseases, such as cancer, hepatitis, HIV/AIDS and thyroid were randomly downloaded and collected from the Medline database. The performance of binary classification and multi-class classification for both datasets using LIBSVM were then tested and compared.

Chapter 3 - Plagiarism Detection Techniques by Salha Mohammed Alzahrani and Naomie Salim

This chapter discussed issues of academic dishonesty that may

occur in academic as well as others publications. Academic dishonesty is one of the critical measures to evaluate the quality of research papers, theses and students' assignments. Therefore, plagiarism detection is an area of concern for many researchers in the academic field as well as in plagiarized news, magazine articles and web resources. Many detection techniques and tools have been developed to address the problem of plagiarism. The authors of this chapter discussed the issues and requirements for implementing plagiarism detection techniques. For example, different types of texts require different techniques to detect plagiarism. The authors described that documents to be retrieved, searched and thence judged according to the existence of plagiarism can be classified into two types.

Chapter 4 - Diversity Based Text Summarization by Mohamed Salem Binwahlan and Naomie Salim

This chapter also discussed the issues in text summarization and the techniques for improving redundancy. Diversity of selected sentences is an important factor in automatic text summarization to control redundancy in the summarized text. In this chapter, the authors proposed a method called maximal marginal importance (MMI) for text summarization. The idea for proposing this method was based on the well-known diversity approach maximal marginal relevance (MMR) where an emphasis is on the diversity. The basis of this method is on binary trees that exploit the diversity among the document sentences, where the whole document is clustered into a number of clusters, and then each cluster is presented as one binary tree or more. This chapter also discussed in detail the method adapted, where the sentence is evaluated based on its importance and its relevance. Also included in this chapter are the experimental results, which indicated that the proposed method outperform the three benchmark methods used in their study.

Chapter 5 – Comparing Prediction Models Using Rice Yields

Data by Ruhaidah Samsudin, Puteh Saad and Ani Shabri

This chapter discussed the techniques for modeling existing data available in a database for forecasting and prediction. This chapter also discussed the results of the authors' experimental work undertaken to predict rice yields. It highlighted the prediction of crop yield like wheat, corn and rice that has been an interesting research area. These types of predictions have become an important economic concern. In this chapter, the authors performed experimental work that compared three techniques, namely Artificial Neural Network (ANN), Autoregressive Integrated Moving Average (ARIMA) and Multiple Linear Regression (MLR) models in modeling the rice yields in Malaysia. The data were collected from the Muda Agricultural Development Authority in the state of Kedah in Malaysia. The data ranging from 1995 to 2001 were used to build the models. The experimental results and conclusion on the best technique for forecasting rice yield are also explained in this chapter.

Chapter 6 - Prediction Models in Dengue Data Analysis by Nor Azura Husin and Naomie Salim

This chapter discussed the approach for modeling existing time series data for predicting diseases in Malaysia. Malaysia has a good dengue surveillance system but at the moment there are insufficient findings on a suitable model to predict future dengue outbreaks since conventional methods are still being used.

This chapter explains the design of a Neural Network Model (NNM) and Nonlinear Regression Model (NLRM) using different architectures and parameters incorporating time series, location and rainfall data. The aim is to define the best architecture for early prediction of dengue outbreak. There are four architectures of NNM and NLRM developed in this study. Architecture I involved only dengue cases data, Architecture II involved combination of dengue cases data and rainfall data, Architecture III involved proximity location dengue cases data, while Architecture IV

involved the combination of all criteria. In this chapter the authors explained the parameters used in their experimental work such as the learning rate, momentum rate and number of neurons in the hidden layer. This chapter also contains the experimental results, which determine the best architecture for dengue outbreak prediction.

Chapter 7 - Peer-to-peer As An Information Resources And Searching Across The Peer-to-Peer Networks by Iskandar Ishak and Naomie Salim

This chapter provides literatures on peer-to-peer concept and discussed how this concept is useful in improving searching and managing query. Apart from peer-to-peer concept, the authors presented the classification of peer-to-peer types of networks. In their study, the authors perceived the peer-to-peer as information resources. The authors also presented the type of searching technique in peer-to-peer networks such as searching in structured and unstructured peer-to-peer networks. The conclusion of this chapter described the peer-to-peer phenomenon of becoming the ultimate information resource on the Internet.

Chapter 8 - Query Similarity Based Search In Unstructured Peer-to-Peer Networks by Iskandar Ishak and Naomie Salim

This chapter discussed the issues and approach for searching in unstructured peer-to-peer network for querying data scattered across the network. The advancements in communication technologies and cheaper cost of storage in recent years have led to the development and innovation of distributed system over the Internet. As such, the data resources are scattered across the network. Peer-to-peer technologies have then surfaced to cater for the needs of users to search and retrieve these scattered data in a quick and cost-saving manner due to the dynamic and expensive nature of the Internet. In recent years, peer-to-peer networks have become one of the media for Internet users to share resources. In a

peer-to-peer network, a peer acts as a client and a server of the system. Peer-to-peer presents an attractive solution through its scalability, fault-tolerance and autonomy. In their basic structure, peer-to-peer suffers high cost when dealing with locating content efficiently due to the use of primitive searching and routing techniques that use large overhead and long query time.

It is crucial to select relevant peers to route query message to reduce the number of messages used and answering time for better searching in unstructured peer-to-peer network without the loss of the unstructured peer-to-peer identity and characteristics. This chapter discussed the searching approach in greater length.

Chapter 9 - Ontology Extraction by Saidah Saad and Naomie Salim

This chapter provides reviews on ontology and its use in handling large volumes of information. Natural language understanding is needed to intelligently handle the large volumes of information that has grown over the last decade on the Internet. Ontology may help in analyzing and understanding text where they provide a capability to represent objects, concepts and other entities as well as the relationships between them. Ontology may be used as a tool for finding possible meanings of words in text, and meaning of text in general. Much of this ontology development has been directed towards extraction of textual data. In this chapter, the author gave a general overview and preliminary study on some of the ontology learning from text that plays a prominent role in knowledge retrieval and how the ontological semantic can be improved through the adoption of semantic web technology. This chapter also provides a comparison of approaches and techniques based on the six main layers of the different subtasks of the learning ontology systems.

Chapter 10 - XML and Relational Data Integration with CWM by Wan Mohd Hafiz b Wan Nasir, Nor Hawaniah Zakaria and Shamsul Sahibuddin

This chapter discussed the integration issues for implementing a data warehouse and the architecture designed for integration solutions. The demand for data integration is rapidly increasing with the rapid emergence of information sources in modern enterprises. Extensible Mark-up Language (XML) is fast becoming the new standard for data representation and exchange in the World Wide Web, making it necessary for data analysis tools to handle XML data as well as traditional data formats. In this chapter, the authors proposed a new architecture for a XML-based data and metadata integration in data warehouse systems for constructing OLAP cubes. This architecture uses the Common Warehouse Metamodel (CWM) for metadata interchange that incorporates a common shared metamodel to agree on metadata syntax and semantics.

The co-editors would like to thank everyone that has contributed to this book, either directly or indirectly. There are many people that deserve a special mention such as the contributors of the book chapters who were able to keep to our strict datelines; the reviewers, who provide valuable feedback to improve the chapters and their presentations; the leaders and members of the research projects as well as their collaborators, whose results became the basis of the book chapters; the team at the Department of Information Systems that worked under extreme pressure to come out with the pre-press of this book; and the team at Penerbit UTM Press, that finally transformed the manuscript into a book.

Some of the results published in this book have been achieved in the context of research projects or research collaborators. In particular we would like to mention the Ministry of Science, Technology and Innovation (MOSTI) for funding the research on rice yield prediction and the Muda Agricultural Development Authority (MADA) for contributing their data. To the State of Health Department of Selangor, especially to Dr. Nor Aini Bt

Mohd Noor, the principal Assistant Director and the Malaysian Meteorological Service for their contribution of data in disease prediction and simulation research.

We also wish to thank the Dean, Prof. Dr. Abdul Hanan Abdullah, the Deputy Dean (Development), Prof. Dr. Mohammad Ishak Desa and the Head of the Information Systems Department, Assoc. Prof. Dr. Azizah Abd. Rahman, for their continuous support and encouragement as well as Dr. Mohd Shahizan bin Othman for his numerous constructive suggestions. Finally, to Prof. Dato' Dr. Zaini Ujang, the Vice-Chancellor who challenged us to publish this book.

We truly hope you will enjoy reading this book, and sincerely welcome any comments to improve the book. We hope after reading the book, you too will be interested to submit a chapter for our forthcoming book in the same series.

Ahmad Zaki b Abu Bakar
Roliana bt Ibrahim
Nor Hawaniah bt Zakaria
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
8 December 2008

1
TEXT SUMMARIZATION

Mohammed Salem Binwahlan
Ladda Suanmali
Naomie Salim

INTRODUCTION

Text summarization is the task of rewriting text in short compressed form representing the original text. This task is accomplished by humans after deep reading and well understanding the document content, selecting the most important points and paraphrasing them to concise version. In daily life, we deal with different kinds of summaries; news headlines, abstract of scientific publication, search results retrieved by a search engine, reviews of movies and overview of books, and so on (Mani, 2001). Newspaper headlines are a natural example of human summarization.

Automatic text summarization is the creation of the summary by a machine. The aim of automatic text summarization is to condense the source text by extracting its most important content that meets a user's or application's needs (Mani, 2001). Summarization is a challenging problem because the characteristics of informativeness, readability, robustness, and length reduction. Those factors must be taken into account when dealing with this problem (Melander, 1993).

This chapter provides reviews over some existing works for text summarization. The next sections of the chapter discuss text summarization basic concepts, summary types, architecture of an automatic text summarization system, text summarization applications and techniques, and evaluation measurement of text summarization.

TEXT SUMMARIZATION BASIC CONCEPTS

The basic concepts used in the field of automatic text summarization are introduced below (Lamkhede, 2005):

- Coherence: A summary was said to be coherent if all its sentences crucial antecedents to form an integrated whole and the sequence of ideas progressed logically.
- Compression Rate: It is a ratio of summary length to source length to express the degree of summarization required. It is calculated as
  - $$\frac{\text{SummaryLength}}{\text{SourceLength}} \quad (1)$$
- Salience or Relevance: It is the score of the information in a document, reflecting both the document content as well as the relevance of the document information to the user's or application's need.
- Compaction of text: It is process of removing less salient phrases or words from sentences.
- A generic summary: It presents the main topics or an overall sense of the document's content.
- A query-relevant summary: It contains the content of the document that is closely related to the user's query. It is also called as topic specific summary.
- Critical summary: It contains the abstractor's opinions towards the quality of the source for evaluation purpose.
- A summarizer: It is a system that creates the summary.

- Monolingual Summarizer: It uses just one language for input and output.
- Multilingual Summarizer: It has ability to use many languages with output in the same language as input.
- Cross-lingual Summarizer: It has ability to use many languages with output in different language from input.
- Single Document Summarizer: It summarizes one document and produces single summary.
- Multi-document Summarizer: It summarizes many documents and produces single summary.

SUMMARY TYPES

There are four types of summaries, which are indicative summary, informative summary, critical summary, and extract. The most important summary types are indicative summary and informative summary.

An informative summary represents (and often replace) the original document by containing all the pertinent information necessary to convey the core information and omit ancillary information.

An indicative summary is a condensed version of the contents of the article without giving away detail on the article content that can serve to entice the user into retrieving the full form (e.g. book jackets, card catalog entries, movie trailers, headline, scientific abstract).

Summaries generated can contain information from a single document (single document summaries) or a collection of documents (multi-document summaries). Summarization process emphasizes mainly on two goals, high compression ratio and redundancy reduction especially in multi-document summarization. These goals are achieved by keeping the important ideas in each document, reducing the size of each document and comparing

ideas across documents. This poses many challenges include (Mani and Maybury, 1999):

- Identifying scaling algorithms (which scale up to large-size collection).
- Eliminating the redundancy.
- Intelligent ways are employed to exploit ordering among documents.
- The relationships are represented by effective presentation and visualizations strategies.

ARCHITECTURE OF AN AUTOMATIC TEXT SUMMARIZATION SYSTEM

Figure 1 represents architecture of an automatic text summarization system. The input to the summarization process could be a single or multiple document, text or multimedia information such as imagery, audio, or video. Text summarization has focused mainly on text input (which can be a representation of some other media). The automatic text summarization process consists of three stages (Mani, 2001):

- Analyzing stage utilizes linguistic and semantic information to determine facts about the input text. This requires some level of understanding of the words and their context (discourse analysis, part of speech tagging, etc.)
- Transformation stage uses statistical data and semantic models to generalize the input text and transform it into a summary representation.
- Synthesizing stage depends on the information created from the previous two stages to synthesize an appropriate output form.

These three stages include three basic condensation operations used in summarization:

- Selection of more salient or non-redundant information.

- Aggregation of information (e.g. from different parts of the source, or of different linguistic descriptions).
- Generalization of specific information with more general, abstract information.

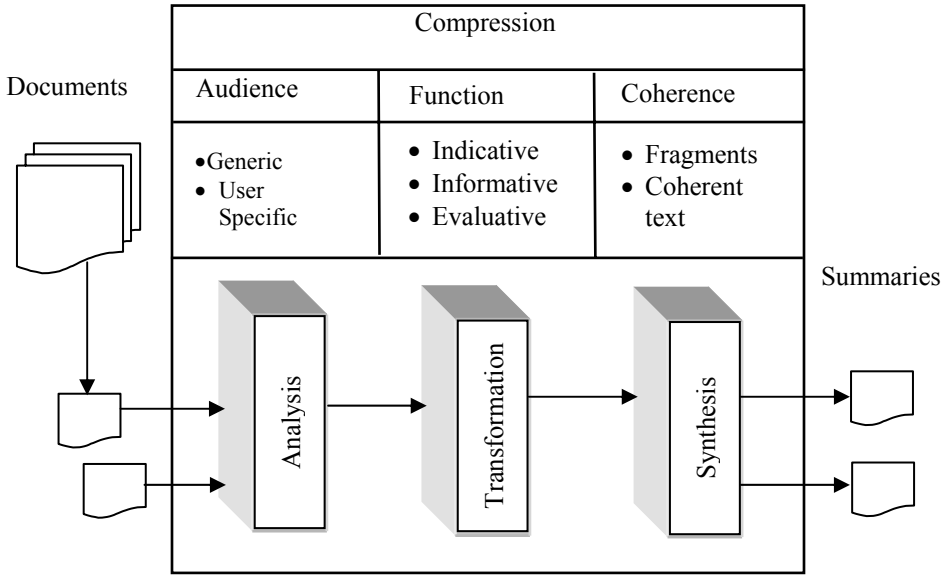

Figure 1 Architecture of an Automatic Text Summarization System

SUMMARIZATION APPLICATIONS

The summarization has gained importance not only in the written news but also in other novel domains including:

- Voice mails, Koumpis and Renals (2005) proposed that summary words are identified through a set of classifiers, with each word being identified by a vector of lexical and

prosodic features. The short text summaries generated are suitable for mobile messaging applications.

- Multi-party dialogs, Zechner (2002) presented a dialogue summarization system for automatically creating extract summaries for open-domain spoken dialogues in multiparty conversations.
- Newsgroups, Newman and Blitzer (2003) described an approach to condense the threads of archived discussion lists; they clustered messages into topic groups, and then extract summaries for each messages group.
- Blogs, Zhou and Hovy (2006) described computational approaches to summarize two types of dynamically introduced information: online discussions and blogs.

TEXT SUMMARIZATION TECHNIQUES

Summarization techniques can be classified based on many different approaches. One approach is by examination of the processing level. In this approach, summarization techniques can be classified as the surface, entity, or discourse level approaches (Mani and Maybury, 1999).

Surface-level approaches exploit shallow features to identify a salience function to extract the most important information. These features include thematic features (statistically salient terms based on term frequency), location feature (position in text, position in paragraph, section depth, particular sections), background features (presence of terms from the title or headings in the text, the initial part of the text, or user's query), and cue words and phrases (e.g., in-text summary cues such as "in summary," "our investigation," emphasizes such as "important," "in particular," as well as domain-specific "bonus" and "stigma" terms).

Entity-level approaches try to model the text entities and their relationships through building an internal representation for

text. The patterns of connectivity in the text also are represented by these approaches (e.g., graph topology) to help determine what is salient. The relationships among the entities can be classified into similarity (e.g., vocabulary overlap), proximity (distance between text units), co-occurrence (i.e., words related based on their occurring in common contexts), logical relations (such as agreement, contradictions, and consistency) and syntactic relations (e.g. based on parse trees).

In discourse-level approaches for communicative goals, the global structure of the text and its relations are modeled by discourse-level approaches. The global structure of the text can include format of the document (e.g., hypertext markup, document outlines), threads of topics (as they are revealed in the text) and rhetorical structure of the text (such as argumentation or narrative structure).

Single Document Summarization

The work on summarization began as early as fifties when Luhn (1958) presented the first summarization research. It is considered as the cornerstone for all works which followed it, that work was accomplished at IBM. Luhn mentioned that the significance of a sentence is gained from an analysis of its words. He proposed that word significance is determined by frequency of its occurrence and the significance of sentence is determined by the relative position of its words, thus both factors can serve as useful measurements. Combination of these two measurements determines the significance factor of a sentence. A preprocessing in this system was to remove stop words and stem the words to their roots. The top ranking sentences are selected to be included in the summary "auto-abstract", where all sentences are ranked based on order of their significance.

In the same year (1958) and same place (IBM), a work related to Luhn was done by Baxendale (1958). In Baxendale's

study, the sentence is selected as a candidate to be included in the summary based on its position. The sentence appearing in the beginning and the end of the paragraph has been given more significance. This assumption was justified by testing 200 paragraphs and found 85% of the paragraphs start with a topic sentence and 7% ends with a topic sentence. Later many complex machine learning based systems used same feature (position) examined in Baxendale's study.

Edmundson (1969) presented a summarization system to generate extracts in which four features used; two of them have been used in the previous studies (i.e. word frequency used by Luhn (1958) and positional importance used by Baxendale (1958)) and the other two features were pragmatic words (cue words, i.e., words would have positive or negative effect on the respective sentence weight like significant, key idea, or hardly) and title and heading words. Each sentence was scored by the weights of the four features, where each feature was given the weight manually. Three evaluation schemes were used in this study and the results showed that 44% of the auto-extracts were correlated to the manual extracts.

The appearance of machine learning methods in NLP in 1990s encouraged many researchers to use widely statistical techniques in the summarization to generate extract. Kupiec *et al.* (1995) developed a system called it "A Trainable Document Summarizer" based on Bayesian classifier algorithm where five weighting heuristics employed in the system. Below more details about those features:

- Sentence length cut-off feature: the sentence consists of a number of words less than predefined threshold be excluded from the summary.
- Uppercase word feature: Proper names considered as an uppercase thematic word under some conditions.
- Paragraph feature: this is for sentence position in the paragraph (initial, final or medal).

- Fixed-phrase feature: Sentence including any of certain cue words or appearing directly after a section header comprising a keyword is included in the summary.
- Thematic word feature: the thematic words are the most frequent words and their function of frequency is the sentence score.

Based on those features, the score of each sentence calculated using Bayesian classifier algorithm where the classification algorithm computes the probability of each sentence. The decision about including the sentence in the summary or excluding it made based on its probability. If the sentence probability equal 1 means the including decision taken, if 0 excluding decision taken. The features paragraph feature, fixed-phrase feature and thematic word feature have been used previously by Edmundson (1969).

The position feature was investigated by Lin and Hovy (1997) through introducing a method called "position method". The researchers built their method on the idea of the most important sentences tend to appear in fixed locations like title. However, the position method can't be defined as directly as Baxendale (1958) did because the discourse structure depends significantly on genre and domain. For this reason, the study investigated on how to tailor the position method to genre and domain and evaluate it for effectiveness. The position method works through determining the sentence score by its position in the text. The manual topic words were used to calculate the yield of each sentence position. Then the sentence positions are ranked by their average yield to generate the Optimal Position Policy (OPP) for topic position for the genre. The method evaluation is used to confirm the goal of creating an Optimal Position Policy that is to adapt the position hypothesis to various domains or genres in order to achieve maximal topic coverage. Baxendale's first/final sentence hypothesis (1958) was examined by this method and the achieved results confirmed the first sentence hypothesis and do not confirm

final hypothesis; it was found that the second sentence from the end of a paragraph contains the most information.

Lin (1999) examined decision tree as machine learning method, the goal of his study was to investigate the influence of the topic importance and query type on the performance of the heuristics, many heuristics used in the study: Baseline (normalized scoring based on the sentence position in the text; the highest score sentence is first and lowest is the last), Title, TF and TFIDF scores, Position score, Query signature, IR signature (score given to sentences depending on number and scores of the *m* most salient terms contains, those terms that occur more often in the top *n* retrieved sentences), Sentence length, average lexical connectivity (score is the number of terms which the sentence sharing with others divided by the total number of sentences in the text) and a boolean value 1 is given to sentences that include numerical data, proper name or pronoun and adjective. The resulting system from this study called "SUMMARIST" which developed at the University of Southern California to extract sentences from the documents and those were matched against ideal human-made extract, like what most previous works on extractive summarization did.

Conroy and O'leary (2001) proposed a method to produce the generic extracts using a hidden Markov model that decides the likelihood of the including sentence in/excluding the sentence from the summary. The probability of the sentence to be a summary sentence or not is calculated through a set of features:

- Position of the sentence in the document. This feature is built into the state-structure of the HMM.
- Number of terms in the sentence. The value of this feature is:

$$o1(i) = \log(\text{number of terms} + 1) \qquad (2)$$

- How likely sentence terms are, given the document terms:

$$o2(i) = \log(\Pr(\text{terms in sentence}_i \mid D)) \qquad (3)$$

Osborne (2002) proposed a system based on log-linear model for the sentence extraction task. He mentioned that his

system ability is to employ dependencies which will likely exist among the heuristics. Such ability is necessary especially when more sophisticated heuristics, with complicated interactions, are brought to bear upon the problem. The available summarization systems don't have such ability because they have always assumed feature independence. To avoid the assumption of feature independence, the researcher used log-linear model and showed empirically that the system produced better extracts than a naive-Bayes model, with a proposed prior to both models. The features used in the study are:

- Word pair feature: It simply indicates whether a particular word pair (consecutive words) exists in the sentence.
- Sentence length: It is encoded in three binary features whether a sentence length(less than 6 words, greater than 20 words, or between the two ranges). Another feature used to indicate whether a previous sentence was less than 5 words or longer.
- Sentence position: It is determined by three features to indicate whether a given sentence exists in (the first 8 paragraphs, or in the last 3 paragraphs, or in a paragraph between these two ranges).
- Limited discourse features: To indicate whether a sentence immediately followed typical headings (such as conclusion or introduction), at the start of a paragraph, or followed some generic heading.

Svore *et al.* (2007) proposed a system to perform a task of newswire articles summarization which could defeat the extremely strong baseline (which no previous summarization could outperform it) for summarizing a newswire article. The system is designed using a neural network algorithm as a machine learning approach to perform the summarization task. Data set (third party) contains 1365 news documents. Each document includes the title, timestamp, story highlights, and article text. The story highlights are human-generated from the article text. Three highlights were planned to extract from each article because all articles include at least 3 story highlights. Ten features used for each sentence in each

document: first sentence, sentence position, word frequency, bigrams score, and the sentence similarity, all these feature used in previous works. The other features used are based on third party data sources (news search engine and Wikipedia). The chance of a sentence based on whether it contains keyword used in anyone of the two third party data sources.

A method presented by Barzilay and Elhadad (1997) generated a summary based on reasonable size of source representation, not full text (as done by McKeown and Radev (1995)) nor just words from text (as done by Luhn (1958)). The researchers used lexical chains as source representation, where lexical chain is a sequence of related words where lexical cohesion occurs among them. Lexical cohesion is meant for sticking together different parts of the text. The cohesion here means the using of semantically related terms. The summarization task in this method performed through three phases: to segment the original text, to form the lexical chain, to identify the strong lexical chains and using the strong lexical chains the most important sentences extracted. Wordnet (Miller, 1995) was used to discover the lexical chains.

Ono *et al.* (1994) proposed a domain-independent abstract generation system based on computational model of discourse for Japanese expository writings; the system extracts discourse structure by one of its forms which are rhetorical structure that represents relations between different chunks of sentences in each unit. The rhetorical structure depends on two structures: intra-paragraph, where its representation units are sentences, and inter-paragraph that represents units as paragraphs and represented by a binary-tree.

Marcu (1998a) criticized all previous multiple heuristic based systems that they deal with the text which required to be summarized as plane sequences of sentences. In those systems, sentence position and semantic similarity with the title play an important role in the scoring of the sentence regardless whether the sentence bearing the main idea of the text or not. Based on this observation the researcher found that paying attention to advantage

of discourse structure can overcome that shortage. A tight coupling of the structure of discourse and a set of summarization heuristics that are employed by current systems provides two advantages which the previous systems couldn't achieve them. First, to learn genre-specific combinations of heuristics can be used for disambiguation during discourse parsing. Second, constructing discourse structures for generating summaries containing textual units where the importance of those units depends on both some heuristics, and relatedness to the main theme of texts. Rhetorical Structure Theory (RST) is the discourse theory used in that study which is a relation (nucleus and the satellite) that holds between two non-overlapping chunks of text spans. This means that the discourse structure is built by two kinds of nodes, one is nucleus and the other is satellite. The discourse structure built by the rhetorical parsing algorithm: the text is divided into ten primary units, next the cue phrases and a simple notion of semantic similarity used for finding out the rhetorical relations among the primary units. Finally, the rhetorical relations can be gathered into rhetorical structure trees. To determine the 'best' discourse interpretations, there were seven metrics used to score each discourse. A weighted linear combination of all these scores gave the score of a discourse structure. To find the best combination of heuristics, the author computed the weights that maximized the F-score on the training dataset, which was constituted by newswire articles. To do this, he used a GSAT-like algorithm that performed a greedy search in a seven dimensional space of the metrics.

Multi-Document Summarization

A new challenge became more interesting by the mid 1990s, which is generating a summary for multi-documents. Generating such summaries must take into account: keeping the important ideas in each document, reducing the size of each document and comparing ideas across documents.

The first real participation on this challenge presented by McKeown and Radev (1995) they developed a system named it "SUMMONS" which was made as extension of Message Understanding systems. It deals with single events about a narrow domain (e.g. news articles about terrorism). A typical language generator used in their study is divided into two main components, a content planner (produces a conceptual representation of text meaning (e. g., a frame)), which selects information from an underlying knowledge base to include in a text, and a linguistic component (uses a lexicon and a grammar of English), which selects words to refer to concepts contained in the selected information and arranges those words, appropriately inflecting them, to form an English sentence. The system summarizes a series of news articles, in form of templates produced by the message understanding systems, on the same event, producing a paragraph consisting of one or more sentences.

The multi-document system restricted on narrow domain done by McKeown and Radev (1995) was a motivation for researchers to think about broader domain. The framework in (McKeown and Radev, 1995) was improved by McKeown *et al.* (1999). The new system consists of two main parts, the analysis component and the generation component. In summarization, the analysis component uses a set of related documents as input and breaks them into smaller text units (i.e. paragraphs). Then the similarities between the text units are calculated as following: a set of features is extracted; based on those features a vector (representing matches on each of the different features) for each pair of paragraphs is created; a machine learning algorithm (William, 1996) uses the created vectors to produce decision rules to classify each pair of text units either as similar or dissimilar; a subsequent clustering algorithm uses those decisions to collect the most related text units together in one group called theme. The extracted features included word co-occurrence (i.e. sharing of a single word between text units), matching noun phrases, WordNet synonyms (i.e. synonyms words that appear in the same synset are matched), common semantic classes for verbs (i.e. two verbs that

share the same semantic class are matched), and TF*IDF (i.e. weight of each word). The second task in summarization, which is making a concise and fluent fusion of information in each theme and highlighting the common facts among all its text units, is performed by generation component. The theme sentences are parsed into phrases using a statistical parser (Michael, 1996). The rule-based component developed by the author converts the phrase-structure output of Collins' parser to dependency grammar in order to create functional roles (e.g., subject, object) and capture predicate argument structure. The comparison algorithm traverses all created sub-trees rooted at verbs recursively searching for each two identical nodes, if found, are added to the output tree, and their children are compared. Once a full phrase (verb with at least two constituents) has been found, it is confirmed for inclusion in the summary. After determining the summary content (represented as predicate-argument structures) is decided, a grammatical text is generated by translating those structures into the arguments expected by the FUF/SURGE language generation system.

Carbonell and Goldstein (1998) proposed topic-driven summarization method called maximal marginal relevance (MMR) for combing query-relevance with information-novelty in the context of text retrieval and summarization.

Radev *et al.* (2000) proposed a multi-document summarizer called MEAD, in which the system depends on cluster centroids. Cluster centroid was previously generated by other system called CIDR (Dragomir *et al*., 1999) that uses modified TF*IDF to generate clusters of news articles on the same even. It uses the clusters of centroids as input to determine which sentences are central to the topic of the cluster, rather than the individual articles. The author presented two metrics to evaluate the sentences: cluster-based relative utility (CBRU) and cross-sentence informational subsumption (CSIS). The first accounts for how relevant a particular sentence is to the general topic of the entire cluster; the second is a measure of redundancy among sentences. The difference between these metrics and those used in MMR is MMR metrics are query-dependent. Given one cluster C of

documents segmented into n sentences, and a compression rate R, a sequence of nR sentences are extracted in the same order as they appear in the original documents, which in turn are ordered chronologically. The selection of the sentences is made by approximating their CBRU and CSIS. For each sentence si, three different features are used, centroid value, positional value and first-sentence overlap. The final score of each sentence is a combination of the three scores above minus a redundancy penalty (Rs) for each sentence that overlaps highly ranked sentences.

EVALUATION

Evaluating a summary is a difficult task because there is not an ideal summary for a given document or set of documents. The absence of a standard human or automatic evaluation metric makes it very hard to compare different systems and establish a baseline. In this section, the famous evaluation measure called rouge is discussed.

ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit Lin (2004b) is used for evaluating the text summarization methods, where ROUGE compares a system generated summary against a human generated summary to measure the quality. ROUGE is found to be the most appropriate evaluation metric; it is the main metric in the DUC text summarization evaluations. ROUGE has the following measures: ROUGE-N (N is the number of ngrams), ROUGE-L, ROUGE-W with weighting factor $\alpha = 1.2$, ROUGE-S and ROUGE-SU (maximum skip distance dskip = 1, 4, and 9).

ROUGE-N calculates the shared ngrams between system generated summary and one or a set of human generated summaries producing recall score;

$$\frac{\sum_{S \in \{Reference summaries\}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in \{Reference summaries\}} \sum_{gram_n \in S} count(gram_n)} \quad (4)$$

where n is the length of the n-gram and $count_{match}$ is the most possible number of n-grams shared between a system generated summary and a set of reference summaries.

ROUGE-L measure calculates the longest common subsequence (LCS). Suppose we have two sentences X and Y, and LCS is a common subsequence with maximum length. ROUGE-L is defined as an LCS based F-measure:

$$\text{ROUGE-L(s)} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \quad (5)$$

$$R_{LCS}(\text{s}) = \frac{\sum_{i=1}^{u} LCS(r_i, s)}{\sum_{i=1}^{u} |r_i|} \quad (6)$$

$$P_{LCS}(\text{s}) = \frac{\sum_{i=1}^{u} LCS(r_i, s)}{|s|} \quad (7)$$

Where |x| denotes the length of sentence x, LCS(x, y) denotes the length of the LCS between sentences x and y, and β is a (usually large) parameter to balance precision and recall.

The ROUGE evaluation measure (version 1.5.51) generates three scores for each summary: recall, precision and F-measure (weighted harmonic mean, Eq. 8), in the literature, we found that the recall is the most important measure to be used for comparison purpose, so we will concentrate more on the recall in this evaluation.

$$F=\frac{1}{\left(alpha\times\left(\frac{1}{P}\right)+\left(1-alpha\right)\times\left(\frac{1}{R}\right)\right)} \quad (8)$$

Where P and R are precision and recall respectively, and alpha is parameter to balance between precision and recall.

SUMMARY

The main goal of any automatic text summarization is to generate high quality summary that depicts the document/s content in short form. Since the beginning of work on summarization, many efforts are dedicated to achieve that goal. The first work on summarization which was done by Luhn in late of fifties (1958) is considered as the cornerstone for all followed works. All works which have been done before the nineties depended on simple techniques for generating summaries based on some document features, but in the second mid of nineties the summarization techniques took new direction by exploiting machine learning

[1] http://haydn.isi.edu/ROUGE/latest.html

methods natural language processing (Kupiec et al., 1995; Lin and Hovy, 1997; Lin, 1999; Conroy and O'leary, 2001; Osborne, 2002; Svore et al., 2007).. After that the natural language method became deeper, where the attention was paid to exploiting the discourse structure (Barzilay and Elhadad, 1997; Ono et al., 1994; Marcu, 1998a). The challenging summarization task is the evaluation of a summary because there is not an ideal summary for a given document or set of documents. The absence of a standard human or automatic evaluation metric makes it very hard to compare different systems and establish a baseline. The standards of automatic evaluation of summaries were proposed by Lin (2004b). They are a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE).

REFERENCES

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97).* August. Madrid, Spain: ACL, 10-17.

Baxendale, P. (1958). Machine-made index for technical literature - an experiment. *IBM Journal of Research Development*. 2(4), 354-361.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 24-28 August. Melbourne, Australia, 335-336.

Conroy, J. M. and O'leary, D. P. (2001). Text summarization via hidden markov models. *Proceedings of SIGIR '01*. 9-12 September. New Orleans, Louisiana, USA, 406-407.

Dragomir R. Radev, Vasileios Hatzivassiloglou, and Kathleen R. McKeown (1999). A description of the CIDR system as used for TDT-2. *DARPA Broadcast News Workshop*. February. Herndon, Virginia.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery*. 16(2), 264-285.

Koumpis, K. and Renals, S. (2005). Automatic summarization of voicemail messages using lexical and prosodic features. *ACM Transactions on Speech and Language* Processing. 2(1), 1–24.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. *In Proceedings of the ACM. SIGIR conference*. July. New York, USA, 68-73.

Lamkhede, S. (2005). *Multi-Document Summarization Using Concept Chain Graphs.* Master Thesis. State University of New York, New York.

Lin, C. Y. and Hovy, E. (1997). Identifying topics by position. *In Proceedings of the Fifth conference on Applied natural*

language processing. March. San Francisco, CA, USA, 283-290.

Lin, C. Y. (2004b). Rouge: A package for automatic evaluation of summaries. . Proceedings of the Workshop on Text Summarization Branches Out, 42nd Annual Meeting of the Association for Computational Linguistics. 25–26 July. Barcelona, Spain, 74-81.

Lin, C. Y. (1999). Training a selection function for extraction. In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM)*. 2-6 Nov. Kansas City, Kansas, 55-62.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*. 2(92), 159-165.

Mani, I. and Maybury, M. T. (1999). *Advances in Automatic Text Summarization*. Cambridge, MA: The MIT Press.

Mani, I. (2001). *Automatic Summarization*. Amsterdam: John Benjamins Publishing Company.

Marcu, D. (1998a). Improving summarization through rhetorical parsing tuning. *Proceedings of The Sixth Workshop on Very Large Corpora*. August. Montreal, Canada, 206-215.

McKeown, K. R. and Radev, D. R. (1995). Generating summaries of multiple news articles. *Proceedings, ACM Conference on Research and Development in Information Retrieval SIGIR'95*. July. Seattle, Washington, 74-82.

McKeown, K., Klavans, J., Hatzivassiloglou, V., Barzilay, R., and Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. *Proceedings of American Association for Artificial Intelligence (AAAI)*. July. Orlando, Florida, 453-460.

Melander, N. M. (1993). *Multiple Document Summarization for Written Argumentative Discourse*. Master Thesis. Johns Hopkins University.

Michael, C. (1996). A New Statistical Parser Based on Bigram Lexical Dependencies. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. June. Santa Cruz, California, 184 – 191.

Miller, G. A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*. 38(11), 39-41.

Newman, P. S. and Blitzer, J. C. (2003). Summarizing archived discussions: a beginning. *Proceedings of the 8th international conference on Intelligent user interfaces*, January 12-15. Miami, Florida, USA, 273-276

Ono, K., Sumita, K., and Miike, S. (1994). Abstract generation based on rhetorical structure extraction. *Proceedings of 15th International Conference on Computational Linguistics (COLING'94)*. 5-9 August. Kyoto, 344-348.

Osborne, M. (2002). Using maximum entropy for sentence extraction. *Proceedings of the ACL'02 Workshop on Automatic Summarization*. July. Morristown, NJ, USA, 1-8.

Radev, D. R., Jing, H. and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. *NAACL-ANLP 2000 Workshop on Automatic summarization*. April. Morristown, NJ, USA, 21-30.

Svore, K., Vanderwende, L., and Burges, C. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. June. Prague: Association for Computational Linguistics, 448–457.

William, C. (1996). Learning Trees and Rules with Set-Valued Features. *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96)*, 1, 709-716 . American Association for Artificial Intelligence.

Zechner, K. (2002). Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*. 28(4), 447-485.

Zhou, L. and Hovy, E. (2006). On the summarization of dynamically introduced information: Online discussions and blogs. *Proceedings of AAAI Spring Symposium on Computational Approaches to Analysing eblogs*. 27-29 March. Stanford, California.

2
CLASSIFYING BIOMEDICAL TEXT ABSTRACTS FOR BINARY AND MULTI-CLASS SUPPORT VECTOR MACHINE USING BALANCED AND UNBALANCED DATA

Rozilawati binti Dollah
Masaki Aono
Mohd Shahizan Othman
Roliana Ibrahim

INTRODUCTION

Overwhelming amount of published biomedical knowledge in texts, especially in Medline database makes it difficult for the researcher to effectively and efficiently organize and retrieve relevant information. For this purpose, systems for finding relevant documents must be able to identify terms related to the search in the abstracts and also must distinguish between relevant and irrelevant results (Leonard et al. 2002). Therefore, text classification systems on biomedical literature aim to select relevant articles to a specific issue from large corpora (Couto et al. 2004). Text classification systems must automatically extract the features that help determine positives from negatives and apply

those features to candidate documents using some kind of decision-making process (Cohen and Hersh 2005).

The text classification task can be defined as assigning category labels to new documents based on the knowledge gained in a classification system at the training stage (Remeikis et al. 2004). It is called "binary" if it assigns a given documents into one of two classes either positive or negative class, meanwhile it is called "multi-class" if it assigns a given documents into one of k classes. Currently, many researchers attempt to investigate and develop more applicable and effective way for classifying biomedical text articles in order to help users find relevant articles on the web. Several of statistical classification methods and machine learning techniques have been applied to text classification including techniques based on Decision Tree (Lewis and Ringuette 1994), Neural Network (Wiener et al. 1995) and Support Vector Machine (SVM) (Mahinovs and Tiwari 2005). SVM has been prominently and widely used for binary and multi-class classification.

The goal of automatic text classification is to learn a classification scheme from training examples of previously classified documents. The learned scheme can then be used to classify future text documents automatically (Ho and Lam 1998). Due to this reason, our focus is on the problem of identify relevant and irrelevant biomedical text abstracts based on binary and multi-class classification, especially in diseases category using balanced and unbalanced data. In this paper, we use the approach that involved term and word frequencies to calculate a score of biomedical paper abstracts, contains four categories of diseases, namely cancer, hepatitis, HIV/AIDS and thyroid diseases.

We choose these diseases due to the number of patients who suffer from these critical diseases were increased lately. Other than that, the awareness among the individuals to get more information about these diseases caused them trying to find the related articles. At the same time, the increasing number of researches on these diseases also influence on this matter (Dollah and Aono 2008).

For the purpose of this study, we have conducted several experiments to compare the performance of binary classification and multi-class classification based on different percentage of training and testing of biomedical paper abstracts dataset using balanced and unbalanced data. Moreover, we also have experimented with both training and testing dataset (with scaling and without scaling).

METHODOLOGY

In order to perform binary and multi-class classification experiments, we have downloaded and collected 600 paper abstracts of four disease categories, which are cancer, hepatitis, HIV/AIDS and thyroid from Medline database using PubMed web site. Then, we have performed the text pre-processing in order to prepare binary and multi-class classification datasets. A main goal of text pre-processing is to transforms the text string representation into numeric feature vectors, where we represent our documents as Vector Space Model. In our experiments, the text pre-processing phase includes, stop word elimination, word stemming, word or feature selection and weighting (Dollah and Aono 2008). Then, a set of vector will be extracted from the collected biomedical paper abstracts after we were finished the text pre-processing.

Stop Word Elimination

The first phase in text pre-processing is eliminating stop word. The main objective of stop word elimination is to purge the list of words from "noise". In this phase, we generated a list of words for each biomedical paper abstract. Then, we eliminated words such as articles (*a, an, the*), preposition (*in, of, at*), conjunction (*and, but,*

or, nor), pronouns *(I, you, them, it*) and etc from each paper abstracts. For this purpose, we used a standard stop-word list.

Word Stemming

Word stemming is a process of reducing a word to it stem or root form by removing suffixes. For instance, the word "finding" which is stemmed as "find". Thus, the keywords of a query or paper abstract are represented by root form rather than by the original words. This phase was performed for each list of words to increase words or features coverage, which will increase the classification accuracy. In our experiments, we employed the Porter's stemming algorithm. After that, we created the vocabulary by combining a list of words that describes all biomedical paper abstracts for our experiments.

Word Selection

Word stemming is followed by word selection. In this phase, the words that appear below than three times in the vocabulary will be removed and the rest will be considered as feature vector and then, will be calculated the weight. The purpose of word selection is to reduce the total number of words in the vocabulary for weighting process and also to remove noise from the dataset in order to optimize the classification performance. And the learning process for a large amount of data is time consuming. Table 1(a), (b) and (c) below indicate the statistic of words that appear in the vocabulary before and after the word selection done.

Table 1(a) Statistics for the dataset (balanced data)

Category	Dataset	BWS	AWS	Difference
Multi-class	Diseases	6081	2693	3388

classification				
Binary classification	Cancer	4070	1721	2349
	Hepatitis	4127	1717	2410
	HIV/AIDS	4227	1719	2508
	Thyroid	4131	1696	2435

Table 1(b) Statistics for the dataset (unbalanced data – 150P, 50N)

Category	Dataset	BWS	AWS	Difference
Multi-class classification	Diseases	6317	2812	3505
Binary classification	Cancer	4010	1690	2320
	Hepatitis	3937	1694	2243
	HIV/AIDS	3823	1593	2230
	Thyroid	4378	1800	2578

Table 1(c) Statistics for the dataset (unbalanced data – 40P, 160N)

Category	Dataset	BWS	AWS	Difference
Multi-class classification	Diseases	6317	2812	3505
Binary classification	Cancer	4284	1810	2474
	Hepatitis	4309	1790	2519
	HIV/AIDS	4122	1754	2368
	Thyroid	4050	1669	2381

Legend: BWS- Total number of word (Before Word Selection)
AWS- Total number of word (After Word Selection)

Weighting

This is the final phase in the implementation of text pre-processing. In this phase, the weight for each word or feature in the vocabulary will be calculated using the TFIDF formulation. TFIDF is the most

common weighting method used to describe documents in the Vector Space Model (Soucy and Mineau 2005). This method used to calculate weight for each word or feature that appear in each biomedical paper abstracts.

EXPERIMENT AND RESULT

The aim of this chapter is to evaluate the accuracy of binary and multi-class classification for biomedical paper abstracts using balanced and unbalanced data (paper abstracts). For this purpose, we have conducted an experiment using 600 paper abstracts that involved four categories of diseases, randomly downloaded from Medline database. Each category of disease consists of 150 paper abstracts. Then, we have performed several experiments using LIBSVM (Chang and Lim 2001) because it supports binary and multi-class classification.

The performance evaluation of text classifier is conducted on a testing dataset which is different from the training dataset. For this purpose, we keep the same dataset, but change the percentage for training and testing dataset into seven groups. In each group, we randomly selected document data vectors from dataset and put them into training and testing data, respectively. The details of percentage for each group are as follows (refer Table 2):

Table 2 Experiment vs training and testing data

Experiment	Training data	Testing data
I	90%	10%
II	80%	20%
III	75%	25%
IV	70%	30%
V	60%	40%
VI	50%	50%
VII	40%	60%

For all the experiments, we have applied five-fold cross validation for training process. Then, we tested each experiment for evaluating the performance of binary and multi-class classification using balanced data. After that, we have repeated runs each experiment to assess the performance of classification for unbalanced data using Radial Basic Function (RBF) kernel in LIBSVM. In the RBF kernel, there are two parameters to be determined in the SVM model, which are C (cost) and γ (gamma). Finally, we have compared the results of classification based on the accuracy for all datasets done in these experiments.

Experiments Using Balanced Data

In balanced data experiments, we have implemented several experiments using binary classification dataset and multi-class classification dataset. For each binary classification dataset, we used 100 positive and 100 negative of paper abstracts, meanwhile for multi-class classification dataset, we combined all of 400 paper abstracts (Dollah and Aono 2008). In all the experiments, we have used two types of dataset, which are (with scaling) dataset and (without scaling) dataset. Table 3 and Figure 1 below show the accuracy of binary classification for balanced data, respectively.

Table 3 A result of binary classification experiments using balanced

Dataset	Experiment	I	II	III	IV	V	VI	VII
Dataset (without scaling)	Cancer	95	87	86	80	83	87	85
	Hepatitis	85	87	80	88	92.5	83	64.17
	HIV/AIDS	100	92	90	95	92.5	89	85
	Thyroid	80	90	88	93.34	87.5	83	85.83
Dataset	Cancer	80	80	92	76.67	85	75	72

(with scaling)	Hepatitis	75	82.5	78	66.67	83.75	64	75.83
	HIV/AIDS	85	85	92	81.67	85	81	76.67
	Thyroid	80	85	98.34	80	88.75	75	85

Based on the result that we have obtained in all the related experiments, we have compared the performance of binary and multi-class classification.

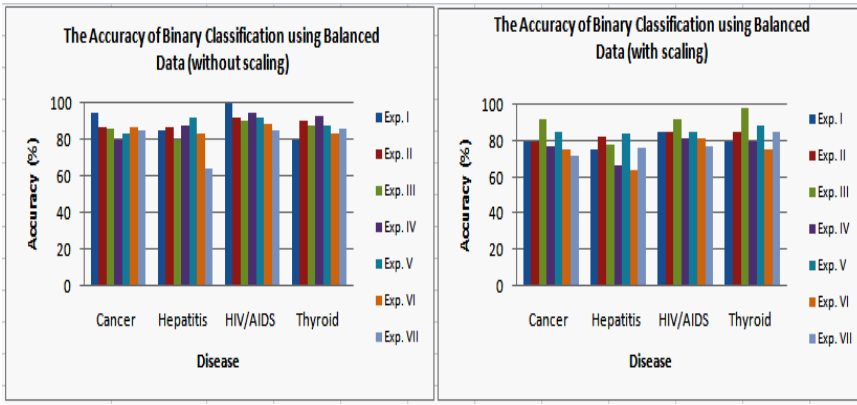

Figure 1 The accuracy of binary classification

As shown in Table 3 and Figure 1, we observed that the dataset (without scaling) outperforms the dataset (with scaling) in most of the experiments conducted, on average 80.89% (with scaling) and 86.92% (without scaling). In Dollah and Aono (2008), the binary classification accuracy of HIV/AIDS dataset shows the best among all four categories without scaling dataset, while cancer dataset shows the second best. This might be caused by the number of content-bearing keywords in cancer and HIV/AIDS

training data was higher than two other diseases in training data to build effective vector space models.

Table 4 A result of multiclass classification experiments using balanced data

Dataset	Experiment	I	II	III	IV	V	VI	VII
Dataset (without scaling)	Cancer + Hepatitis + HIV/AIDS + Thyroid	92.5	83.8	80.0	85.0	87.0	83.5	85.0
Dataset (with scaling)	Cancer + Hepatitis + HIV/AIDS + Thyroid	70.0	76.5	72.0	78.8	75.5	75.5	81.9

Table 4 show the performance of multi-class classification achieved in multi-class classification experiments using dataset (with scaling and without scaling) for balanced data.

Generally, the performance of multi-class classification (without scaling) in all the percentages of division between training and testing dataset, outperforms the dataset (with scaling), on the average 75.73% (with scaling) and 85.25% (without scaling). From the result of accuracy in all the experiments, we have observed that different percentage of training and testing data produce different performance of accuracy (Dollah and Aono 2008).

Experiments Using Unbalanced Data

For unbalanced data, we have conducted several experiments for binary and multi-class classification dataset. Each binary classification dataset, we have implemented two categories of experiments. In first category, we used 150 positive and 50 negative of paper abstracts (150P & 50N), while second category, we used 40 positive and 160 negative of paper abstracts (40P & 160N). However, for multi-class classification dataset, we have randomly selected 400 paper abstracts that consist of various total number of paper abstracts for four categories of diseases, which are 90 paper abstracts of cancer disease, 110 paper abstracts hepatitis of disease, 150 paper abstracts of HIV/AIDS disease and 50 paper abstracts of thyroid disease.

In all the experiments, we have used two types of dataset, which are (with scaling) dataset and (without scaling) dataset. The results for all experiments of binary and multi-class classification using unbalanced data are listed in Table 5, 6 and 7. Generally, Table 5, 6 & 7 and Figure 2 & 3 show different accuracy of binary and multi–class classification for each dataset (with scaling and without scaling).

Table 5 A result of binary classification experiments using unbalanced data (150P & 50N)

Dataset	Experiment	I	II	III	IV	V	VI	VII
Dataset (without scaling)	Cancer	95	90	80	78.3	83.8	82	78.3
	Hepatitis	75	85	82	80	83.8	79	78.3
	HIV/AIDS	95	80	90	83.3	87.5	78	85
	Thyroid	85	80	82	83.3	86.3	80	85
Dataset (with scaling)	Cancer	85	80	72	76.7	80	72	78.3
	Hepatitis	85	82.5	82	81.7	78.8	83	81.7
	HIV/AIDS	90	90	90	88.3	87.5	74	85
	Thyroid	70	85	86	83.3	85	79	81.7

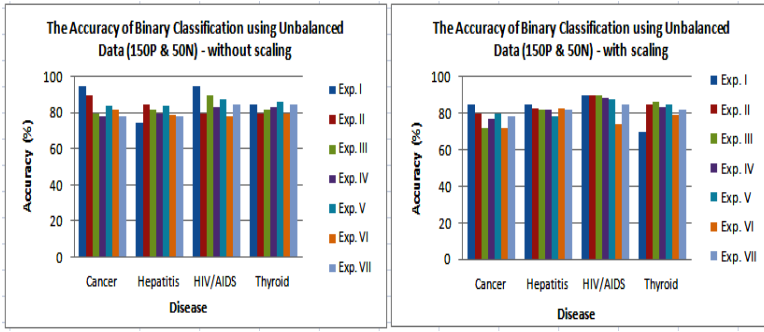

Figure 2 The accuracy of binary classification (unbalanced data –150P & 50N)

Table 5 and 6 describe different performance of binary classification achieved in both categories of dataset whether (150P & 50N) or (40P & 160N).

Table 6 A result of binary classification experiments using unbalanced data (40P & 160N)

Dataset	Experiment	I	II	III	IV	V	VI	VII
Dataset (without scaling)	Cancer	80	90	86	85	80	83	80
	Hepatitis	95	90	92	91.7	87.5	87	85
	HIV/AIDS	95	100	96	93.3	96.3	94	93.3
	Thyroid	95	92.5	84	86.7	81.3	83	80
Dataset (with scaling)	Cancer	75	80	72	80	80	80	80
	Hepatitis	95	90	84	90	86.3	87	81.7
	HIV/AIDS	90	90	88	90	93.8	90	89.2
	Thyroid	90	92.5	80	81.7	83.8	84	78.3

From the result, we can determine that the accuracy of binary classification using (150P & 50N) dataset shows almost

equal in all the experiments (with scaling and without scaling) conducted, on the average 81.91% (with scaling) and 83.25% (without scaling). Meanwhile, the performance of binary classification using (40P & 160N) dataset shows better accuracy, on the average 85.08% (with scaling) and 88.66% (without scaling) dataset in most of experiments (without scaling dataset) for all diseases.

Other than that, for dataset (without scaling), we observed that the accuracy of binary classification using (40P & 160N) dataset shows better performance compared to the accuracy of binary classification using (150P & 50N) dataset in almost all experiments. And for dataset (with scaling), we found the result of binary classification (using 40P & 160N) dataset shows better accuracy in most of experiments conducted. Table 6 are the result of accuracy for multi-class classification experiments based on different percentage of training and testing using unbalanced data.

Table 7 A result of multiclass classification experiments using unbalanced data

Dataset	Experiment	I	II	III	IV	V	VI	VII
Dataset (without scaling)	Cancer + Hepatitis + HIV/AIDS + Thyroid	75.0	72.5	67.0	70.0	71.9	68.5	67.5
Dataset (with scaling)	Cancer + Hepatitis + HIV/AIDS + Thyroid	62.5	67.5	66.0	62.5	66.9	62.5	65.4

Table 7 shows the performance of multi-class classification experiment for both of datasets (with scaling and without scaling)

using unbalanced data. Generally, we found that the accuracy of multi-class classification (without scaling) in all the percentages of division between training and testing data outperforms the dataset (with scaling) in all experiments, on the average 64.42% (with scaling) and 70.34% (without scaling) dataset.

In addition, experimental results using unbalanced data revealed that the performance of binary and multi-class classification using dataset (without scaling) is better than that using dataset (with scaling) for both of balanced data and unbalanced data. However, in general, it is very difficult to distinguish which category of percentages of training and testing data can produce the best result for binary and multi-class classification.

Overall, from the experiments that have been done, we observe that the choice of percentage for training and testing dataset has little influence on the classification performance, irrespective of (with scaling or without scaling) and (balanced or unbalanced data) dataset. Other than that, the performance of binary classification shows almost equal in all the experiments for both (with scaling or without scaling) and (balanced or unbalanced) dataset. However, the performance of multi-class classification using dataset (without scaling) outperforms the dataset (with scaling), while the average percentage of correct classification with unbalanced data is about 15% lower than with balanced data.

CONCLUSION

In this chapter, we attempt to study the impact of (with scaling or without scaling) dataset and (balanced or unbalanced dataset) in binary classification and multi-class classification for our

biomedical text data with four categories of diseases. For our experiments, we have employed LIBSVM to classify our datasets. Experimental results demonstrate that LIBSVM performed well in the binary classification and multi-class classification for both dataset (with scaling and without scaling) using balanced and unbalanced data.

Even though, we conducted several experiments using different percentage of training and testing data, in most of the experiments done, it produced 60% and more accuracy in classification. In addition, we also observe that the choice of percentage for training and testing dataset has little influence on the classification performance in all (with scaling or without scaling) experiments using both dataset (balanced or unbalanced data).

REFERENCES

Chang, C.-C. and Lim, C.-J. 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cohen, A.M and Hersh, W.R. 2005. A Survey of Current Work in Biomedical Text Mining. *Briefings in Bioinformatics*, **6**(1): 57-71.

Couto, F.M., Martins, B. and Silva, M.J. 2004. Classifying Biological Articles using Web Resources. *Proceeding of the 2004 ACM Symposium on Applied Computing*, pp.111-115.

Dollah, R. and Aono, M. 2008. Classifying Biomedical Text Abstracts using Binary and Multi-class Support Vector Machine, *The 22nd Annual Conference of the Japanese Society for Artificial Intelligence*, Hokkaido.

Ho, C. Y. and Lam, W. 1998. Automatic discovery of document classification knowledge from text databases. available at http://citeseer.ist.psu.edu/310941.html.

Leonard, J.E. Colombe, J.B. and Joshua, L.L. 2002. Finding Relevant References to Genes and Proteins in Medline using a Bayesian Approach. *Bioinformatics*, **18**(11): 1515-1522.

Lewis, D.D. and Ringuette, M.A. 1994. Comparison of Two Learning Algorithms for Text Categorization. *Proceeding of 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp. 81-93.

Mahinovs, A. and Tiwari, A. 2005. Text Classification Method Review. *Decision Engineering Report Series*, Canfield University, UK.

Remeikis, N. Skucas, I. and Melninkaite, V. 2004. Hybrid Machine Learning Approach for Text Categorization. *International Journal of Computational Intelligence*, **1**(1): 63-67.

Soucy, P. and Mineau, G.W. 2005. Beyond TFIDF Weighting for Text Categorization in the Vector Space Model,

International Joint Conferences on Artificial Intelligence, Scotland, pp. 1130-1135.

Wiener, E.D., Pedersen, J.O. and Weigend, A.S. 1995. A Neural Network Approach to Topic Spotting. *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp. 317-333.

3
PLAGIARISM DETECTION TECHNIQUES

Salha Mohammed Alzahrani
Naomie Salim

INTRODUCTION

Academic dishonesty is one of the critical measures to evaluate research papers, theses and students' assignments. Therefore, plagiarism detection is an area of concern for many researchers especially in the academic field. Other fields such as plagiarized news, magazine articles and web resources are also area of concern. In that regard, many detection techniques and tools have been developed to address the problem of plagiarism.

Different types of texts require different techniques to detect plagiarism. Documents to be retrieved, searched and thence judged according to the existence of plagiarism can be classified into two types: programming source code documents and natural language documents.

The first type of documents is programming source code. Several researches have been developed for source code plagiarism detection or so-called code clones detection (John et al. 1981; Sam 1981; Marguerite et al. 1988; Parker et al. 1989; Wise 1992; Edward 2001a, 2001b; Shauna 2001; Belkhouche et al. 2004; Kim and Choi 2005; Mike et al. 2005; Mozgovoy et al. 2005; Peter and Julian 2005; Seunghak and Iryoung 2005; Chao et al. 2006;

Christian and Tahaghoghi 2006; Samuel and Zelda 2006; Son et al. 2006; Jeong-Hoon et al. 2007; Lingxiao et al. 2007).

This type of documents has specific structure which is language dependent. The word "language" here refers to one of the programming languages such as FORTRAN, PASCAL, C, JAVA and many more. Thus, the detection algorithm is based on what programming language is used. Most of the early techniques were used for one programming language. For instance, John et al. (1981) developed plagiarism detection system for FORTRAN source code, Sam (1981) developed a tool that detect plagiarism in PASCAL programs and some other systems that can be found in the literature. In addition, there exist other techniques used to detect code clones in two or more programming languages. For example, Whale (1990) developed a system called Plague that works with Pascal and Prolog source code. Xin et al (2004) developed SID system (Shared Information Distance) which supports Java and C++ source code.

Early code clone detection techniques focus on keeping track of metrics such as number of lines, variables, statements, subprograms, call to subprograms and other parameters. However, current research makes a quantum leap and uses the structure or style of the source code. Thus, such technique is called stylometric (i.e. based on the style or structure) since some research has also been involved to use this technique in natural language plagiarism detection. The latest trends for code clone detection use artificial neural networks (Steve et al., 2007) in which neural networks were trained based on some common features of the submitted documents. The network input uses number of metrics as input unites. The network output with low error rate can measure how relevance two documents are. In brief, code clones detection techniques aim to locate plagiarized code in one or more programming language(s) and rely on either metrics or style/structure of the code.

The second type of documents is natural language documents written in English, Arabic or any other languages. Detecting plagiarism in this type of documents is much more

difficult than the first type because natural languages are not easy to be modeled. In contrast to code clone detection techniques, neither metrics nor structures can be maintained easily in natural language documents. Although the research of detecting plagiarism started more than a decade after the first type (1981 for code clones vs. 1997 for natural language documents), many applicable techniques and useful tools have been developed for plagiarism detection in natural language documents (Antonio et al. 1997; Culwin et al. 2001; Zaslavsky et al. 2001; Monostori et al. 2002; Bao et al. 2003; Bao et al. 2004; Daniel and Mike 2004; Weir et al. 2004; Xin et al. 2004; Ye et al. 2004; Heon et al. 2005; Hui and Jamie 2005; Stefan and Stuart 2005; Yerra and Ng 2005; Bao et al. 2006a; Bao et al. 2006b; Byung-Ryul et al. 2006; Eissen and Stein 2006; Hui and Jamie 2006; Kang et al. 2006; Koberstein and Ng 2006; Manuel et al. 2006; Sebastian and Thomas 2006; Sorokina et al. 2006; Benno et al. 2007; Liu et al. 2007; Meyer zu Eissen et al. 2007; Řehůřek 2007; Romans et al. 2007; Steve et al. 2007). The following sections discuss different representations of natural language documents for use in plagiarism detection.

DESCRIPTORS OF NATURAL LANGUAGE DOCUMENTS

There are several schemes to characterize documents before applying one of the plagiarism detection techniques. Simple document descriptors can be listed as follows:

- Character-based representation, the simplest form, in which documents are represented as a sequence of characters with ignoring spaces between words, periods (full stops) between statements and lines.
- Word-based representation, in which documents are represented as a collection of words with ignoring periods (full stops) between statements and lines.
- Phrase-based representation, in which a phrase (part of a statement) is used as a unit of comparison. For

example, 3-word phrase or so-called trigrams can be used as a comparison.

- Sentence-based representation, in which documents are segmented into statements using periods (full stop) as a statement-end indicator.
- Line-based representation, in which documents are characterized, and then compared, line-by-line. This representation is useful to compare operating system files.
- Paragraphed-based representation, in which documents are described as a collection of paragraphs or passages.
- Structure-based representation that characterizes documents as sections, subsections, subsubsections...etc. It is usually used in structured documents such as books, theses and academic papers. Structured-based documents are most likely to be stored as XML files for easier processing.
- Document-based representation, in which the whole document is treated as one It can be used with hash function, as we will discuss shortly, to detect duplicate documents.

In many cases, different descriptors can be combined to assist in plagiarism detection. For example, similar documents that discuss the same subject can be characterized as sections, each section can be characterized as paragraphs, and each paragraph can be described as a collection of statements and the plagiarism detection technique might work based on statement matching. Thus, plagiarism detection here goes from global detection (whole documents) to local detection (statements). In contrast, plagiarism detection can find similarities among statements (local detection), which may lead to similar paragraphs if all statements are similar and that possibly lead to similar documents (global detection).

Character-based and word-based representations are not worth themselves to detect plagiarism but they can be used with

more sophisticated representations such as hash-based, suffix-tree and fingerprints descriptors.

One sophisticated descriptor is hash-based which employs a hash function to transfer some kind of data that can be any of the simple representations: words, phrases, statements, paragraphs, lines or even the whole documents, into a relatively small integer called hash value, hash code, hash sum, or simply hash (Wikipedia, 2008).

Hash functions assist in detecting duplicated or similar statements, paragraphs or documents by comparing their hash values. For example, if the hash values of two documents are exactly the same, then documents are duplicated. The same thing with statements; i.e. hash values can be used to detect exact match among them. Moreover, hash values in specific ranges can guide to similar documents or statements. A typical hash function works as shown in Figure 1. The given example uses `hash_file` function in PHP to generate hash values using the algorithm `md5` and the contents of a given file.

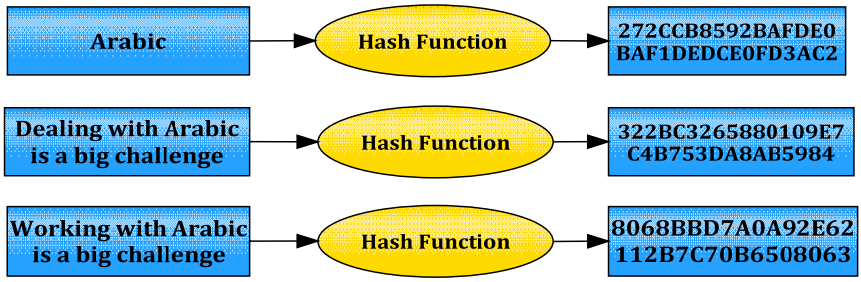

Figure 1 Typical Hash Function Example

Then, suffix-tree descriptor that presents the suffixes of a string as a tree whose edges are labeled with strings, and such that each suffix corresponds to exactly one path from the tree's root to a leaf (Wikipedia, 2008). Constructing such a tree for the string

consumes time and space especially for large-size documents. The word "string" here can be any one of the simple representations; i.e. words, phrases, statements, paragraphs or even documents. As a straightforward example, the string `BANANA` padded with `$` has six paths from the root to leaves correspond to the six suffixes `A$`, `NA$`, `ANA$`, `NANA$`, `ANANA$` and `BANANA$`, and suffix links drawn dashed as shown in Figure 2. However, once the suffix-tree constructed, several operations can be performed quickly such as utilizing string matching algorithms to find similarities.

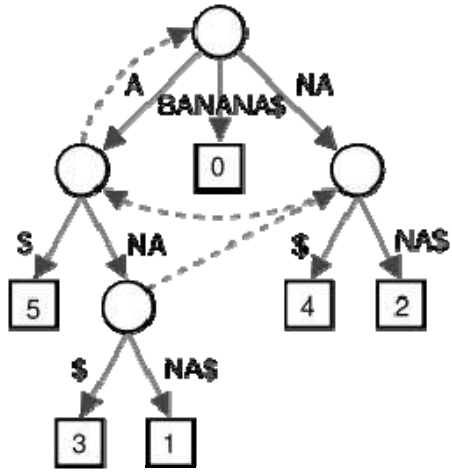

Figure 2 Typical suffix-tree example (from Wikipedia)

Another common descriptor in plagiarism detection applications is document fingerprint. Fingerprints can be defined as all possible substrings of certain length. The collection of the generated substrings is known as the fingerprint of the document and the process of generating fingerprints is called fingerprinting. Document fingerprint can identify the document uniquely as well

as human fingerprint does. Part of fingerprinting scheme will be used in this study and thus more details about fingerprinting scheme are provided in Section 3.1.

All in all, there are varieties of descriptors used to characterized natural language texts. It can be word-based, phrase-based, statement-based, paragraph-based, line-based, structure-based or document-based. Documents can be characterized by one or fusion of these descriptors to help in finding local or global plagiarism. Moreover, three additional representations can be used. First is hash-based representation in which hash values are extracted, and then compared, to detect duplicates or similarities based on the technique used, between whole documents, some parts, paragraphs, statements or phrases. Then is suffix-tree representation that can be built for words and thence phrases, statements, paragraphs, sections till the whole document. The process of building suffix-tree is expensive but once the tree is constructed, it can be used to detect duplicate or near duplicate documents. Lastly, fingerprint scheme which can be character-based, phrase-based or statement-based. It can be used to detect exact or similar paragraphs, sections and documents; based on the technique as well. Figure 3 summarizes our conclusion about different descriptors of natural language text.

In this chapter, we are focusing on statement-based representation when applying the proposed plagiarism detection techniques for several reasons. The first reason is because text is easy to be segmented into statements since Arabic, just as English, use periods (full stops) at the end of each statement. Besides, text segmentation into statements reduces the computations burden in comparison with other descriptors that require building suffix-tree or computing hash function. Lastly, but more significantly, statement-based representation is in the heart of the pyramid (Figure 3) and we can drill down or roll up to detect more or less plagiarized text.

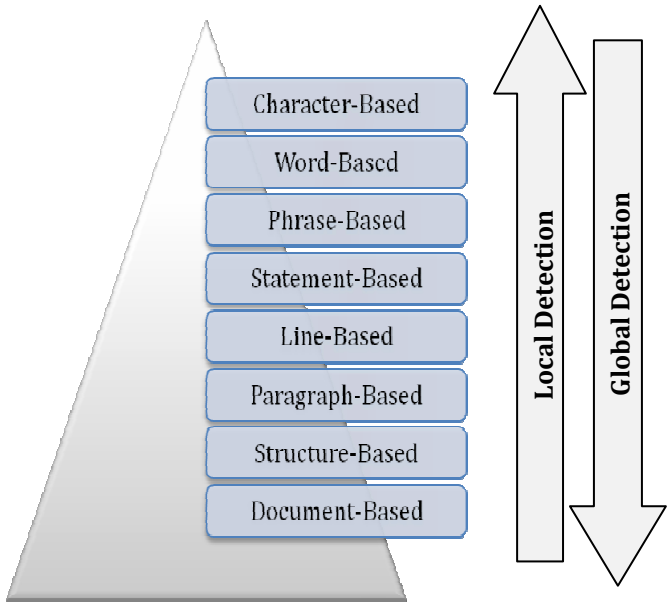

Figure 3 Different Document Representations

PLAGIARISM DETECTION TECHNIQUES IN NATURAL LANGUAGES

This research aims to detect plagiarism in natural language documents written in Arabic; therefore, this section discusses natural language plagiarism detection techniques in detail. Different document descriptors entail different techniques to be used for plagiarism detection. Several techniques have been developed or adapted for plagiarism detection in natural language documents. They can be classified into four main approaches. The first technique is **Fingerprint Matching** (Heintze, 1996; Lyon et al., 2001; Yerra and Ng, 2005) which involves the process of scanning and examining the fingerprints of two documents in order to detect plagiarism. depends on Then, **Clustering** (Antonio et al., 1997; Yerra and Ng, 2006) that uses specific words (or keywords)

to find similar clusters between documents. As an exhaustive type of clustering, **Fuzzy-Set IR** (Yerra and Ng, 2005) which used to detect plagiarism based on the fuzzy-set theory and degree of membership. Finally, structure-related technique or so-called **Stylometry Measurement and Comparison** (Stefan and Stuart, 2005; Byung-Ryul et al., 2006; Meyer zu Eissen et al., 2007) that focuses on the trend of structure that the overall document has. Even though this technique is mainly used for analyzing the source code of programs (See section 2.1, paragraph 5), some research has been involved to figure out the style that the author has in natural language documents. The following sections discuss each technique in detail.

Fingerprints Matching Technique

Fingerprinting techniques mostly rely on the use of K-grams (Manuel et al., 2006) because the process of fingerprinting divides the document into grams of certain length k. Then, the fingerprints of two documents can be compared in order to detect plagiarism. It have been observed through the literature that fingerprints matching approach differs based on what representation or comparison unit (i.e. grams) is used. It can, therefore, be classified further into three categories: character-based fingerprints, phrase-based fingerprints and statement-based fingerprints.

Character-Based Fingerprints Matching Technique

The early fingerprinting technique uses sequence of characters to form the fingerprint for the whole document. In this regard, Heintze (1996) divides fingerprinting techniques into two types: full and selective. In full fingerprinting, document fingerprint

consists of the set of all possible substrings of length K. As a simple example, if we have a document of length |D| = 5 consisting only one statement that has only one word "house", then we can see that "hous" and "ouse" are the all possible substrings of length K = 4. So, generally speaking, there are |D| – k + 1 such substrings, where |D| is the length of the document. Comparing two documents under this scheme is simply a matter of counting the number of substrings common in both fingerprints (Heintze, 1996).

Thus, if we compare a document A of size |A| against a document B, and if N is the number of substrings common in both then the resemblance measure R of how much of A is contained in B can be computed as follows:

$$R = \frac{N}{|A|}, 0 \leq R \leq 1$$

It is critical to choose the right value of k to provide good discrimination among documents. If the value of k is chosen appropriately, then full fingerprinting gives reliable exact match results. The value given by Heintze (1996) was effectively 30-45 characters. Although full fingerprinting is not practical for space-consuming and time-consuming reasons, it is a very useful measure for document copy detection.

On the other hand, selective fingerprinting aims to reduce the size of the fingerprint. Instead of using all possible substrings of length K, a subset of the substrings from the full fingerprint can be used. In this regard, there are two methods to determine the number of substrings: fixed-size or variable-size.

In fixed-size selective fingerprinting, a fixed number of substrings is used regardless of the size of the document. To design the document fingerprint using this scheme, two issues raise here: the fingerprint size and the selection strategy. For fingerprint size, Heintze employed different size of fingerprints; 100 characters for storage and 1000 characters since the search fingerprint for a document can be treated as a superset of the storage fingerprint. Heintze claimed that this choice made more reliable document matching. Besides, the selection strategy can be either random but it gives poor results, or using a string hash function which gives

better results. In the later strategy, the hash function is first employed and then a fingerprint of size k can be obtained by choosing k substrings with lowest hash values.

An alternative method is called variable-size selective fingerprinting that selects proportion (a part chosen in relation to the whole) of the substrings. Thus, the size of the fingerprint is proportional to the size of the document. Last of all, the computations burden in both types of selective fingerprinting is far less than full fingerprinting. The only drawback of this alternative is the space consumption for large documents.

Phrase-Based Fingerprints Matching Technique

Lyon et al. (2001) generates fingerprints using phrase-based mechanism to measure the resemblance between two documents. The first stage in the process is to convert each document to a set of trigrams (three words). Thus, a sentence like "*Dealing with Arabic Text is a big challenge*" will be converted to the set of trigrams {"*Dealing with Arabic", "with Arabic Text", "Arabic Text is", "Text is a", "is a big", "a big challenge*"}. Then, the set of trigrams for each document is compared with all the others using string matching algorithms. Finally, the measure of resemblance for each pair of documents is computed as follows:

$$R = \frac{S(A) \cap S(B)}{S(A) \cup S(B)}, 0 \leq R \leq 1$$

where S(A) and S(B) are the set of all trigrams in documents A and B, respectively. Usually, the results are presented in a ranked table with the most similar pairs at the top and if the documents are identical, the resemblance becomes one. Clearly, plagiarism detection using phrase-based fingerprinting works better and faster than character-based fingerprinting since it deals with words rather than letters. Nevertheless, this method is time-and-space-consuming for large-size documents.

Statement-Based Fingerprints Matching Technique

The pros and cons of character-based and phrase-based fingerprinting have led Yerra and Ng (2005) to represent the fingerprints of each statement (and so the whole document) by three least-frequent 4-grams. Although any value of K can be considered, yet K = 4 is an ideal choice (Yerra and Ng, 2005). This is because smaller values of K (i.e., K = 1, 2, or 3), do not provide good discrimination between sentences. On the other hand, the larger values of K (i.e., K = 5, 6, 7...etc), the better discrimination of words in one sentence from words in another. But each K-gram requires K bytes of storage and hence space-consuming becomes too large for larger values of K. So, we can conclude that K = 4 is an optimal or near optimal choice. Here is an explanation of how this 3-least frequent 4-grams works.

Firstly, a **4-gram** of a string is a set of all possible 4-character substrings. For example, let take a string S = *"Arabic Text"*, then the possible set of 4-grams include *"arab, rabi, abic, bict, icte, ctex, text"* with ignoring spaces.

Secondly, **three least-frequent** 4-grams are the best option to represent the sentence uniquely. To illustrate the three lest-frequent 4-gram construction process, consider the following sentence S *"Dealing with Arabic Text is a big challenge!"*. After applying stopwords removal and stemming algorithms (explained in Section 2.6), we obtain S_{new} *"deal arabic text challenge"*. The 4-grams are *deal*, *eala*, *alar*, *lara*, *arab*, etc. In this method, instead of comparing all possible 4-grams, only three 4-grams which have the least frequency over all 4-grams will be chosen. On gauge the frequency of each n-grams was stated by Damashek (1995). Let the document contain J distinct n-grams, with m_i occurrences of n-gram number i. Then the weight assigned to the i^{th} n-gram will be

$$x_i = \frac{m_i}{\sum_{j=1}^{J} (m_j)}$$

where

$$\sum_{i=1}^{i} (x_i) = 1$$

Thirdly, the three least-frequent 4-grams are concatenated to represent the fingerprint of a sentence from a document to be compared with the three least-frequent 4-gram representations of sentences in another document. Thus, in our example, if the three least-frequent 4-grams from S_{new} are *eala*, *alar*, *lara*, they will be concatenated to form the fingerprint F of S_{new} "*ealaalarlara"*.

Finally, two sentences are treated the same if their corresponding three least-frequent 4-gram representations are the same. A measure of resemblance for each pair of documents is computed as follows:

$$R = \frac{F(A) \cap F(B)}{F(A) \cup F(B)}, 0 \leq R \leq 1$$

where F(A) and F(B) are the common fingerprints in documents A and B, respectively.

The advantages of this method include faster processing time and less space consumption in comparison character-based and phrase-based fingerprinting. However, in the case of rewording and restructuring of statements, all techniques will fail in detecting that kind of plagiarism, a deficiency that can be recovered by fuzzy-set IR plagiarism detection approach explained in Section 3.3.

CLUSTERING TECHNIQUE

Clustering is concerned with grouping together objects that are similar to each other and dissimilar to the objects belonging to other clusters. Finding clusters of documents with related content is a significant process in plagiarism detection. According to

Bramer (2007) in the book "Principles of Data Mining", there are two main algorithms for clustering that use a measure of the distance between clusters: hierarchical clustering and k-means clustering, and they will be discussed in Section 3.2.3 and 3.2.4.

The Use of Clustering in Plagiarism Detection

Clustering is not worth by itself to judge plagiarism but it can be used as a first level of detection to find similar documents that discuss the same subject. It can be followed by another level of plagiarism detection to find plagiarized patterns using another technique such as fingerprints matching techniques. As an example, Antonio et al. (1997) developed a plagiarism detection system that uses clustering to find only documents that discuss same subject. Documents in the same cluster are compared until two similar paragraphs are found. Then, the two paragraphs are compared in detail, i.e. on a sentence-per-sentence basis to highlight plagiarism.

Distance Measure

An important step in any clustering is to select a distance measure, which will determine how the similarity of two documents (statements) is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and further away according to another. Common distance functions are discussed in Table 1.

Hierarchical Clustering

Hierarchical algorithms find successive clusters using previously established clusters. It can be agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters while divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

The basic agglomerative algorithm is for hierarchical clustering, more often used, can be summarized in the following steps (Bramer, 2007) and Figure 4 shows the basic process in agglomerative hierarchical clustering.

- First, assign each object to its own single-object cluster.
- Second, calculate the distance between each pair of clusters using one of the distance measures shown in Table 1.
- Third, choose the closest pair of clusters and merge them into a single cluster (so reducing the total number of clusters by one).
- Fourth, calculate the distance between the new cluster and each of the old clusters.
- Repeat third and fourth steps until all the objects are in a single cluster.

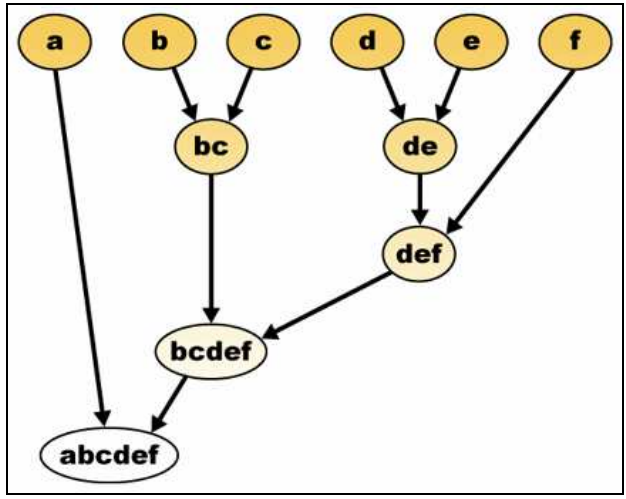

Figure 4 Agglomerative Hierarchical Clustering

Table 1 Different Distance Measures and their Mathematical Representations

Euclidean Distance	
The most commonly chosen type. Simply, it measures the geometric distance between two objects (documents).	$d(x,y) = \{\sum_i (x_i - y_i)^2\}^{1/2}$
Squared Euclidean Distance	
This measure is used to place progressively greater weight on objects that are further apart.	$d(x,y) = \sum_i (x_i - y_i)^2$
Manhattan Distance	
This distance is simply the average difference across dimensions. In most cases, this distance measure yields results similar to the simple Euclidean distance.	$d(x,y) = \sum_i \lvert x_i - y_i \rvert$
Chebychev Distance	
This distance measure may be appropriate in cases when one wants to define two objects as "different" if they are different on any one of the dimensions.	$d(x,y) = \text{Max}\lvert x_i - y_i \rvert$
Power Distance	
Sometimes one may want to increase or decrease the progressive weight that is placed on dimensions on which the respective objects are very different. * *r* and *p* are user-defined parameters. If *r* and *p* are equal to 2, then this distance is equal to the Euclidean distance.	$d(x,y) = (\sum_i \lvert x_i - y_i \rvert^p)^{1/r}$
Hamming Distance (also called Edit Distance)	
It measures the minimum number of substitutions required to change one member into another cluster. The hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different (Wikipedia, 2008).	HammingDistance ("toned", "roses") = 3.

K-Means Clustering

In k-means clustering, each object is assigned to precisely one of a set of clusters. First of all, we need to determine how many clusters we need, say k. The entire algorithm for k-means clustering is summarized in the following points (Bramer, 2007) and Figure 5 shows the basic process in agglomerative hierarchical clustering.

- First, choose the number of clusters *k*.
- Second, select *k* documents in an arbitrary fashion. Use these as the initial set of *k* centroids (centre of the cluster).
- Third, assign each of the objects to the cluster for which it is nearest to the centroid using one of the distance measures discussed in Table 1.
- Fourth, recalculate the centroids of the *k* clusters. The centroid is calculated as the mean distance in the cluster.
- Repeat third and fourth steps until the centroids no longer move.

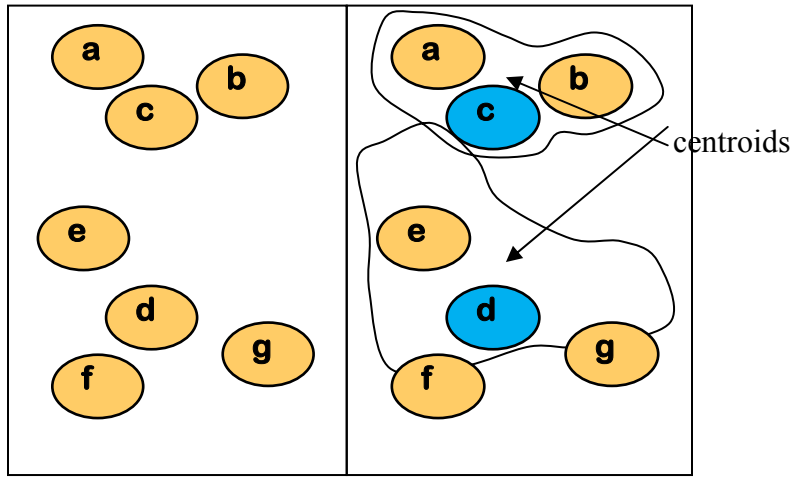

Figure 5 K-Means Clustering

Previous types of clustering are crisp; i.e. documents should belong to exactly one cluster. However, clustering can be fuzzy or vague. In other words, each document has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. In this study, we will use fuzzy-set IR model as an extended type of clustering in which a word might belong to more than one fuzzy-set with different degree of membership.

Fuzzy-Set Information Retrieval Plagiarism Detection Technique

A sentence can be treated as a group of words arranged in a particular order. In Arabic, likewise English, two sentences can be semantically the same but differ in structure (such as using the active versus passive voice), and matching two sentences is approximate or vague. This can be modeled, according to Yerra and Ng 2005, by considering that each word in a sentence is associated with a fuzzy-set that contains words with same meaning, and there is a degree of similarity between (words in) a sentence and the fuzzy-set. Fuzzy-set IR for plagiarism detection is effective since detect not only exact match but also similar statements based on the degree of similarity between words in the statement and the fuzzy-set.

The question now is how to construct the fuzzy-set and the degree of similarity between words. The answer is a *term-to-term correlation matrix* should be constructed before using the fuzzy-set IR. It consists of words and their corresponding *correlation factors* that measure the degrees of similarity (degree of membership between 0 and 1) among different words, such as "automobile عربة نقل" and "car سيارة" The fuzzy-set IR model obtains the degrees of similarity among sentences by computing the correlation factors between any pair of words from two different sentences in their respective documents. The *word-word correlation factor*, $c_{i,j}$,

defines the degree of similarity between any two words i and j in the term-term correlation matrix is calculated as follows:

$$c_{i,j} = n_{i,j}/(n_i + n_j - n_{i,j}) \quad (1)$$

where $c_{i,j}$ is the correlation factor between words i and j, $n_{i,j}$ is the number of documents in a collection with both words i and j, n_i (n_j , respectively) is the number of documents with word i (word j, respectively) in the collection.

Then, the degree of similarity of two sentences is the extent to which the sentences match. To obtain the degree of similarity between two sentences S_l and S_j, we first compute the *word-sentence correlation factor* $\mu_{i,j}$ of word i in S_l with all the words in S_j , which measures the degree of similarity between word i and (all the words in) S_j, as follows:

$$\mu_{i,j} = 1 - \prod_{k \in Sj}(1 - c_{i,k}) \quad (2)$$

where k is one of the words in S_j and $c_{i,k}$ is the correlation factor between words i and k as defined in (1).

Based on the μ-value of each word in a sentence Si, which is computed against sentence S_j , the *degree of similarity* of S_i with respect to S_j can be defined as follows:

$$Sim(S_i, S_j) = (\mu_{w1,j} + \mu_{w2,j} + \ldots + \mu_{wn,j}) / n \quad (3)$$

where w_k ($1 \leq k \leq n$) is a word in S_i, and n is the total number of words in S_i. $Sim(S_i, S_j)$ is a normalized value. Likewise, $Sim(S_j, S_i)$, which is the *degree of similarity* of S_j with respect to S_i, is defined accordingly.

Using (3) as defined above, two sentences S_i and S_j should be treated the same, i.e. equal (EQ), according to the following equation:

$$EQ(S_i, S_j) = \begin{cases} 1 \text{ if } MIN(Sim(S_i, S_j), Sim(S_i, S_j)) \geq p \text{ AND} \\ \quad |Sim(S_i, S_j) - Sim(S_i, S_j)| \leq v \end{cases} \quad (4)$$

0 otherwise

where p is called the *permission threshold value* and is set to 0.825 whereas v is called the *variation threshold value* and is set to 0.15 (Yerra and Ng, 2005). The permissible threshold is a value set to obtain the minimal similarity between any two sentences S_i and S_j in fuzzy-set IR plagiarism detection approach, which is used partially to determine whether S_i and S_j should be treated as equal (EQ). On the other hand, the variation threshold value is used to decrease the number of false positives (statements that are treated as equal but they are different) and false negatives (statements that are treated as different but they are equal).

From comparing fingerprints matching and fuzzy-set IR, it can be concluded that fingerprints matching can detect same patterns whilst fuzzy-set IR is useful in detecting not only same but also similar patterns. In sharp contrast to Stylometric comparison approaches, fuzzy-set IR procedure does not take into account the behavioral pattern of the plagiarist.

Stylometry Measurement and Comparison Technique

Stylometry, according to Stefan and Stuart (2005), is an approach for determining authorship of literature. It is based on the presumption that every author has a unique style of writing based on subconscious habits, such that authorship could be identified by analyzing a variety of stylistic characteristics which are inherent to an available text of sufficient length. One may assume that stylometry in insufficient to assist plagiarism detection not only in the cases of academic plagiarism but also in legal cases and criminal justice. Byung-Ryul et al. (2006) claimed that it is difficult to maintain a special structure for natural language documents as the program code. However, some efforts have been done to use the structure (or style) of natural language documents.

Stefan and Stuart (2005) proposed a tool support for plagiarism detection in text documents. The tool uses a stylometric language-sensitive approach. It takes two English text "under the hypothesis that they stem from different authors" and reports plagiarism if high stylistic similarity is found.

Meyer zu Eissen et al. (2007) defines the domain of stylometry plagiarism detection. They showed that it is possible to identify potentially plagiarized passages by analyzing a single document with respect to variations in writing style. Thus, plagiarism detection using stylometry does not require reference collection to be compared with the suspected document.

In addition, Byung-Ryul et al. (2006) developed an application of detecting plagiarism using Dynamic Incremental Comparison Method (DICM) which focuses on creating detectors and then use them for reporting plagiarism. DICM was build to avoid the weaknesses of line-by-line or word-by-word comparisons. Instead, it uses the structure or style as a comparison metric between various documents. The architecture of DICM has been built based on detectors of different sizes such as phrase, clause, sentence, document, etc collected by so-called a detector collector.

In conclusion, stylometry technique has not gained popularity in the majority of plagiarism detection tools because it is hard to maintain the style or structure of natural languages.

ESTABLISHED PLAGIARISM DETECTION TOOLS FOR NATURAL LANGUAGE DOCUMENTS

Several tools have been developed for plagiarism detection. They use variety of document descriptors that entail different techniques. Here is a brief exploration of eleven plagiarism detection tools: Diff, SCAM, SIF, COPS, KOALA, CHECK, MDR, PPChecker, SNITCH, WCopyFind, and Ferret.

Diff is a Unix/Linux Command (Yerra and Ng, 2005) that uses line-based representation for source code, text, and other line-

oriented files. It compares files line-by-line and captures the differences between two text documents one line at a time.

SIF, developed by Manber (1994), finds similar documents by using the fingerprinting scheme to characterize documents. However, it cannot measure the degree of overlap between two documents nor display the location of plagiarism. Moreover, if files containing the same information but using different sentence structures, they will be considered dissimilar.

SCAM (Stanford Copy Analysis Mechanism), developed by Shivakumar (1995), performs word-based copy detection, does not specify the plagiarism location and can handle only small documents.

COPS, developed by Brin (1995), uses hash-based scheme for copy detection. It compares hash values of given documents with that in the database for copy detection. COPS has several limitations reported by Yerra and Ng (2005). First, the use of hash function produces large number of collisions. Next, documents to be compared by COPS must have at least 10 sentences. Lastly, it has problems selecting correct sentence boundaries.

KOALA, designed by Heintze (1996), selects substrings of a document based on their usage and compares their fingerprints. This results increase the accuracy of KOALA in comparison to COPS.

CHECK is a structured-based plagiarism detection system developed by Antonio et al. (1997). It has some mechanism to determine the subject related to the document and then search domain is limited to only document with the same or relevant subjects. CHECK studies the semantics of the documents in addition to their syntax and is applied to only documents discuss same subject until two paragraphs which are highly related semantically are found. The paragraphs are then compared in detail, i.e., on a sentence-per-sentence basis, to determine plagiarized paragraphs.

MDR (Match Detect Reveal) system was developed by Zaslavsky et al. (2001) to detect plagiarism in documents. It uses suffix-tree representation to index the documents in a digital

library. **MDR** applies string-matching algorithms based on suffix trees to identify the overlap between a suspicious document and candidate documents. It is very powerful for finding exact copy. However, constructing suffix tree for documents is very expensive. Besides, this system is very weak at detecting modified documents.

PPChecker (Plagiarism Pattern Checker in Document Copy Detection) was developed by Kang et al. (2006). It uses statement-based representation for original documents and query document. The degree of similarity between two statements is calculated using "local-similarity-extractor" function proposed by the author. Then, "document-similarity-extractor" function is used to find the degree of overlap between two documents.

SNITCH (Spotting and Neutralizing Internet Theft by CHeaters) was developed by Sebastian and Thomas (2006) to detect copy and paste (exact match) plagiarism in paragraph-based representation. SNITCH implements a fast and accurate plagiarism detection algorithm using the Google Web API. It uses a sliding-window to scan documents and locate candidate passages that might be plagiarized. The sliding-window mechanism works as follows. First, SNITCH reads a window containing certain number of words. Then, it calculates the number of characters in each word. After that, the weight of the window is measured as the average of the number of characters per word and the words in the window. Next, the program stores the window's weight for use later. The process will be repeated for all such windows in the document by shifting the window forward in the document one word at a time. SNITCH, then, orders windows in decreasing order according to their weights, eliminates overlapping windows, and selects the top N weighted windows. Lastly, it searches the Internet for each, gathering the top search result (if any) for each. The output is an annotated HTML report containing the original document with hypertext links inserted for any passages that were found on the Internet.

WCopyFind developed by The University of Virginia (2006). It uses phrase-based representation with six or more words as a unit of comparing. It counts the number of words from

matching phrases and calculates plagiarism rate as a ratio of the number of matching words and the total number of words in the document. WCopyfind could find a partial overlap, but the user should set an adequate word number in a phrase.

Ferret (Lyon et al. 2001; Lyon et al. 2006) is a free standalone tool for detecting similar passages in large collections of students' coursework. It enables large numbers of documents to be analyzed quickly, and can also be used to identify plagiarism. The Ferret copy detector works on phrase-based mechanism to determine the similarity between two documents. Usually, the results are presented in a ranked table with the identical or most similar pairs at the top. Bao et al. (2006) used Ferret for copy detection in Chinese documents. Corpora of students' coursework from two Chinese universities were collected, and Ferret was applied to investigate the detection of plagiarism. Experiments showed that Ferret can find plagiarism in Chinese documents efficiently.

Plagiarism Detection Evaluation

It has been found through the literature that plagiarism detection can be evaluated in two levels. The first level is to determine the degree of overlap (also called degree of similarity or measure of resemblance) between two documents doc1 and doc2. In statement-based plagiarism detection, the degree of overlap can be calculated according to the following equation:

$$Overlap(doc1, doc2) = \frac{(|doc1| \cap |doc2|)}{(|doc1| \cup |doc2|)}$$

where $(|doc1| \cap |doc2|)$ is the number of common sentences in doc1 and doc2, and $|doc1| \cup |doc2|$ is number of all sentences in doc1 and doc2, respectively.

The second level is to evaluate the retrieval process of plagiarized documents regarding the whole corpus collection. For this purpose, precision and recall can be used. They can be defined in terms of a set of retrieved documents (i.e. the list of highly similar documents as determined by the degree of overlap equation) and a set of relevant documents (i.e. the list of all documents in the corpus collection that really have plagiarized text). The following equations describe how to calculate precision and recall values (Wikipedia, 2008).

$$Recall = \frac{|\{relevant\ docs\} \cap \{docs\ retrieved\}|}{|\{relevant\ docs\}|}$$

$$Precision = \frac{|\{relevant\ docs\} \cap \{docs\ retrieved\}|}{|\{docs\ retrieved\}|}$$

SUMMARY

To sum up, the literature review have been investigated in : (i) document descriptors in natural languages, (ii) plagiarism detection techniques, (iii) established plagiarism detection tools applied on English, (iv) and finally evaluation of plagiarism detection.

Document descriptors in natural languages are summarized in Figure 1. Finally, Figure 6 below shows a classification of plagiarism detection techniques as they have been illustrated in the literature.

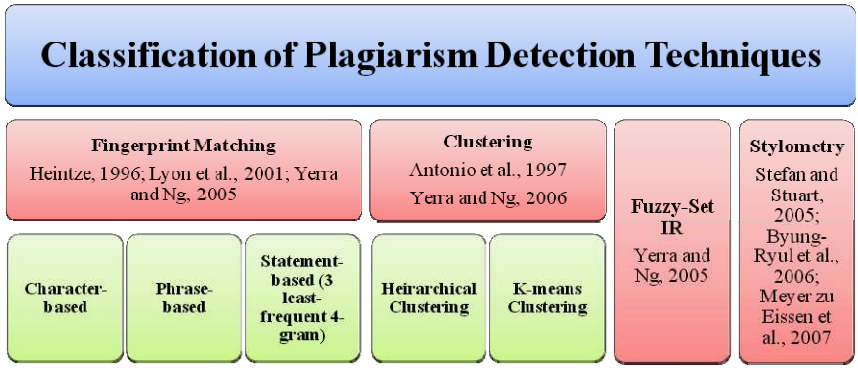

Figure 6 Classification of Plagiarism Detection Techniques

REFERENCES

Adeva, J. et al. (2006). *Applying Plagiarism Detection to Engineering Education Applying Plagiarism Detection to Engineering Education*. 7th International Conference on Information Technology Based Higher Education and Training. ITHET '06.

Antonio, S. et al. (1997). CHECK: a document plagiarism detection system. *Proceedings of the 1997 ACM symposium on Applied computing*. San Jose, California, United States, ACM.

Bao, J. et al. (2003). Document copy detection based on kernel method. *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*.

Bao, J. et al. (2004). *Semantic Sequence Kin: A Method of Document Copy Detection*. Advances in Knowledge Discovery and Data Mining: 529-538.

Bao, J. et al. (2006a). *A fast document copy detection model*. Soft Computing - A Fusion of Foundations, Methodologies and Applications, 10(1): 41-46.

Bao, J. et al. (2006b). *Copy detection in Chinese documents using Ferret*. Language Resources and Evaluation 40(3): 357-365.

Belkhouche, B. et al. (2004). Plagiarism detection in software designs. *Proceedings of the 42nd annual Southeast regional conference*. Huntsville, Alabama, ACM.

Benno, S. et al. (2007). Strategies for retrieving plagiarized documents. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. Amsterdam, The Netherlands, ACM.

Bramer, M. (2007). *Clustering. Principles of Data Mining*: 221-238.

Brin, S., Davis, J., Garcia-Molina, H. (1995). Copy Detection Mechanisms for Digital Documents. In *Proceedings of the ACM SIGMOD* (1995) 398–409.

Byung-Ryul, A. et al. (2006). *An Application of Detecting Plagiarism using Dynamic Incremental Comparison Method*. International Conference on Computational Intelligence and Security.

Campbell, D., Chen, W., Smith, R. (2000). Copy Detection Systems for Digital Documents. In *Proceedings of IEEE Advances in Digital Libraries*.

Chao, L. et al. (2006). GPLAG: detection of software plagiarism by program dependence graph analysis. *Proceedings of the 12th ACM SIGKDD international conference on*

Knowledge discovery and data mining. Philadelphia, PA, USA, ACM.

Christian, A. and Tahaghoghi, S. (2006). Plagiarism detection across programming languages. *Proceedings of the 29th Australasian Computer Science Conference* - Volume 48. Hobart, Australia, Australian Computer Society, Inc.

Culwin, F. et al. (2001). Visualising intra-corpal plagiarism Visualising intra-corpal plagiarism. *Proceedings of the Fifth International Conference on Information Visualisation*.

Dailey Paulson, L. (2002). *Professors use technology to fight plagiarism*. Computer 35(8): 24-25.

Damashek, M. *Gauging Similarity with N-grams: Language-Independent Categorization of Text*. Science 267 (1995) 843–848 570 R.

Daniel, R. W. and Mike, S. J. (2004). *Sentence-based natural language plagiarism detection*, ACM. 4: 2.

Dreher, H. (2007). Automatic Conceptual Analysis for Plagiarism Detection. Issues in Informing Science and Information Technology, Vol 4.

Edward, L. J. (2001a). *Metrics based plagiarism monitoring*, Consortium for Computing Sciences in Colleges. 16: 253-261.

Edward, L. J. (2001b). Plagiarism monitoring and detection - towards an open discussion. *Proceedings of the seventh annual consortium for computing in small colleges central plains conference on The journal of computing in small colleges*. Branson, Missouri, United States, Consortium for Computing Sciences in Colleges.

Eissen, S. and Stein, B. (2006). *Intrinsic Plagiarism Detection*. Advances in Information Retrieval: 565-569.

Hanakawa, N. et al. (2006). *A case study of an empirical approach to component requirements in developing a plagiarism detection tool*. 13th Asia Pacific Software Engineering Conference, APSEC 2006.

Heintze, N. (1996). Scalable document fingerprinting. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 191–200.

Helfman, J. (1994). *Similarity patterns in language*. IEEE Symposium on Visual Languages.

Heon, K. et al. (2005). An application of DICOM architecture for detecting plagiarism in natural language. *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design*.

Hui, Y. and Jamie, C. (2005). Near-duplicate detection for eRulemaking. *Proceedings of the 2005 national conference on Digital government research*. Atlanta, Georgia, Digital Government Research Center.

Hui, Y. and Jamie, C. (2006). Near-duplicate detection by instance-level constrained clustering. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. Seattle, Washington, USA, ACM.

Jeong-Hoon, J. et al. (2007). A source code linearization technique for detecting plagiarized programs. *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*. Dundee, Scotland, ACM.

John, L. et al. (1981). *A plagiarism detection system*, ACM. 13: 21-25.

Kang, N. et al. (2006). *PPChecker: Plagiarism Pattern Checker in Document Copy Detection*. Text, Speech and Dialogue: 661-667.

Kienreich, W. et al. (2006). *Plagiarism Detection in Large Sets of Press Agency News Articles*. 17th International Conference on Database and Expert Systems Applications. DEXA '06.

Kim, Y. and Choi, J. (2005). *A Program Plagiarism Evaluation System*. Computational Science and Its Applications – ICCSA: 10-19.

Koberstein, J. and Ng, Y. (2006). *Using Word Clusters to Detect Similar Web Documents*. Knowledge Science, Engineering and Management: 215-228.

Lingxiao, J. et al. (2007). Context-based detection of clone-related bugs. *Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. Dubrovnik, Croatia, ACM.

Liu, Y. et al. (2007). *Extending Web Search for Online Plagiarism Detection*. IEEE International Conference on Information Reuse and Integration.

Lane, P. C. R., Lyon, C., & Malcolm, J. A. (2006). Demonstration of the Ferret plagiarism detector. In *Proceedings of the 2nd International Plagiarism Conference*.

Lucca, G.A.D., Penta, M.D., Fasolino, A.R. (2002). An Approach to Identify Duplicated Web Pages. In *Proceedings of COMPSAC* (2002) 481–486

Lyon, C., Barrett, R., & Malcolm, J. A. (2003). *Experiments in plagiarism detection*. Technical report 388. School of Computer Science, University of Hertfordshire.

Lyon, C., Malcolm, J. A., & Dickerson, R. G. (2001). Detecting short passages of similar text in large document collections. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Lyon, C., Barrett, R., & Malcolm, J. (2006). *Plagiarism Is Easy, But Also Easy To Detect*. Plagiary: Cross□Disciplinary Studies in Plagiarism, Fabrication, and Falsification, 1 (5): 1□10

Manuel, Z. et al. (2006). *Plagiarism Detection through Multilevel Text Comparison*. Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution.

Marguerite, K. et al. (1988). *Program plagiarism revisited: current issues and approaches*, ACM. 20: 224-224.

Meyer zu Eissen, S. et al. (2007). *Plagiarism Detection Without Reference Collections*. Advances in Data Analysis: 359-366.

Mike, J. et al. (2005). *The boss online submission and assessment system*, ACM, 5: 2.

Mohammed Salem. (2006). *Comparison and Fusion of Retrieval Schemes Based on Different Structures, Similarity Measures and Weighting Schemes*. Universiti Teknologi Malaysia: MCS Thesis.

Monostori, K. et al. (2002). *Comparison of Overlap Detection Techniques*. Computational Science - ICCS 2002: 51-60.

Mozgovoy, M. et al. (2005). *Fast Plagiarism Detection System. String Processing and Information Retrieval*: 267-270.

Neill, C. et al. (2004). *A Web-enabled plagiarism detection tool*. IT Professional 6(5): 19-23.

Ogawa, Y., Morita, T., Kobayashi, K. (1991). *A Fuzzy Document Retrieval System Using the Keyword Connection Matrix and a Learning Method*. Fuzzy Sets and Systems. Vol. 39 (1991) 163–179.

Parker, A. et al. (1989). *Transactions on Computer algorithms for plagiarism detection*. Education, IEEE. 32(2): 94-99.

Peter, V. and Julian, D. (2005). An anti-plagiarism editor for software development courses. *Proceedings of the 7th Australasian conference on Computing education* - Volume 42. Newcastle, New South Wales, Australia, Australian Computer Society, Inc.

Reisman, S. (2005). *Plagiarism or ignorance? you decide*. IT Professional 7(1): 7-8.

Roxas, R. et al. (2006). *Automatic Generation of Plagiarism Detection Among Student Programs*. 7th International Conference on Information Technology Based Higher Education and Training.

Řehůřek, R. (2007). *Text Segmentation Using Context Overlap*. Progress in Artificial Intelligence: 647-658.

Romans, L. et al. (2007). Computer-based plagiarism detection methods and tools: an overview. *Proceedings of the 2007 international conference on Computer systems and technologies*. Bulgaria, ACM.

Sam, G. (1981). *A tool that detects plagiarism in Pascal programs*, ACM. 13: 15-20.

Samuel, M. and Zelda, F. (2006). Similarity and originality in code: plagiarism and normal variation in student assignments. *Proceedings of the 8th Australian conference on Computing education*. Volume 52. Hobart, Australia, Australian Computer Society, Inc.

Sebastian, N. and Thomas, P. (2006). *SNITCH: a software tool for detecting cut and paste plagiarism*, ACM. 38: 51-55.

Seunghak, L. and Iryoung, J. (2005). *SDD: high performance code clone detection system for large scale source code*. Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications. San Diego, CA, USA, ACM.

Shauna, D. (2001). Using metrics to detect plagiarism (student paper). *Proceedings of the twelfth annual CCSC South Central conference on The journal of computing in small colleges*. Amarillo College, Amarillo, Texas, United States, Consortium for Computing Sciences in Colleges.

Shivakumar, N., Garcia-Molina, H. (1995). SCAM: *A Copy Detection Mechanism for Digital Documents*. D-Lib Magazine (1995) http://www.dlib.org.

Son, J. et al. (2006). *Program Plagiarism Detection Using Parse Tree Kernels*. PRICAI 2006: Trends in Artificial Intelligence: 1000-1004.

Sorokina, D. et al. (2006). *Plagiarism Detection in arXiv Plagiarism Detection in arXiv*. Sixth International Conference on Data Mining, 2006. ICDM '06.

Stefan, G. and Stuart, N. (2005). Tool support for plagiarism detection in text documents. *Proceedings of the 2005 ACM symposium on Applied computing*. Santa Fe, New Mexico, ACM.

Steve, E. et al. (2007). *Plagiarism detection using feature-based neural networks*, ACM. 39: 34-38.

Sorokina, D. et al. (2006). *Plagiarism Detection in arXiv*. Sixth International Conference on Data Mining, 2006. ICDM '06.

Taghva, K. et al. (2005). *Arabic stemming without a root dictionary*. International Conference on Information Technology: Coding and Computing, ITCC 2005.

Thomas, L. and Fintan, C. (2001). *Towards an error free plagiarism detection process*, ACM. 33: 57-60.

Tomlinson, S. (2002). *Hummingbird SearchServer at TREC 2001*. Retrieved August 23, 2008 from http://trec.nist.gov/pubs/trec10/papers/HumTREC2001.pdf.

Weir, G. et al. (2004). *Work in progress - technology in plagiarism detection and management*. 34th Annual Frontiers in Education, FIE 2004.

Wise, M. (1992). *Detection of similarities in student programs: YAP'ing may be preferable to plague'ing*, ACM. 24: 268-271.

Xin, C. et al. (2004). *Shared information and program plagiarism detection*. Transactions on Information Theory, IEEE. 50(7): 1545-1551.

Ye, S. et al. (2004). *A Query-Dependent Duplicate Detection Approach for Large Scale Search Engines*. Advanced Web Technologies and Applications: 48-58.

Yerra, R. and Ng, Y. (2005). *A Sentence-Based Copy Detection Approach for Web Documents*. Fuzzy Systems and Knowledge Discovery: 557-570.

Zaslavsky, A. et al. (2001). *Using Copy-Detection and Text Comparison Algorithms for Cross-Referencing Multiple Editions of Literary Works*. Research and Advanced Technology for Digital Libraries: 103-114.

Cluster Analysis, StatSoft. Retrieved August 24, 2008 from http://www.statsoft.com/textbook/stcluan.html.

EVE2. (2008). Retrieved from http://www.canexus.com/eve/download.shtml.

Turnitin. (2008). Retrieved from http://www.turnitin.com.

My DropBox. (2008). Retrieved from http://www.mydropbox.com.

Scriptum. (2008). Retrieved from http://www.scriptum.ca.

WCopyFind. (2008). Retrieved from http://www.plagiarism.phys.virginia.edu. *Wikipedia*. (2008). Retrieved from http://www.wikipedia.org.

4
DIVERSITY BASED TEXT SUMMARIZATION

Mohammed Salem Binwahlan
Naomie Salim
Ladda Suanmali

INTRODUCTION

The automatic text summarization has gained high importance as an active research field in the recent years. The benefits of automatic text summarization system's availability increase the need for existence of such systems; the most important benefit of using a summary is its reduced reading time and providing quick guide to the interesting information.

Diversity, which refers to distinct ideas included in the document, became a very important factor in automatic text summarization to control the redundancy in the summarized text. Many approaches have been proposed for text summarization based on the diversity. For example, MMR (maximal marginal relevance) (Carbonell and Goldstein, 1998), maximizes marginal relevance in retrieval and summarization. The sentence with high

maximal relevance means it is highly relevant to the query and less similar to the already selected sentences. Our modified version of MMR maximizes the marginal importance and minimizes the relevance. This approach treats sentence with high maximal importance as one that has high importance in the document and less relevance to already selected sentences.

MMR has been modified by many researchers (Kraaij *et al.*, 2001; Mori *et al.*, 2005; Liu *et al.*, 2006; Zajic *et al.*, 2006; Filippova *et al.*, 2007; Ye *et al.*, 2005; Lin *et al.*, 2007). Our modification for MMR formula is similar to Mori *et al.*'s modification (2005) and Liu *et al.*'s modification (2006) where the importance of the sentence and the sentence relevance are added to the MMR formulation. Ribeiro and Matos (2007) proved that the summary generated by MMR method is closed to the human summary, motivating us to choose MMR and modify it by including some documents features. The proposed approach uses a binary tree to exploit the diversity among the document sentences. Neto *et al.* (2002) presented a procedure for creating approximate structure for document sentences in the form of a binary tree, in our study, we build a binary tree for each cluster of document sentences, where the document sentences are clustered using the K-means clustering algorithm into a number of clusters equal to the summary length. An objective of using the binary tree for diversity analysis is to optimize and minimize the text representation; this is achieved by selecting the most representative sentence of each sentences cluster. The redundant sentences are prevented from getting the chance to be candidate sentences for inclusion in the summary, serving as penalty for the most similar sentences. Our idea is similar to Zhu *et al.*'s idea (2007) in terms of improving the diversity where they used absorbing Markov chain walks.

The rest of this chapter is described as follows: section 2 presents the features used in this study, section 3 discusses the importance and relevance of the sentence, section 4 introduces the document-sentence tree building process, section 5 gives full description of the proposed method, section 6 discusses the

experimental design, section 7 presents the experimental results and section 8 concludes our work and draws the future study plan.

SENTENCE FEATURES

The proposed method makes use of eight different surface level features; these features are identified after the preprocessing of the original document is done, like stemming using porter's stemmer[1] and removing stop words. The features are as follows.

a. Word sentence score (WSS): it is calculated using the summation of terms weights (TF-ISF, calculated using eq. 1, (Neto *et al*., 2000)) of those terms synthesizing the sentence and occur in at least in a number of sentences equal to half summary length (LS) divided by highest term weights (TF-ISF) summation of a sentence in the document (HTFS) as shown in eq. 2. The idea of making the calculation of word sentence score under the condition of occurrence of its term in specific number of sentences is supported by two factors: excluding the unimportant terms and applying the mutual reinforcement principle (Zha, 2002). MAN`A-LO`PEZ *et al*. (2004) calculated the sentence score as proportion of the square of the query-word number of a cluster and the total number of words in that cluster.

[1] http://www.tartarus.org/martin/PorterStemmer/

Term frequency-inverse sentence frequency (TF-ISF) (Neto *et al*., 2000), term frequency is very important feature; its first use dates back to fifties (Luhn, 1958) and still used.

$$W_{ij} = tf_{ij} \times isf = tf(t_{ij}, s_i)\left[1 - \frac{\log(sf(t_{ij})+1)}{\log(n+1)}\right] \quad (1)$$

Where W_{ij} is the term weight (TF-ISF) of the term t_{ij} in the sentence S_i.

$$WSS(S_i) = 0.1 + \frac{\sum_{t_j \in S_i} W_{ij}}{HTFS} \quad | no.of\ sentences\ containing\ t_j >= \frac{1}{2} LS \quad (2)$$

Where 0.1 is minimum score the sentence gets in the case its terms are not important.

b. Key word feature: the top 10 words whose high TF-ISF (eq. 1) score are chosen as key words (Jaruskulchai and Kruengkrai, 2003; Kiani –B and Akbarzadeh –T, 2006). Based on this feature, any sentence in the document is scored by the number of key words it contains where the sentence receives 0.1 score for each key word.

c. N-friends feature: the n-friends feature measures the relevance degree between each pair of sentences by the number of sentences both are similar to. The friends of any sentence are selected based on the similarity degree and similarity threshold (Erkan and Radev, 2004).

$$N-friends(s_i,s_j)=\frac{s_i(friends)\bigcap s_j(friends)}{|s_i(friends)\bigcup s_j(friends)|}|i\neq j \quad (3)$$

d. N-grams feature: this feature determines the relevance degree between each pair of sentences based on the number of n-grams they share. The skipped bigrams (Lin, 2004b) used for this feature.

$$N-grams(s_i,s_j)=\frac{s_i(n-grams)\bigcap s_j(n-grams)}{|s_i(n-grams)\bigcup s_j(n-grams)|}|i\neq j \quad (4)$$

e. The similarity to first sentence (sim_fsd): This feature is to score the sentence based on its similarity to the first sentence in the document, where in news article, the first sentence in the article is very important sentence (Ganapathiraju, 2002). The similarity is calculated using eq. 11.

f. Sentence centrality (SC): the sentence has broad coverage of the sentence set (document) will get high score. Sentence centrality widely used as a feature (Erkan and Radev, 2004; Zajic, 2007). We calculate the sentence centrality based on three factors: the similarity, the shared friends and the shared n-grams between the sentence in hand and all other the document sentences, normalized by n-1, n is the number of sentences in the document.

$$SC(S_i)=\frac{\sum_{j=1}^{n-1}sim(S_i,d(S_j))+\sum_{j=1}^{n-1}n-friends(S_i,d(S_j))+\sum_{j=1}^{n-1}n-grams(S_i,d(S_j))}{n-1}|i\neq j\ and\ sim(S_i,d(S_j))>=\theta \quad (5)$$

Where $d(S_j)$ is a document sentence except S_i, n is the number of sentences θ in the document. is the similarity threshold which

is determined empirically, in an experiment was run to determine the best similarity threshold value, we have found that the similarity threshold can take two values, 0.03 and 0.16.

The following features are for those sentences containing n-grams (Villatoro-Tello *et al.*, 2006) (consecutive terms) of title where n=1 in the case of the title contains only one term, n=2 otherwise:

g. Title-help sentence (THS): the sentence containing n-gram terms of title.

$$THS(s_i) = \frac{s_i(n-grams) \bigcap T(n-grams)}{|s_i(n-grams) \bigcup T(n-grams)|} \quad (6)$$

h. Title-help sentence relevance sentence (THSRS): the sentence containing n-gram terms of any title-help sentence.

$$THSRS(s_j) = \frac{s_j(n-grams) \bigcap THS(s_i(n-grams))}{|s_j(n-grams) \bigcup THS(s_i(n-grams))|} \quad (7)$$

The sentence score based on THS and THSRS is calculated as average of those two features:

$$SS\_NG = \frac{THS(s_i) + THSRS(s_i)}{2} \quad (8)$$

THE SENTENCE IMPORTANCE (IMPR) AND SENTENCE RELEVANCE (REL)

The sentence importance is the main score in our study; it is calculated as linear combination of the document features. Liu *et al*.

(2006) computed the sentence importance also as linear combination of some different features.

$$IMPR(S_i) = avg(WSS(S_i) + SC(S_i) + SS\_NG(S_i) + sim\_fsd(S_i) + kwrd(S_i)) \quad (9)$$

Where WSS: word sentence score, SC: sentence centrality, SS_NG: average of THS and THSRS features, Sim_fsd: the similarity of the sentence s_i with the first document sentence and $kwrd(S_i)$ is the key word feature.

The sentence relevance between two sentences is calculated in (Liu *et al*., 2006) based on degree of the semantic relevance between their concepts, but in our study the sentence relevance between two sentences is calculated based on the shared friends, the shared n-grams and the similarity between those two sentences:

$$Rel(s_i, s_j) = avg(n-friends(s_i, s_j) + n-grams(s_i, s_j) + sim(s_i, s_j)) \quad (10)$$

DOCUMENT - SENTENCE TREE BUILDING (DST)

The first stage for building the document-sentence tree is to cluster the document sentences into a number of clusters. The clusters number is determined automatically by the summary length (number of sentences in the final summary). The initial centroids are selected as the following:

- Pick up one sentence which has higher number of similar sentences (sentence friends).
- Form a group for the picked up sentence and its friends, the maximum number of sentences in that group is equal to the total number of document sentences divided by the number of clusters.
- From the created group of sentences, the highest important sentence is selected as initial centroid.
- Remove the appearance of each sentence in the created group from the main group of document sentences.
- Repeat the same procedure until the number of initial centroids selected is equal to the number of clusters.

To calculate the sentence similarity between two sentences s_i and s_j, we use *TF-ISF* and *cosine* similarity measure as in eq. 11 (Erkan and Radev, 2004):

$$sim(s_i, s_j) = \frac{\sum_{w_i \in s_i, s_j} tf(w_i, s_i)\, tf(w_i, s_j) \left[1 - \frac{\log(sf(w_i)+1)}{\log(n+1)}\right]^2}{\sqrt{\sum_{w_i \in s_i} \left(tf(w_i, s_i) \left[1 - \frac{\log(sf(w_i)+1)}{\log(n+1)}\right]\right)^2} \times \sqrt{\sum_{w_i \in s_j} \left(tf(w_i, s_j) \left[1 - \frac{\log(sf(w_i)+1)}{\log(n+1)}\right]\right)^2}} \quad (11)$$

Where *tf* is term frequency of term w_i in the sentence $s_i or s_j$, *sf* is number of sentences containing the term w_i in the document, *n* is number of sentences in the document.

Each sentences cluster is represented as one binary tree or more. The first sentence which is presented in the binary tree is that sentence with higher number of friends (higher number of similar sentences), then the sentences which are most similar to already presented sentence are selected and presented in the same

binary tree. The sentences in the binary tree are ordered based on their scores. The score of the sentence in the binary tree building process is calculated based on the importance of the sentence and the number of its friends using eq. 12. The goal of incorporating the importance of sentence and number of its friends together to calculate its score is to balance between the importance and the centrality (a number of high important friends). $Score_{BT}(s_i)$

$$Score_{BT}(s_i) = impr(s_i) + (1 - (1 - impr(s_i) \times friendsNo(s_i))) \quad (12)$$

Where $Score_{BT}(s_i)$ is the score of the sentence s_i in the binary tree building process, $impr(s_i)$ is importance of the sentence and s_i is the number of $friendsNo(s_i)$ sentence friends.

Each level in the binary tree contains 2^{ln} of the higher score sentences, where *ln* is the level number, ln=0, 1, 2, ….., n, the top level contains one sentence which is a sentence with highest score. In case, there are sentences remaining in the same cluster, a new binary tree is built for them by the same procedure.

METHODOLOGY

The proposed method for summary generation depends on the extraction of the highest important sentences from the original text, we introduce a modified version of MMR, and we called it MMI (maximal marginal importance). MMR approach depends on the relevance of the document to the query, and it is for query based summary. In our modification we have tried to release this restriction by replacing the query relevance with sentence importance for presenting the MMI as generic summarization approach.

Most features used in this method are accumulated together to show the importance of the sentence, the reason for including the importance of the sentence in the method is to emphasize on the high information richness in the sentence as well as high information novelty. We use the tree for grouping the most similar sentences together in easy way, and we assume that the tree structure can take part in finding the diversity.

MMI is used to select one sentence from the binary tree of each sentence cluster to be included in the final summary. In the binary tree, a level penalty is imposed on each level of sentences which is 0.01 times the level number. The purpose of the level penalty is to reduce the noisy sentences score. The sentences which are in the lower levels are considered as noisy sentences because they are carrying low scores. Therefore the level penalty in the low levels is higher while it is low in the high levels. We assume that this kind of scoring will allow to the sentence with high importance and high centrality to get the chance to be a summary sentence. This idea is supported by the idea of PageRank used in Google (Brin and Page, 1998) where the citation (link) graph of the web page or backlinks to that page is used to determine the rank of that page. The summary sentence is selected from the binary tree by traversing all levels and applying MMI on each level sentence.

$$MMI(S_i) = Arg \max_{S_i \in CS \setminus SS} \left[(Score_{BT}(S_i) - \beta(S_i)) - \max_{S_j \in SS} (\mathrm{Re}l(S_i, S_j)) \right] \quad (13)$$

Where $\mathrm{Re}l(S_i, S_j)$ is the relevance between the two competitive sentences, S_i is the unselected sentence in the current binary tree, S_j is the already selected sentence, SS is the list of already selected sentences, CS is the competitive sentences of the current binary tree and β is the penalty level.

In MMR, the parameter λ is very important, it controls the similarity between already selected sentences and unselected sentences, and where setting it to incorrect value may cause creation of low quality summary. Our method pays more attention for the redundancy removing by applying MMI in the binary tree structure. The binary tree is used for grouping the most similar sentences in one cluster, so we didn't use the parameter λ because we just select one sentence from each binary tree and leave the other sentences.

Our method is intended to be used for single document summarization as well as multi-documents summarization, where it has the ability to get rid of the problem of some information stored in single document or multi-documents which inevitably overlap with each other, and can extract globally important information. In addition to that advantage of the proposed method, it maximizes the coverage of each sentence by taking into account the sentence relatedness to all other document sentences. The best sentence based on our method policy is the sentence that has higher importance in the document, higher relatedness to most document sentences and less similar to the sentences already selected as candidates for inclusion in the summary.

EXPERIMENTAL DESIGN

The Document Understanding Conference (DUC) data collection became as standard data set for testing any summarization method; it is used by most researchers in text summarization. We have used DUC 2002 data to evaluate our method for creating a generic 100-word summary, the task 1 in DUC 2001 and 2002, for that task, the training set comprised 30 sets of approximately 10 documents each, together with their 100-word human written summaries. The test set comprised 30 unseen documents. A part of this data is used in our experiment which is document set D061.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit (Lin, 2004b) is used for evaluating the proposed method, where ROUGE compares a system generated summary against a human generated summary to measure the quality. ROUGE is the main metric in the DUC text summarization evaluations. It has different variants, in our experiment, we use ROUGE-N (N=1 and 2) and ROUGE-L, the reason for selecting these measures is what reported by same study (Lin, 2004b) that those measures work well for single document summarization.

The ROUGE evaluation measure (version 1.5.5[2]) generates three scores for each summary: recall, precision and F-measure (weighted harmonic mean, eq. 14), in the literature, we found that the recall is the most important measure to be used for comparison purpose, so we will concentrate more on the recall in this evaluation.

$$F=\frac{1}{\left(alpha\times\left(\frac{1}{P}\right)+(1-alpha)\times\left(\frac{1}{R}\right)\right)} \tag{14}$$

Where P and R are precision and recall, respectively. Alpha is the parameter to balance between precision and recall; we set this parameter to 0.5.

[2] http://haydn.isi.edu/ROUGE/latest.html

EXPERIMENTAL RESULTS

The similarity threshold plays very important role in our study where the most score of any sentence depends on its relation with other document sentences. Therefore we must pay more attention to this factor by determining its appropriate value through a separate experiment, which was run for this purpose. The data set used in this experiment is document set d01a (one document set in DUC 2001 document sets). The document set d01a contains eleven documents; each document is accompanied with its model or human generated summary. We have experimented with 21 different similarity threshold values ranging from 0.01 to 0.2, 3 by stepping 0.01. We found that the best average recall score can be gotten using the similarity threshold value 0.16. However, this value doesn't do well with each document separately. Thus, we have examined each similarity threshold value with each document and found that the similarity threshold value that can perform well with all documents is 0.03. Therefore, we decided to run our summarization experiment using the similarity threshold 0.03.

We have run our summarization experiment using DUC 2002 document set D061 which contains two model or human generated summaries for each document. We gave the names H1 and H2 for those two model summaries. The human summary H2 is used as benchmark to measure the quality of our method summary, while the human summary H1 is used as reference summary. Beside the human with human benchmark (H2 against H1), we also use two more benchmarks, the baseline (outperformed all systems participated in DUC 2001 and DUC 2002 in creating 100 words summary, (Nenkova, 2005)) and the MS word summarizer. The baseline is the first 100 words from the beginning of the document as determine by DUC 2002.

The proposed method and the three benchmarks are used to create a summary for each document in the document set used in this study. Each system created good summary compared with the reference (human) summary. The results using the ROUGE variants (ROUGE-1, ROUGE-2 and ROUGE-L) demonstrate that our method performs better than the three benchmarks. Although the recall score is the main score used for comparing the text summarization methods when the summary length is limited[3], we found that our method outperforms all three benchmarks for all average ROUGE variants scores. The overall analysis for the results is concluded in Table-1 and the MMI average recall at the 95%-confidence interval is shown in Table-2:

Table-1: MMI, Baseline, MS Word Summarizer and H1-H2 comparison: Recall, Precision and F-measure using ROUGE-1, ROUGE-2 and ROUGE-L

	ROUGE-1			**ROUGE-2**			**ROUGE-L**		
Method	**Avg-R**	**Avg-P**	**Avg-F**	**Avg-R**	**Avg-P**	**Avg-F**	**Avg-R**	**Avg-P**	**Avg-F**
Baseline	0.44008	0.44979	0.44456	0.18023	0.18596	0.18291	0.41241	0.42149	0.41660
MS Word Summarizer	0.43681	0.52798	0.47356	0.21578	0.25889	0.23315	0.40328	0.48754	0.43729
H1-H2	0.47379	0.48641	0.47993	0.17955	0.18494	0.18218	0.44018	0.45230	0.44608
MMI	**0.53484**	**0.55043**	**0.54243**	**0.29655**	**0.30536**	**0.30085**	**0.50129**	**0.51574**	**0.50832**

[3] http://haydn.isi.edu/ROUGE/latest.html

Table-2: MMI average recall at the 95%-confidence interval.

Metric	95%-confidence interval
ROUGE-1	0.47519 - 0.60689
ROUGE-2	0.20742 - 0.39742
ROUGE-L	0.43929 - 0.57523

For ROUGE-1 average recall score, our method performance is better than the three benchmarks by: 0.06105, 0.09803 and 0.09476 for H1-H2, MS word summarizer and baseline respectively. For ROUGE-2 average recall score, our method performance is better than the three benchmarks by: 0.117, 0.08077 and 0.11632 for H1-H2, MS word summarizer and baseline respectively. For ROUGE-L average recall score, our method performance is better than the three benchmarks by: 0.06111, 0.09801 and 0.08888 for H1-H2, MS word summarizer and baseline respectively. The results obtained demonstrated that our proposed method - despite its simplicity where it doesn't make use of any deep natural language processing - is effective in creating extracts.

CONCLUSION AND FUTURE WORK

In this study, we have presented an effective diversity based method for single document summarization. Two ways were used for finding the diversity: the first one is as preliminary way where

the document sentences are clustered based on the similarity - similarity threshold is 0.03 determined empirically - and all resulted clusters are presented as a tree containing a binary tree for each group of similar sentences. The second way is to apply the proposed method on each branch in the tree to select one sentence as summary sentence. The advantages in our introduced method are: it doesn't use external resource except the original document given to be summarized and deep natural language processing is not required. Our method has shown good performance when comparing with the benchmark methods used in this study. For future work, our research is extending the proposed method for multi document summarization and using a large data set.

REFERENCES

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 24-28 August. Melbourne, Australia, 335-336.

Mori, T., Nozawa M. and Asada, Y. (2005). Multi-Answer-Focused Multi-document Summarization Using a Question-Answering Engine. *ACM Transactions on Asian Language Information Processing*. 4 (3), 305–320.

Liu, D., Wang, Y., Liu, C. and Wang, Z. (2006). Multiple Documents Summarization Based on Genetic Algorithm. In Wang L. et al. (Eds.) *Fuzzy Systems and Knowledge Discovery*. (355–364). Berlin Heidelberg: Springer-Verlag.

Zajic, D. M., Dorr, B. J., Schwartz, R. and Lin, J. (2006). Sentence Compression as a Component of a Multi-Document

Summarization System. *Proceedings of the 2006 Document Understanding Workshop*. 8-9 June. New York.

Filippova, K., Mieskes, M., Nastase, V., Ponzetto, S. P. and Strube, M. (2007). Cascaded Filtering for Topic-Driven Multi-Document Summarization. *Proceedings of the Document Understanding Conference*. 26-27 April. Rochester, N.Y., 30-35.

Ye, S., Qiu, L., Chua, T. and Kan, M. (2005). NUS at DUC 2005: Understanding documents via concept links. *Proceedings of Document Understanding Conference*. 9-10 October. Vancouver, Canada.

Lin, Z., Chua, T., Kan, M., Lee, W., Sun, Q. L. and Ye, S. (2007). NUS at DUC 2007: Using Evolutionary Models of Text. *Proceedings of Document Understanding Conference*. 26-27 April. Rochester, NY, USA.

Kraaij, W., Spitters, M., and Heijden, M. v. d. (2001). Combining a mixture language model and naive bayes for multi-document summarization. *Proceedings of Document Understanding Conference*. 13-14 September. New Orleans, LA, 109-116.

Zhu, X., Goldberg, A. B., Gael, J. V. and Andrzejewski, D. (2007). Improving diversity in ranking using absorbing random walks. *HLT/NAACL*. 22-27 April. Rochester, NY.

The Document Understanding Conference (DUC). http://duc.nist.gov.

Neto, J. L., Santos, A. D., Kaestner, C. A. A. and Freitas, A. A. (2000) Document Clustering and Text Summarization. *Proc. of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining*. April. London, 41-55.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*. 2(92), 159-165.

Neto, J. L., Freitas, A. A. and Kaestner, C. A. A. (2002). Automatic Text Summarization using a Machine Learning Approach. In Bittencourt, G. and Ramalho, G. (Eds.).

Proceedings of the 16th Brazilian Symposium on Artificial intelligence: Advances in Artificial intelligence. (pp. 386-396). London: Springer-Verlag.

MAN`A-LO`PEZ, M. J., BUENAGA, M. D. and GO´ MEZ-HIDALGO, J. M. (2004). Multi-document Summarization: An Added Value to Clustering in Interactive Retrieval. *ACM Transactions on Information Systems*. 22(2), 215–241.

Zajic, D. M. (2007). *Multiple Alternative Sentence Compressions As A Tool For Automatic Summarization Tasks*. PhD theses. University of Maryland.

Villatoro-Tello, E., Villaseñor-Pineda, L. and Montes-y-Gómez, M. (2006). Using Word Sequences for Text Summarization. In Sojka, P., Kopeček, I., Pala, K. (eds.). Text, Speech and Dialogue. vol. 4188, (pp. 293–300). Berlin Heidelberg: Springer-Verlag.

Ribeiro, R. and Matos, D. M. d. (2007). *Extractive Summarization of Broadcast News: Comparing Strategies for European Portuguese*. In sek, V. M. and Mautner, P. (Eds.). *Text, Speech and Dialogue*. (pp. 115–122). Berlin Heidelberg: Springer-Verlag.

Erkan, G. and Radev, D. R. (2004) LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22, 457-479. AI Access Foundation.

Zha, H. (2002). Generic summarization and key phrase extraction using mutual reinforcement principle and sentence clustering. *In proceedings of 25th ACM SIGIR*. 11-15 August. Tampere, Finland, 113-120.

Ganapathiraju, M. K. (2002, November 26). Relevance of Cluster size in MMR based Summarizer: A Report 11-742: Self-paced lab in Information Retrieval. *http://www.cs.cmu.edu/~madhavi/publications/Ganapathiraju_11-742Report.pdf*

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*. 30(1–7), 107–117.

Jaruskulchai, C. and Kruengkrai, C. (2003). Generic Text Summarization Using Local and Global Properties. *Proceedings of the IEEE/WIC international Conference on Web Intelligence*. 13-17 October. Halifax, Canada: IEEE Computer Society, 201-206.

Kiani –B, A. and Akbarzadeh –T, M. R. (2006). Automatic Text Summarization Using: Hybrid Fuzzy GA-GP. *IEEE International Conference on Fuzzy Systems*. 16-21 July. Vancouver, BC, Canada, 977 -983.

Lin, C. Y. (2004b). Rouge: A package for automatic evaluation of summaries. . Proceedings of the Workshop on Text Summarization Branches Out, 42nd Annual Meeting of the Association for Computational Linguistics. 25–26 July. Barcelona, Spain, 74-81.

Nenkova, A. (2005). Automatic text summarization of newswire: Lessons learned from the document understanding conference. *Proceedings of American Association for Artificial Intelligence*. 9–13 July. Pittsburgh, USA, 1436-1441.

5
COMPARING PREDICTION MODELS USING RICE YIELDS DATA

Ruhaidah Samsudin
Puteh Saad
Ani Shabri

INTRODUCTION

Prediction of crops yield like wheat, corn and rice has always been an interesting research area to agro meteorologist and it has become an important economic concern. Rice is the world's most important food crop and a primary source of food for more than half of the world's population (Khush, 2004). In Malaysia, the Third Agriculture Policy (1998-2010) was established to meet at least 70% of Malaysia's demand a 5% increase over the targeted 65%. The remaining 30% Raising comes from imported rice mainly from Thailand, Vietnam and China (Loh, 2004). The level of national rice self-sufficiency has become a strategic issue in the agricultural ministry of Malaysia. The ability to predict the future enables the farm managers to take the most appropriate decision in anticipation of that future.

Several prediction techniques, which mainly include multiple linear regression (MLR), autoregressive integrated

moving average (ARIMA) and artificial neural network (ANN), have been previously studied in the time series forecasting. These prediction methods have their own specialties.

MLR is very simple to understand and very easy to be accepted by average engineers. However, it may be a little rough because non-linear relationships always exist in actual situations. MLR is usually used to predict the development trend of happenings. MLR is a very powerful statistical tool that can be used as both an analytical and predictive technique in examining the contribution of potential new items to the overall estimate reliability (Skitmore & Patchell, 1990). Although it is not appropriate when describing non-linear relationships, which are multidimensional, consisting of a multiple input and output problem (Tam & Fang, 1999).

One of the most important and widely used time series models is ARIMA model. The popularity of the ARIMA model is due to its statistical properties as well as the well-known Box-Jenkins methodology in the model building process. For its rapid forecasting speed, ARIMA is usually applied in the short-term prediction (Zhang, 2003).

ANN are widely accepted as a technology offering an alternative way to tackle complex problems in actual situations. Much reported literature about ANN applications in the prediction time series that this technique is one of the most successful in forecasting areas. The advantages of ANN with respect to other models is their ability of modeling a multivariable problem given by the complex relationships between the variables by means of learning with training data.

The purpose of this study is to present a further comparison of ARIMA, ANN and MLR. We compare their advantages and disadvantages as well as their differences of performance in modeling a set of rice yields data. The comparison made in this study can provide reference for the choice of the three alternative approaches and their applications.

METHODOLOGY

ARIMA Model

The ARIMA method was developed by G.E.P. Box and G.M. Jenkins is one of the popular linear models in time series forecasting almost three decades ago (Box & Jenkins, 1976). The general non-seasonal model is known as ARIMA (p,d,q) can be written as the linear expression

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-1} + \sum_{j=1}^{q} \theta_j a_{t-j} + \varepsilon_j$$

where ϕ_t and θ_j are polynomials of AR and MA model , respectively; p order of nonseasonal auto regression; q order of the nonseasonal moving average and The ARIMA model consisted of three stages: identification, estimation and diagnostic checking. In the identification stage, when the observed time series presented trend and seasonality, differencing and data transformation is often applied to data to remove the trend and stabilize the variance to make the time series is stationary. The autocorrelation function (ACF) and partial autocorrelation function (PACF) of the stationary series were used to identify a tentative model. Once a model is identified, the parameters of the model are estimate and the Ljung–Box statistic was used to investigate model adequacy. If the model is not adequate, a new tentative model should be

identified, which is again followed by the steps of parameter estimation and model verification.

Multiple regression Model

The simplest form of regression is multiple linear regression (MLR). The MLR has been one of the most popular methods during the second half of the twentieth century of making predictive models and is known as a conventional method. The objective of regression analysis is to predict a single dependent variable from the knowledge of one or more independent variable. MLR has been widely used to model the cause-effect relationship between inputs and outputs and can be expressed as

$$Y = b_0 + b_1 X_1 + b_2 X_2 + ... + b_n X_n + \varepsilon \quad (1)$$

where Y is a dependent variable (i.e., output variable), $X_1,...,X_n$ are independent or explanatory variable (i.e., input variables), $b_1,...,b_n$ are regression parameters, ε is a random error, which is assumed to be normally distributed with zero mean and constant variance σ^2. The regression parameters $b_1,...,b_n$ are estimated using the least square method (LSM).

The Artificial Neural Network Model

ANN is widely applied in many fields such as corn and soybean yield, financial services, biomedical application, time series prediction, decision making and many others. Generally, neurons in an ANN are arranged in input, hidden and output layers and linked to others with associated weights and biases, which will be adjusted to optimal values during the training. The activation function is usually a sigmoid or hyperbolic tangent, which is a non-linear function. The back-propagation (BP) algorithm presented by Rumelhart and McClelland,was used to develop the ANN model used in this research (Rumelhart et al, Rumelhart & McClelland, 1986). The Neural Network Toolbox available in MATLAB is implemented in this study to design and train the BP. The ANN architecture used in this study is shown in Figure 1.

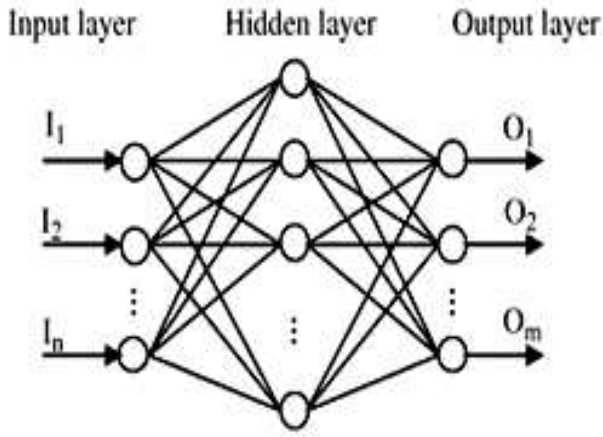

Figure 1 ANN architecture used in this study

One critical decision of ANN is to determine the appropriate architecture, that is, the number of inputs nodes, the number of hidden layers and the number of output nodes. The most common way in determining the number of hidden nodes is via experiments or by trial-and-error.

DESCRIPTION OF DATA

The data were collected from muda agricultural development authority (mada), kedah, malaysia ranging from 1995 to 2001. There are 4 areas with 27 locations. With two planting season for each year, total of 14 seasons is generated. There are 35 parameters that affect the rice yield. The parameters were classified to 5 groups. There are 3 types of weed; *rumpai*, *rusiga* and *daun lebar*, 3 types of pests; rats, type of worms and *bena perang*, 3 types of diseases; bacteria (*blb & bls*), *jalur daun merah* (*jdm*) *and hawar seludang*, one type of lodging and one type of wind paddy, making a total 11 input parameters. Out of 35 parameters, only 11 parameters are chosen since these are the most significant ones that were recommended by the domain expert from mada. The characteristics input parameter and output of rice yields prediction models was shown in table 1. Dataset used in this study is not the time series prediction because the weather in malaysia is not consistent. There are two types of season symptom that influenced the crop yield in malaysia. There are drought season and raining season.

Table 1 Characteristics input and output of rice yields

	description	name	min.	max.	mean	Standard deviation	Coefficient variation
INPUT	V1	rumpai,	0.1	2119	323	301.0	0.932
	V2	rusiga	0.1	1312	156	198.7	1.272
	V3	daun lebar	0.1	1858	99	169.7	1.719
	V4	wind paddy	0.1	423	13	39.4	2.976
	V5	bena perang	0.3	455	66	71.6	1.089
	V6	worns	0.1	1880	128	153.5	1.202
	V7	rats	0.1	280	61	53.7	0.881
	V8	bacteria	0.1	686	112	149.0	1.326
	V9	- jdm	0.1	98	11	22.3	2.080
	V10	hawar	0.1	322	21	38.7	1.805
	V11	rebah	0.1	610	17	55.8	3.389
OUTPUT	O	rice yields	51538.0	194179	101291	19192.6	0.189
	V11	rebah	0.1	610	17	55.8	3.389
OUTPUT	O	rice yields	51538.0	194179	101291	19192.6	0.189

.

PERFORMANCE EVALUATION CRITERIA

The performances of all the models developed in this study were evaluated using a wide variety of standard statistical performance evaluation measures. These criteria include the sum of square error (SSE), mean square error (MSE), mean absolute error (MAE), root mean squared error (RMSE), correlation coefficient, and so on. Among of them, RMSE, MAPE and R are the most widely used

performance evaluation criteria and will be used in this study. They are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(o_i - f_i)^2}$$

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{o_i - f_i}{o_i}\right| \times 100$$

$$R = \frac{\sum_{i=1}^{n}(o_i - \overline{o})(f_i - \bar{f})}{\sqrt{\sum_{i=1}^{n}(o_i - \overline{o})\sum_{i=1}^{n}(f_i - \bar{f})}}$$

where o_i and f_i are actual and fitted values respectively, and $\overline{o}$ and $\bar{f}$ are their average. The model with the smallest RMSE and MAPE, is considered to be the best.

The correlation coefficient is commonly used statistic and provides information on the strength of linear relationship between the actual and fitted values. A high R^2 value close to 1 indicates a good model fit with observed data. The value of R^2 describes the percentage of total variation explained by the model.

FORECAST PROCEDURE

The selection of ARIMA model

The plots in Fig. 1 indicate that the time series of rice yields are non-stationary in the mean and the variance. The transformed time series using the natural logarithm was taken, and then differencing was applied. The sample ACF and PACF for the transformed series are plotted in Fig. 2 and 3, respectively. The sample ACF of the transformed data revealed significant time lags at 1 and 27, while the PACF spikes at lag 1, 2 and 27. The sample ACF and PACF indicated that the rice yields have exhibited some pattern of seasonality in the series.

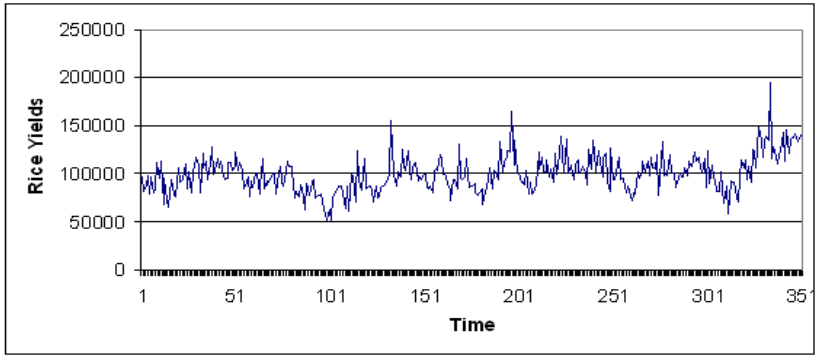

Figure 3 Rice Yields Series (1995-2001)

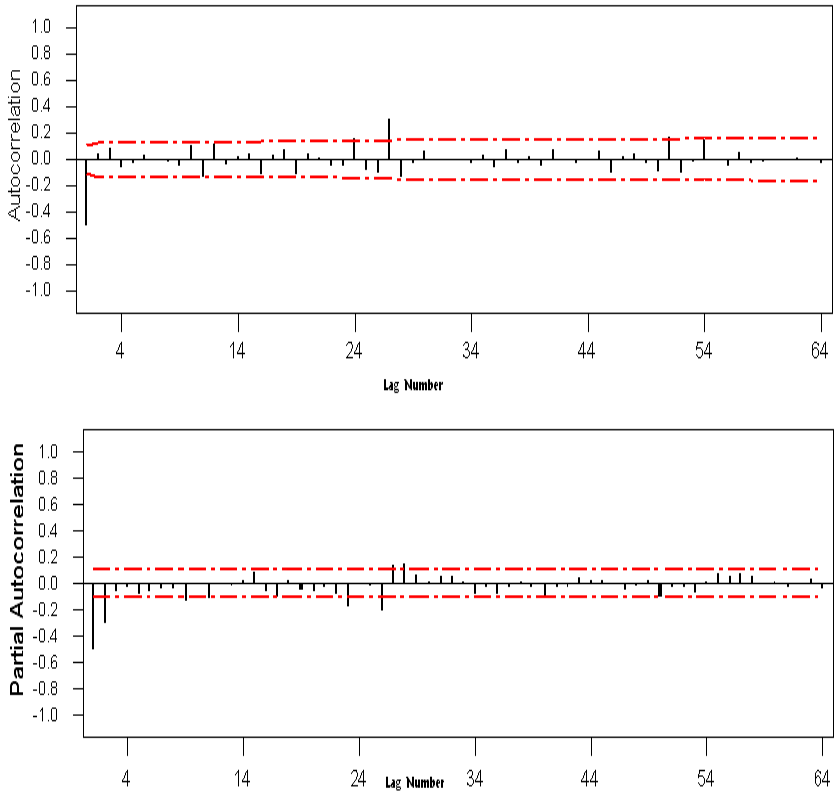

Figure 2 ACF and PACF for differenced series of natural logarithms

The plots suggest that ARIMA model is appropriate. Several models were identified and the statistical results during training are compared in the following Table 2. The criterions to judge for the best model based on MAPE, RMSE and R show that the ARIMA (0,1,1)X(1,0,1) is a relatively best model. This model has both non-seasonal and seasonal components.

Table 2 Comparisons of ARIMA models' statistical results

Criterion	**ARIMA Model**				
	(2,1,1)X(0,0,1)	(2,1,1)X(1,0,1)	(0,1,1)X(1,0,1)	(0,1,1)X(0,0,1)	(2,1,0)X(1,0,1)
MAPE (%)	0.010	0.0086	**0.0085**	0.0093	0.0088
RMSE	0.1494	0.1291	**0.1287**	0.1381	0.1317
R	0.6557	0.7328	**0.7344**	0.6873	0.7295

Fitting multiple linear regression

Table 3: Results of multiple regression analysis The MINITAB package were used develop the multiple linear regression model (MLR). The MLR models were developed and tested with the same data sets used for ANN development. The regression equations that were developed are referred to as trained models. These model were then validated with the same data sets used to test the ANN models, thus making the results comparable, and are referred to as validated models. The results of a multiple linear regression analysis data are given in Table 3.

```
The regression equation is
Rice Yields = 96585 + 5.01 v1 + 1.61 v2 - 2.76 v3 + 52.4 v4 + 60.4 v5 -
19.8 v6 + 32.6 v7 + 10.6 v8 - 317 v9 + 19.0 v10 - 19.9 v11

Predictor        Coef     SE Coef          T        P
Constant        96585        1902      50.79    0.000*
v1               5.011       4.342       1.15    0.249
v2               1.607       7.324       0.22    0.826
v3              -2.760       6.757      -0.41    0.683
v4               52.43       26.09       2.01    0.045*
v5               60.41       15.09       4.00    0.000*
v6             -19.799       6.918      -2.86    0.004*
v7               32.56       18.25       1.78    0.075
v8              10.598       9.659       1.10    0.273
v9             -317.42       64.16      -4.95    0.000*
v10              19.02       28.73       0.66    0.509
v11             -19.93       16.49      -1.21    0.228

S = 17113       R-Sq = 21.0%     R-Sq(adj) = 18.4%
```

In Table 3, the *p*-value in the analysis of variance section is less than 0.0001, showing that the regression is indeed significant, i.e. there does exist an influential relation between the dependent variable and the other independent variables selected here. Using the student's t-distribution, eleven of variables are significant at the $\alpha = 0.05$ level (indicated by an * in the last column). The most important variables in this regression are the v4, v5, v6 and v9.

The R^2 value is 21 percent showing that 21 percent of variability in the rice yields data is explained by the linear combination of the specific eleven independent variables.

Neural Network design and architecture selection

Before the training process begins of the ANN models, data normalization using the following formula is used

$$x_i = \frac{z_i}{z_{\max}}$$

where x_i are the normalized input or output values, z_i the original data, $z_{\max,}$ the maximum value.

One of the most important tasks in developing a successful time series forecasting model is the selection of the input variables, which determines the architecture of the model. In this study, three different types of neural networks models have been developed that differ in the types of input values.

Model I

The first type of ANN model (referred to as ANN1 model) performs a nonlinear functional mapping from the past observations $(X_{t-1}, X_{t-2}, ..., X_{t-p})$ to the future value X_t, i.e.

$$X_t = f(X_{t-1}, X_{t-2}, ..., X_{t-p}, w) + e_t$$

where w is a vector of all parameters. The ACF and PACF were used as a first step for the selection of useful variables. Figure 3 presents the ACF and PACF of data sets for the rice yields time series. Based on these analyses, the maximum number of lags, 27, was identified suitable to use as inputs for the proposed ANN. The ANN architecture of 27-H-1 was explored for capturing the

complex, non-linear and seasonality of rice yields data. Table 4 shows the performance of ANN during training with varying the number of neurons in the hidden layer (H).

Table 3 Performance Variation of a Three-Layer ANN with the number of neurons in the hidden layer for ANN1

Criterion	Number of neurons in the hidden layer											
	3	9	15	21	27	33	39	45	51	57	63	70
RMSE	16093	15733	15043	14263	14197	13794	13768	12863	13199	13334	**12590**	12791
MAE	0.118	0.129	0.115	0.114	0.114	0.113	0.106	0.099	0.102	0.103	**0.097**	0.101
R	0.618	0.599	0.644	0.668	0.672	0.708	0.700	0.744	0.729	0.725	**0.756**	0.744

It is observed that the performance of ANN is improved as the number of hidden neurons increases. However, too many neurons in the hidden layer may cause over-fitting problem, which results in the network can learn and memorize the data very well, but lacks the ability to generalize. If the number of neurons in hidden layer is not enough then the network may not be able to learn. So, an ANN with 63 neurons in the hidden layer seems to be appropriate.

Model II

The ANN model (referred to as ANN2 model) proposed in this section used eleven independent variables as in the multiple regression analysis. The formula created using ANN analysis is the following implicit expression (Shi et al., 2004):

$$O_t = ANN(v1, v2, ..., v11)$$

where ANN is a non-linear function which cannot be expressed with a usual mathematical formula, but is just designated as being a 'knowledge base'. Table 5 shows the performance of ANN2 during training varying with the number of neurons in the hidden layer. An ANN2 with 11- 22-1 gives the best prediction among several neural networks architecture trained.

Table 4 Performance Variation of a Three-Layer ANN2 with the number of neurons in the hidden layer

	Number of neurons in the hidden layer												
Criterion	2	4	6	8	10	12	14	16	18	20	22	24	26
RMSE	17472	17086	18310	16966	17019	16960	17333	16687	16562	16677	**16509**	17506	17356
MAE	0.139	0.135	0.159	0.140	0.136	0.132	0.145	0.134	0.135	**0.129**	0.134	0.129	0.148
Ra	0.620	0.656	0.641	0.680	0.662	0.668	0.666	0.688	0.697	0.697	**0.703**	0.697	0.690

COMPARISON OF PERFORMANCES

Forecasting of rice yields is very important for its prevention and control. In our study, we compared the performance of the best forecasting methods of ARIMA, MLR, ANN1, and ANN2 models. The performances of the models for both training and forecasting

data sets are summarized in Table 5. Figure 4 shows the comparison of the predicted rice yields of each of the six methods against the corresponding of the observed values for the 27 point-of-sample. As it can be seen from Table 6, the application of the ANN1 took on the smallest in term of RMSE and MAE, and the highest R in training. The ARIMA ranked second, ANN2 ranked third, followed by MRL. ANN2 model give the best performance in forecasting rice yields data. The results of ANN, with more hits of minimal errors, seem to give the best performance of the other models. ANN1 is more suitable and can be applied to prediction of rice yields data.

Table 5 Performances of four alternative models for rice yields

	Performance criterion	**ARIMA**	**MRL**	**ANN1**	**ANN2**
Testing	MAPE (%)	0.0989	0.1345	**0.0972***	0.1339
	RMSE	13138.80	16818.02	**12590.22***	16508.96
	R	0.7260	0.4584	**0.8697***	0.7028
Forecasting	MAPE (%)	0.1469	0.1268	0.1103	**0.1035***
	RMSE	20072.59	17812.95	16782.406	**16145.67***
	R	0.4228	**0.4923***	0.3526	0.4628

* The best fitting

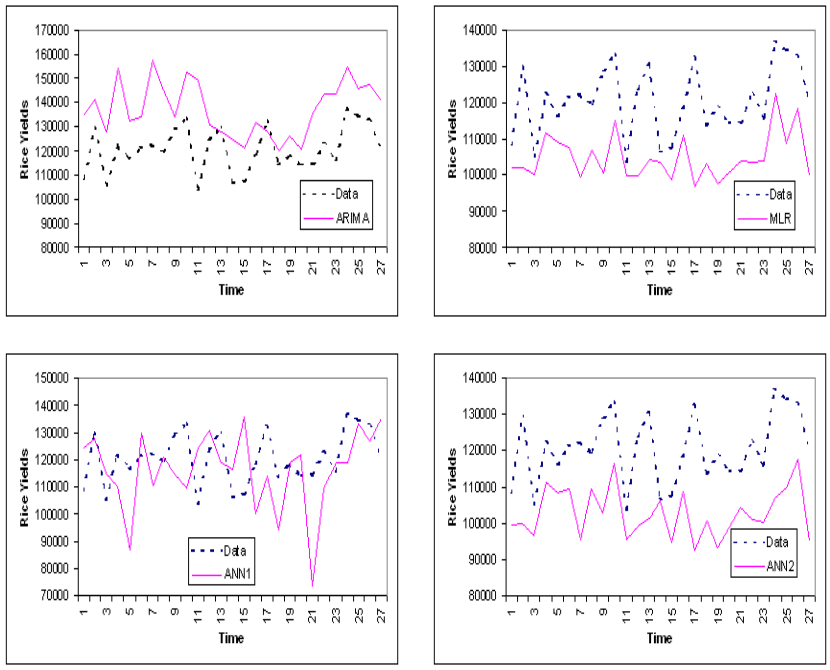

Figure 4 ARIMA, MLR, MLR1, ANN1, ANN2 and ANN3 prediction of rice yield

CONCLUSIONS

In this chapter, we conducted a comparative analysis of three alternative approaches for predicting rice yields, which are neural network, the Box-Jenkins approach and multiple regression analysis. In the regression analysis, $R = 0.458$ ($R^2 = 21\%$) showing

that 21 percent of variability in the rice yields data is explained by the linear combination of the specific eleven independent variables.

From the experimental results comparing the performance of four models, we can conclude that ANN1 is an effective method to model rice yields forecasting followed by ARIMA, ANN2 and MLR.

REFERENCES

Box, G.E.P. and Jenkins,G.M. (1976). Time Series Analysis: Forecasting and Control. San Francisco: Holden Day.

Khush, G.S., "Harnessing Science and Technology For Sustainable Rice Based Production Systems", Conf. on Food and Agricultural Organization Rice Conference, 2004.

Loh, F. F., "Future of Padi in The Balance", The Star May 21, 2001, retrieved March 21, 2004, from Koleksi Keratan Akhbar Perputakaan Sultanah Bahiyah UUM.

Rumelhart et al., Rumelhart, D.E., McClelland,J. 1986. Parallel Distributed Processing. MIT Press, Cambridge

Skitmore R.M. & Patchell, B.R.T. 1990. Developments in contract price forecasting and bidding techniques. Cost Modelling. London: E & FN Spon; 53-84.

Tam C.M. & Fang C.F. 1999. Comparative cost analysis of using high performance concrete in tall building construction by artificial neural networks. ACI Structural Journal 96(6): 927-936.

Zhang G.P. 2003. Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocoumpting 50:[6]9-175.

6
PREDICTION MODELS IN DENGUE DATA ANALYSIS

Nor Azura Husin
Naomie Salim

INTRODUCTION

This chapter contains a review of the dengue outbreaks in Malaysia, importance of dengue outbreak prediction and some explanation upon prediction methods and discussion about their strengths and weaknesses. It also includes critical discussion on the weaknesses and strengths of neural network and regression model approached and important terminologies.

DENGUE OUTBREAK IN MALAYSIA

Dengue is a mosquito-borne disease affecting at least 50 million people around the world annually (WHO, 2002). It continues to be a public health problem in Malaysia showing an upward trend from 27.5 cases/100,000 population in 1990 to a high of 123.4 cases/100,000 population in 1998 during the global pandemic, declining to 31.99 cases /100,000 population in the year 2000

based on notification of clinically-diagnosed cases (Teng and Sing, 2001).

Dengue fever has been endemic in Malaysia for a long time. As early as 1902, Skae recorded the disease in Penang (Rudnick, 1986). Malaysia had dengue haemorrhagic fever (DHF) first outbreak in Penang in 1962 and it was made a notifiable disease in 1971. The next outbreak occurred in 1973, starting in Kuala Lumpur and spreading to involve mostly the urban centre's. Dengue infection is predominant in urban areas where 61.8% of the country's population lives, as compared to rural area with only 34% in 1980. Rapid industrial and economic development over the last two decades has brought about massive infrastructure development and a very active construction sector for housing and commercial development, creating many manmade opportunities for *Aedes* mosquito breeding. This factor, coupled with rural-urban migration and pockets of illegal settlements, indiscriminate solid-waste disposal and a tropical rainfall, provide fertile grounds for *Aedes* breeding and the rise of dengue transmission in the country (Ministry of Health Malaysia, 1990-2000).

The case-fatality for DHF was especially high but this was partly contributed by under-reporting of DHF where the initial notification as dengue fever (DF) was not rectified when these cases subsequently were diagnosed as DHF. Thus, the denominator for DHF remains the old lower figure (Teng and Sing, 2001). In terms of the total number of cases, or yearly incidence, disease of dengue fever cause special problem to health services because of their epidemic potential, the often high case-fatality rate and unusual difficulties in their treatment and prevention.

IMPORTANCE OF PREDICTION

Recently, predictions on dengue outbreak become very important (Gubler, 2002). With prediction, government and health

departments may provide plans and arrange early intervention programs including campaigns to those susceptible groups of communities before an outbreak occurs. This will be possible only when knowledge about the relationship of past and future dengue outbreak, with temporal and location information, were discovered. Various factors such as dengue fever prevalence, population distribution and meteorological factors like rainfall are important in determining the mosquito survival and reproduction (Seng, Chong and Moore, 2005). Climate factors influence the transmission of dengue fever, the world's most widespread vector-borne virus. Based on this purpose, predictions can also be used to identify the possibility of future dengue outbreak. If climate change occurs, as many climatologists believe, this will increase the epidemic potential of dengue-carrying mosquitoes, given viral introduction and susceptible human populations. Increased incidence may first occur in regions bordering endemic zones in latitude or altitude. Endemic locations may be at higher risk from hemorrhagic dengue if transmission intensity increases (Patz, Willem, Dana and Theo, 1998).

PREDICTION MODEL

Regression

Regression analysis is any statistical method where the mean of one or more random variables is predicted conditioned on other (measured) random variables. Regression analysis is more than curve fitting (choosing a curve that best fits given data points). It involves fitting a model with both deterministic and stochastic components. The deterministic component is called the predictor and the stochastic component is called the error term. The simplest form of a regression model contains a dependent (y variable) and a

single independent variable (*x* variable). Regression is usually posed as an optimization problem as we are attempting to find a solution where the error is at a minimum. The most common error measure that is used is the least squares. In a certain sense, least square is an optimal estimator.

Linear Regression

Linear regression is the most common case in practice because it is the easiest to compute and gives good results. Indeed, by restraining the variations of the factors to a small enough domain, the response variable can be approximated localy by a linear function. When we do a linear regression, we are implicitly supposing that given a set of factors *x*1, *x*2…*x*n, the best approximation of the response variable *y* we can find is a linear combination of these factors *x1*, *x*2…*x*n. In simple linear regression the model function represents a straight line. The results of data fitting are subject to statistical analysis. Formally, the model for linear regression is written as:

$$yi = \alpha + \beta_{xi} + \varepsilon_i$$

where

yi represent the data,
α represent the intercept,
β represent slope as fitting model
εi represent an error.

Multiple Linear Regression

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the

independent variable x is associated with a value of the dependent variable y. The population regression line for n explanatory variables $x_1, x_2, ..., x_n$ is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 ... \beta_n x_n$. This line describes how the mean response μ_y changes with the explanatory variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation σ. The fitted values $b_0, b_1, ..., b_n$ estimate the parameters $\beta o, \beta 1 ... \beta n$ of the population regression line. Formally, the model for multiple linear regression, given n observations, is

$$y_i = \beta_0 + \beta_{1xi1} + \beta_{2xi2} + ... \beta_{pxip} + \varepsilon_{i\,for\,i = 1, 2, ...n.}$$

where

yi represent the data,

$\beta o + \beta 1xi1 + \beta 2xi2 + \ldots \beta pxip$ represent as fitting model and

εi represent an error.

Nonlinear Regression

Nonlinear regression is a method of finding a nonlinear model of the relationship between the dependent variable and a set of independent variables. Unlike linear regression, which is restricted to estimating linear models, nonlinear regression can estimate model with arbitrary relationship between independent and dependent variables. This is accomplished using iterative estimation algorithms. For the example, if the population be predicted based on time, a scatter plot shows that there seems to be a strong relationship between population and time, but the relationship is nonlinear, so it requires the special estimation

method of the nonlinear regression procedure. By setting up an appropriate equation, such as logistic population growth model, we can get a good estimate of the model, allowing us to make predictions about population for times that were not actually measured. Nonlinear regression is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables. The data are fitted by a method of successive approximations. The results of data fitting are subject to statistical analysis, but because of the nonlinear nature of the model, the statistics are biased. A nonlinear model is one in which the calculated value, f (x, β), is a nonlinear function of the parameters, β. The examples of standard nonlinear regression model are log-logistic, Gaussian, polynomial, log-linear, weibull, gompertz, density and other.

Neural Networks (NN)

Neural networks have been increasingly used as a model for time series forecasting other application. The high interest in NN comes from their ability to approximate complex nonlinear function. In the forecasting field, neural networks have shown their great ability in their learning process and predicting the desired output. A neural network is a powerful data-modeling tool that is able to capture and represent complex input/output relationships. Various NN models have been proposed since its conception in the 1940's, but the multi-layer perceptron (MLP) and the radial basis function (RBF) network are the most widely used (Cherkassky, Vladimir and Wechsler, 1993). Both models have been proven to be universal function approximation which means that given enough data, the underlying function can be approximated with accuracy (Yang, 2006 and Yao, 1999).

Multi-Layer Perceptron (MLP)

This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. The MLP neural networks learn using an algorithm called back propagation. With back propagation, the input data is repeatedly presented to the neural network. With each presentation, the output of the neural network is compared to the desired output and an error is computed. This error is then feedback (back propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as training. MLP network typically consists of three layers of neurons. An input layer for the predictor variables, a hidden layer, and output layer for the criterion variables (Figure 9.1). On the one hand, MLP networks possess an extremely high flexibility, which allows them to generate a wide variety of basic function shape suitable to the specific problem. On the other hand, the risk of convergence to local optima and saturation of neurons requires an extensive trial-and-error approach, which is particularly difficult and tedious when the problems become more complex and an understanding of the network is scarcely possible.

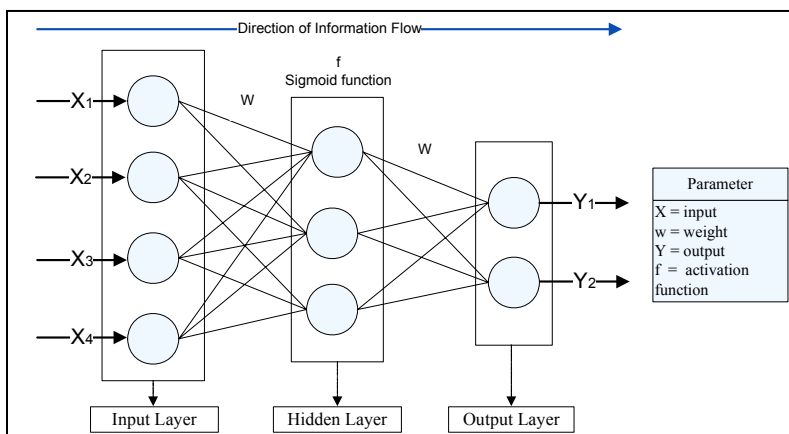

Figure 9.1 Basic architecture of Multilayer Perceptron.

The most important properties of MLP networks can be summarized as follows:

- Accuracy is typically very high. Owing to the optimization of the hidden layer weights, the MLP is extremely powerful and usually requires fewer neurons and fewer parameters than model architectures to achieve comparable approximation accuracy. This property can be interpreted as a high information compression capability, which is paid for by long training times and the other disadvantages caused by nonlinear parameters.
- Smoothness is very high. Owing to the tendency to monotonic interpolation behavior, the model output is typically very smooth.
- Sensitivity to noise is very low since, owing to the global character, almost all training data samples are exploited to estimate all model parameters.
- Parameter optimization generally has to be performed by a nonlinear optimization technique and thus is slow.
- Structure optimization requires computationally quite expensive pruning or growing methods.
- Online adaptation is slow and unreliable owing to the nonlinearity of the parameters and the global approximation characteristics.
- Training speed is very slow since nonlinear optimization techniques have to be applied and perhaps repeated for several

network weight initializations if an unsatisfactory local optimum has been reached.
- Evaluation speed is fast since the number of neurons is relatively small compared with other neural network architectures.
- Curse of dimensionality is very low because the ridge construction mechanism utilizes projections. The MLP is the neural network for very high dimensional problems.
- Interpretation is extremely limited since projections can be hardly interpreted by humans.
- Incorporation of constraints is difficult owing to the limited interpretation.
- Incorporation of prior knowledge is difficult owing to the limited interpretation.
- Usage is very high. MLP networks are still the standard neural networks.

MLP AS PREDICTION MODEL

Artificial NN were originally inspired by attempts to provide simple models of brain function and learning (McCulloch and Pitts, 1943). An NN is an approach to modeling the structure and function of the brain. It is an attempt to simulate with specialized hardware or software, the simple information processing capabilities of neurons connected in multiple layers. The idea behind developing artificial neural networks was to transfer the idea of parallel processing to the computer in order to take advantage of some of the brain's features (McCulloch and Pitts,

1943). Most types of neural networks can be covered by the definition a system of simple processing elements, neurons. Those are connected into a network by a set of (synaptic) weights. The architecture of the network, the magnitude of the weights and the processing element's mode of operation determine the function of the network (Minns and Balkema, 1998). In terms of the vocabulary used in the neural network analogy, there are few terms, which are carried over from the biological NN.

Architecture of NNM

A typical architecture of a NNM is somewhat straightforward. However, it useful in providing predictions for training inputs was limited until the advent of a self-correcting routine that automatically alters the weight to minimize errors was added. There are several aspects underlined in the architecture of NNM and these are input, output and hidden layer (Roselina Salleh@Sallehuddin, 1999).

Neuron

The neuron or node or unit as it is also called is a processing element that takes a number of inputs, weight them, sums them up, and uses the result as the argument for a singular valued function, the activation function. An extremely over-simplified description of the operation of biological neurons forms the basis of the model neuron proposed by (McCulloch and Pitts, 1943).

Layer

An approximation of the 3-dimentional interconnectedness of biological neurons is achieved in artificial neural networks by the use of layers. The three types of layers involved are input, output, and hidden. The input layer accepts some kind of real-world stimulus. This is transmitted to one or many hidden layers, which process the input information with regard to its connections with the input layer, and the weights of those connections. The results of these transformations are then passed to the output layer, and again processed with regard to connections and weights and are communicated to the user or environment.

Determination of input nodes

No definite method is known to determine the number of input nodes that should be used in research (Maier and Dandy, 2000). A try and error method in order to decide the number is suggested. This finding seemed to confirm the findings of a study by (Zhang, Patuwo and Hu, 1998). Most of the researcher determined the number of input node by the number of attributes that exists in the data and the number of output node is determined based on the number of target output set. However, the nodes number must be suited the data size since less number of nodes may cause

insufficient learning process, while exceeding number will lead to over specification.

Determination of nodes and number of hidden layer

The number of hidden layers and number of hidden nodes in each are some of the more difficult parameters to determine for a neural network model, as there is no formal procedure for creating the most efficient network. The hidden nodes are important because they directly relate to model performance. However, too many will lead to poor generalization and too few will result in an ineffective model.

There are several formulas that are suggested by researchers to determine the number of hidden layers in a study. The networks with the number of hidden nodes being equal to the number of input nodes are reported to have better forecasting results in several studies (Zhang, Patuwo and Hu, 1998). The use of Kolmogorov's Theorem to initialize the number of hidden nodes is a sufficient number of hidden nodes to model any continuous function (Subramanian, Yajnik and Murthy, 2003). The number of hidden nodes can be determined by the formula $2n+1$, where n is the number of input nodes. Another problem concerned is to define the number of hidden layers. Through the literature, the number of hidden layers must be adjusted until it shows the best result. However, Toth (Toth, Brath and Montanari, 2000) believed that one hidden layer would give the best result, as the uses of more than one hidden layer will result in the addition of parameter number.

Determination of output nodes

The determination of output nodes is based on the problem to be solved. The number of output nodes is relatively easy to specify as it is directly related to the problem under study.

Activation Function

Activation / transfer function is a nonlinear function that is used in determining net the output net for each neuron. The activation function has a range between 0 to 1 or –1 to 1. The transformations of the input to the output are performed by the transfer / activation function (*f*), which is also called as activation function. There are various choices for the transfer or activation function (Zhang, Patuwo and Hu, 1998) and they can be expressed mathematically as:

i) The sigmoid (logistic) function:
$$f(x) = (1 + \exp(-x))^{-1}$$

ii) The hyperbolic tangent function:
$$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$$

iii) The sine or cosine function:
$$f(x) = \sin(x) \text{ or } f(x) = \cos(x)$$

iv) The step function:

$$\begin{Bmatrix} 0\ x < 0 \\ +1\ x \geq 0 \end{Bmatrix}$$

v) The ramp function:

$$\begin{Bmatrix} -1\ x \leq -1 \\ x\ -1 < x < +1 \\ +1\ x \geq +1 \end{Bmatrix}$$

vi) The softmax/ exponential function:

$$\frac{e^{x}}{\sum_{i} e^{x_i}}$$

Neural network supports wide range of activation functions. Sigmoid function is a S-shape curve, with output in the range (0,1). Hyperbolic tangent function except the output lies in the range (-1,+1). Hyperbolic tangent often performs better than the logistic function because of its symmetry and ideal for customization of multilayer perceptrons, particularly the hidden layer. Sine function is possibly useful if recognizing radially-distributed data. Step function can be used to model simple networks such as perceptrons and outputs either 1.0 or 0.0, depending on whether the synaptic value is positive or negative. Ramp function is a piece-wise linear version of the sigmoid function and relatively poor training performance, but fast execution. Exponential function with results normalized so that the sum of activations across the layer is 1.0. It can be used in the output layer of multilayer perceptrons for

classification problem, so that the outputs can be interpreted as probabilities of class membership.

There are some heuristic rules for selection of activation function. However, it is not clear whether different activation functions have major effects on the performance of the networks (Zhang, Patuwo and Hu, 1998). Among them, sigmoid transfer function is the most popular choice [(Maier and Dandy, 2000, Toth, Brath and Montanari, 2000, Pan, Qin, Yang and S, 2008, Tang and Fishwick, 1993, Sharda and Patil, 1992)]. A commonly used function is the sigmoid function because it is self-limiting and has simple derivative. An advantage of this function is that the output cannot grow infinitely large or small (Hagan and Demuth, 2002). Besides, this function is especially advantageous for use in neural networks trained by back propagation, because it is easy to differentiate, and thus can dramatically reduce the computation burden for training. It applies to applications whose desired output values are between 0 and 1. However, other transfer functions may be used as long as they are differentiable (Maier and Dandy, 2000).

Training and testing data

Back propagation is used in two stages of information process, which are training and testing. Therefore, data used in a study should be divided into two dataset. The training data is a collection of data used to train and develop neural network, while the testing dataset is used to test the performance of data after the developed network is trained and stable.

Most researchers select the training and the test sample based on the rule of 90%:10%, 80%: 20% or 70%:30%. Some

choose them based on their particular problems. Gorr (Gorr, 1994) employ a bootstrap resampling design method to partition the whole sample into ten independent subsamples. Granger (Granger, 1993) suggests that for nonlinear forecasting models, at least 20% of any sample should be held back for an out-of-sample forecasting evaluation.

The normalized data must be divided into two parts. One will be the training data, the other one will be testing. To do so, a validation technique named k-fold cross validation will be used. The general idea of this technique is to divide the overall data into a number of *k* subsamples. Of the *k* subsamples, one part is retained as the validation data for testing the model, and the remaining part is used as training data. The cross validation technique is then repeated k times, with each of the k subsamples used exactly once as the validation data. Average of the result is compute from each of the experiment that has been performed.

Measurement of performance

Analysis of results is needed to measure performance of NNM in the prediction of dengue outbreak. An accuracy measure is often defined in terms of the forecasting error which is the difference between the actual and the predicted value. Although the authors of previous research use various evaluation measures in their networks, RMS error and its variations such as MSE and MAE are the most frequently used objective functions which can be explained by the simplicity of observing them. In the most analyzed applications, NN results outperform statistical methods, such as multiple linear regression analysis, logistic regression and others.

The MAD, SSE, MSE and RMSE that are frequently used performance measures are absolute measures and are of limited

value when used to compare different time series (Zhang, Patuwo and Hu, 1998). However, one method judged to be the best along one dimension is not necessarily the best in terms of the other dimensions. On the other hand, MSE will be used to measure the performance of neural network model since this technique has been approved in some previous studies (Subramanian, Yajnik and Murthy, 2003, Roliana Ibrahim, 2001 and Jastini Mohd Jamil, 2003). The formula of MSE is given below:

$$\text{Mean Square Error (MSE)} = \sum \frac{(e_t)^2}{N} \tag{9.7}$$

where;

ei is the individual forecast error and
N is the number of error terms.

NONLINEAR REGRESSION AS PREDICTION MODEL

Regression is used to study relationship between interval-ratio variables in which a single dependent (criterion) variable regresses with one or several independent (predictor) variables. It is used to analyze relationships that are characterized by straight line, or by generalizations of straight lines (Ramsay and Silverman, 2002).

Basic structure of nonlinear regression model

In econometric modeling, there are two types of variables namely the dependent variables and the independent variables. Usually in econometric model, the dependent variable is located on the left hand-side of the equation and on the right side of the equation is the independent variables or as so called explanatory variables.

Standard nonlinear regression models

i. Asymptotic regression

$$\beta_1 + \beta_2 \times exp(\beta_3 \times x) \tag{9.8}$$

ii. Gaussian

$$\beta_1 \times (1 - \beta_3 \times exp(-\beta_2 \times x^2)) \tag{9.9}$$

iii. Log-Log

$$ln(y_t) = \beta_0 + \beta_1 \times ln(x) + \varepsilon_t \tag{9.10}$$

iv. Polynomial

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t}^2 + ... + \beta_m x_m^m + \varepsilon_t \tag{9.11}$$

where;

Yt denotes the dependent variables

Xjt is the jth. Independent variable in period t

εt denotes the error term in period t and assumed identically, dependently and normally distributed with mean

$\beta 0$ and βj for m = 1, 2… m are unknown parameters to be estimated.

Another standard of nonlinear regression model are density, gompertz, johnson-schumacher, michaelis menten, ratio of cubic, ratio of quadratics and others but the recent used are log-logistic, linear and polynomial (Ramsay and Silverman, 2002). The data used to be predicted in this research is based on time. A scatter plot shows that there seems to be a strong relationship between dengue cases, rainfall data and nearest location of dengue cases data with time, but the relationship is nonlinear, so it requires the special estimation method of the nonlinear regression procedure. Besides polynomial is very versatile and fits many kinds of data.

The fundamentals of the ordinary least square technique

The ordinary least square (OLS) technique is used to minimize the sum of squares of the deviations of the estimated values. The deviation are known as the error or residuals, and represented as below and the architecture which has the smallest OLS is chosen as the best-fitted model. The discrepancy between the predicted values from the model fitted, $\hat{y}t$ and actual value yt is used to measure the model goodness of fit. The difference between the actual and the estimated value as know as the model error and can be written as equation 9.12. If the model performance is good, the error model will be relatively small.

$$e_t = y_t - \hat{y}_t \tag{9.12}$$

where;

et denotes the error term in period *t*
yt denotes the dependent variables
ŷ*t* denotes the estimated values
t is 1,2,3, …, *n*

The assumption pertaining to the model yt

The dependent variable should be linearly related to set of the independent variables, not only to one independent variable. All relevant should be included into the model, whereas the irrelevant variables should be removed from the model specification. Each of independent variables should non-dependent to any other independent variables. There should no multicollinearity among the independent variables. Lastly, the number of observations, *n* should be more than the number of regressor *p*.

The assumption pertaining to the error term εt

Error terms are also known as residuals, *εt* .The error terms are random in nature and they are assumed to be normally distributed with mean *E* (*et*) = 0 and constant variance.

$$E(e_t^2) = \delta^2 \tag{9.13}$$

The correlation between the errors terms are 0 (non-correlated). The covariance between the independent variables and the error terms is also 0.

Model validation and testing procedure

There are several statistical tests such as chi-square test, F test, t-test, coefficient of correlation, determination coefficient (R-square) and mean square error (MSE) but in this research we considered R-square test and mean square error (MSE) to select the best architecture.

R-Square test (R^2 - Architecture fitness)

In this test, R^2 is used to measure the total variation in *yt* that explained or accounted for by an estimated regression line. It shows the proportion if variation of estimated values *yt* t
y, and to the variation of actual values *yt*. The formul (9.14)

$$R^2 = 1 - \frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{n} (y_t - \bar{y}_t)^2}$$

below;
where;

yt denotes the dependent variables

$\hat{y}t$ denotes the estimated values
n number of observation,
$\overline{y}$ is the mean of $\hat{y}t$ observation
t is 1,2,3, …, n
R^2 with value 1 shows perfect fit and R^2 with value 0 shows extremely poor fit.

Mean square error (MSE)

For the purpose of the measuring the accuracy of model fitting, we consider the analysis of results is needed to the measure the performance of NLRM in prediction of dengue outbreak. For the purpose of the measuring the accuracy of model fitting, researcher will use MSE. The formula is given below;

$$MSE = \frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{n} \tag{9.15}$$

where;
yt denotes the dependent variables
$\hat{y}t$ denotes the estimated values
n number of observation,
t is 1,2,3, …, n

SUMMARY

This chapter contains a review of the dengue outbreaks in Malaysia, the importance of dengue outbreak prediction and brief revisions on neural network and regression model. Many experiments had

compared the results of using both NN and RM for modeling and predicting system. Previous studies on prediction proved that both of these models have the advantages and disadvantages. Most studies concluded that the used of NN produced data predictions more accurate or at least comparable to regression. However, many of these studies also found inconclusive or conflicting results and it is obvious that more research needs to be done.

REFERENCES

Cherkassky, J.H., Vladimir, H.F. and Wechsler, "From Statistics to Neural Networks," Berlin: Springer-Verlag, 1993.

Gorr, L., "Research Prospective on Neural Network Forecasting," International Journal of Forecasting, 10, pp. 1–4, 1994.

Granger, C.W.J., "Strategies for modelling nonlinear time series relationships," The Economic Record, 69 (206), pp. 233–238, 1993.

Gubler, D.J., "How Effectivelly is Epidemiological Surveillance used for Dengue Programme Planning and Epidemic Response?," Dengue Bulletin, Volume 26, Haemorrhagic Fevers, World Health Organization, Geneva, 2002.

Hagan, M.T. and Demuth, H.B., "Neural Network Design," PWS, Boston, Mass, USA, 1996 and 2002.

Jastini Mohd Jamil, "Pengkelasan Terhadap Data Pra-Pendiskretan dan Pasca- Pendiskretan Menggunakan Set Kasar dan Rambatan Balik: Satu Perbandingan," Tesis Sarjana, Universiti Teknologi Malaysia., Skudai, 2003.

Maier, H.R. and Dandy, G.C., "Neural Networks for the Prediction and Forecasting of water Resources Variable: A Review of Modelling Issues and Applications," Environmental Modelling & Software, 15, pp. 101-124, 2000.

McCulloch, W.S. and Pitts, W., "A logical Calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, 5, pp. 115-133, 1943.

Ministry of Health, Malaysia, "Annual Reports of Vector-Borne Disease Control Programme," 1990-2000.

Minns, A.W. and Balkema, A.A., "Artificial Neural Networks as Subsymbolic Process Descriptors," 17- 36, 1998.

Muammer N., Hasan G. and Ihsan T., "Comparison of Regression and Artificial Neural Network Models for Surface Roughness Prediction with the Cutting Parameters in CNC Turning," Research Articles: Modelling and Simulation in Engineering, Volume 2007, Article ID 92717, 14, 2007.

Pan, L., Qin, L., Yang, S.X. and S, J., "A Neural Network Based Method for Risk Factor Analysis of West Nile Virus," Risk Analysis, 28: 2, 2008.

Patz J.A., Willem J.M.M., Dana A.F, Theo H.J., **"**Dengue Fever Epidemic Potential as Projected by General Circulation Models of Global Climate Change," Environmental Health Perspective, Volume 106, No. 3, pp. 147-153, 1998.

Ramsay, J.O. and Silverman, B.W., "Applied Functional Data Analysis," Springler-Verlay, New York, 2002.

Roliana Ibrahim, "Carian Corak Kelas Data Indeks Komposit BSKL Dalam Perlombongan Data Menggunakan Model Rambatan Balik," Tesis Sarjana. Universiti Teknologi Malaysia, Skudai, 2001.

Roselina Salleh @ Sallehuddin, "Penggunaan Model Rangkaian Neural Dalam Peramalan Siri Masa Bermusim," Tesis Sarjana. Universiti Teknologi Malaysia, Skudai, 1999.

Rudnick, A., "Dengue Fever Epidemiology in Malaysia 1901-1980. In Dengue Fever Studies in Malaysia," Edited by A. Rudnich and T.W. Lim, Kuala Lumpur: Institute of Medical Research, Bulletin 23, pp. 9-38, 1986.

Seng, S. B., Chong, A. K. and Moore, A., "Geostatistical Modelling, Analysis and Mapping of Epidemiology of Dengue Fever in Johor State, Malaysia," The 17th Annual Colloquium of the Spatial Information Research Centre University of Otago, Dunedin, New Zealand, 2005.

Sharda, R. and Patil, R.B., "Connectionist Approch to Time Series Prediction: An Empirical Test," Journal of Intelligent Manufacturing, 3, pp. 317-323, 1992.

Subramanian, N., Yajnik, A. and Murthy, R.S.R., "Artificial Neural Network as an Alternative to Multiple Regression Analysis in Optimizing Formulation Parameters of Cytarabine Liposomes," AAPS PharmSciTech, 5(1), pp. 1 – 9, 2003.

Tang, Z., Fishwick, P.A., "Feedforward Neural Nets as Models for Time Series Forecasting," ORSA Journal on Computing. 5(4), pp. 374-385, 1993.

Teng, A. K. and Sing, S., "Epidemiology and New Initiatives in the Prevention and Control of Dengue in Malaysia," 25, pp. 7-14, 2001.

Toth, E., Brath, A. and Montanari, A., "Comparison of Short-Term Rainfall Prediction Models for Real-Time Flood Forecasting," Journal of Hydrology, 2000.

WHO, "Dengue and Dengue Haemorrhagic Fevers," WHO Fact Sheet 117, http://www.who.int/inffs/en/fact117.html, 2002.

Yang, Z.R., "A Novel Radial Basis Function Neural Network for Discriminant Analysis," IEEE Transaction on Neural Networks, 17(3), pp. 604-612, 2006.

Yao, X., "Evolving Artificial Neural Networks," Proceeding of the IEEE, 87(9), pp. 1423-1447, 1999.

Zhang, G., Patuwo, B.E. and Hu, M.Y., "Forecasting with Artificial Neural Networks: The State of the Art," International Journal of Forecasting, 35-62, 1998.

7
PEER-TO-PEER AS AN INFORMATION RESOURCES AND SEARCHING TECHNIQUES ACROSS THE PEER-TO-PEER NETWORKS

Iskandar Ishak
Naomie Salim

INTRODUCTION

The advancements of communication technologies and cheaper cost of storage in recent years have led to the development and innovation of distributed system over the internet. Data resources are scattered across the network and around the world. Peer-to-peer technologies has then surfaced to cater the need for users to search and retrieve these scattered data in a quick and cost-saving manned due to the dynamic and expensive nature of the internet.

In recent years, peer-to-peer networks have become one of the medium for the Internet users to share resources. In a peer-to-peer network, a peer acts as client and a server of the system. Peer-to-peer presents attractive solution through its scalability, fault-tolerance and autonomy. However, in their basic structure, peer-to-peer suffers high cost when dealing with locating content efficiently due to use of primitive searching and routing technique that uses large overhead and long query time. Therefore it is crucial to select relevant peers to route query message to reduce the number of messages used and answering time for better searching in unstructured peer-to-peer network without the loss of the unstructured peer-to-peer identity and characteristics.

PEER-TO-PEER CONCEPT

Schollmeier defines peer-to-peer as a distributed network architecture in which the participants share a part of their own hardware resources (processing power, storage capacity, network link, capacity, printers)(Schollmeier 2002). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration).

These peers are accessible by other peers directly, without going through intermediary entities. The participants of such a network are acting as (Service and content) providers as well as resource (Service and content) requestors (Servent-concept). Servent in this definition refers to server and client concept. This concept of server-client is different than the traditional client-server concept in which in peer-to-peer, a peer can act both as the server as well as the client.

Peer-to-peer Network Topologies

Existing peer-to-peer network topology has 2 architectures namely unstructured and structured. In unstructured systems, there are two more concepts that are being used which is the pure unstructured and ones that employ super-peer or super node model. Centralized peer-to-peer architecture employs the use of central server to provide directory services. Structured peer-to-peer system is where every peer is statically position on a coordinate based on the calculation of the system.

Unstructured

In unstructured approach, there is no central server that control the network topology and sharing services. There are 2 types of unstructured which is the pure unstructured approach, centralized and super-peers.

a. Pure Unstructured

Pure unstructured model is a peer-to-peer that features no centralized server or directory in the network. Peers join the network by connecting the existing participated peers randomly. All client computers (nodes) are in unstructured mode and considered to be equal in their capability for sharing resources with other network members. All requests are submitted through broadcasting using the flooding algorithm (Ripeanu 2001; Kwok, Chan et al. 2005). An example of application that is based on pure unstructured model is the Gnutella. In Gnutella, the query is routed by flooding to nodes until the query TTL (Time to Live) exhausted and the query has found its answers. Thus, lookup for this approach may generate O (N) message where N is the number of node. Pure unstructured network is illustrated in Figure 2.1

a. Centralized

In centralized peer-to-peer (Yang and Garcia-Molina 2001; Yang and Molina 2003; Koloniari and Pitoura 2005), a single peer maintained information about the contents of all peers in the system. Peers that enter the system publish information about their data in this central index that is consulted when a query is submitted. When a node makes a request for a data item, it searches at the server and returns the desired data address to the requesting peer. Then the requesting peer will access the node that stores the data directly without accessing through the centralized

peer. An example of peer-to-peer system that used this approach is Napster.

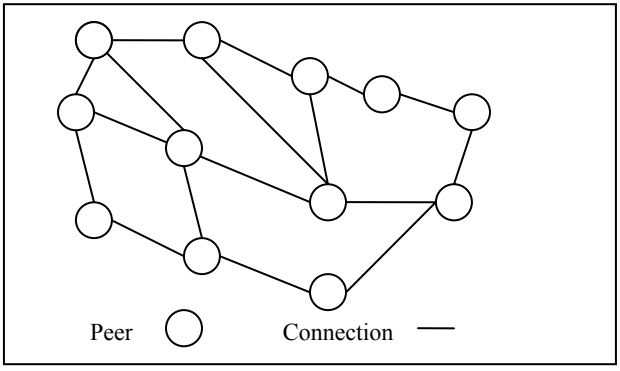

Figure 1 Pure Unstructured Topology

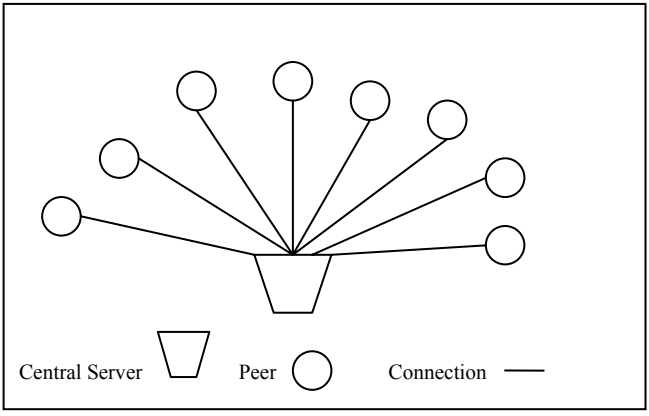

Figure 2 Centralized Topology

b. Super peers

In this approach, unlike unstructured peer-to-peer which has peers that are considered to have equal capability for sharing resources with other members, super-peer is a peer that is dynamically assigned based on the computer resources of peers. Super peers are peers which provide file indexing services for their connected peers (Kwok, Chan et al. 2005). In this context peers that are not super peers are called leaf node. A query from a node will be transmitted to the supernode which the query peer belongs to. The supernode will search its index records and will return the results to the query leaf node at once. Else, the query will be transmitted to other super peers and the results (if there is any) will be returned to the initial super peer. Figure 2.3 shows the architecture of the super-peer approach.

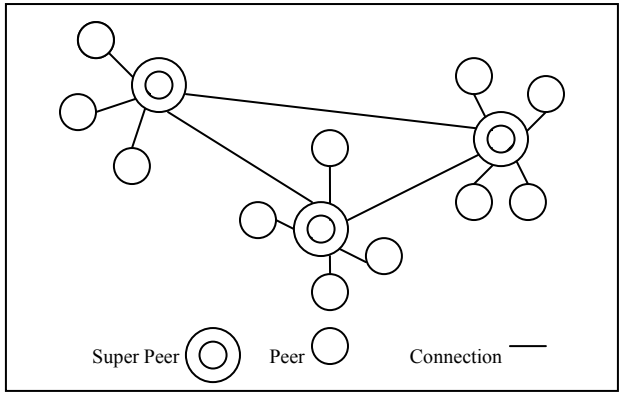

Figure 3 Super-peer Architecture

Structured peer-to-peer

In structured peer-to-peer system, data items are placed at specific nodes. The distribution of data items is based on Distributed Hash Table (DHT) methods such as CAN (Ratnasamy, Francis et al. 2001; Schmidt and Parashar 2005) and CHORD (Stoica, Morris et al. 2001; Schmidt and Parashar 2005). Hash table is a dictionary in which keys are mapped to array positions by a hash function. In DHT, each peer has its own hash table and stores keys that are mapped to them. DHT implements one operation which is the lookup (key), which routes the request to the peer responsible for storing the key.

In CHORD, each peer has a unique identifier ranging from *0* to 2^k*-1*. These identifiers are arranged as a circle with modulo 2^k. Each peer maintains information about its predecessor and successor on the circle. Each peer also stores information about *k* other neighbors, which is called fingers, in a table called the finger table. The ith finger peer is the first peer on the circle that succeeds the current peer by at least 2^{h-1}, where $1 \leq i \leq k$. Data elements are mapped to peers based on their keys which will be as its identifier. A data element is mapped to the first peer identifier which is equal to or follows its key. This peer is called the successor of the key.

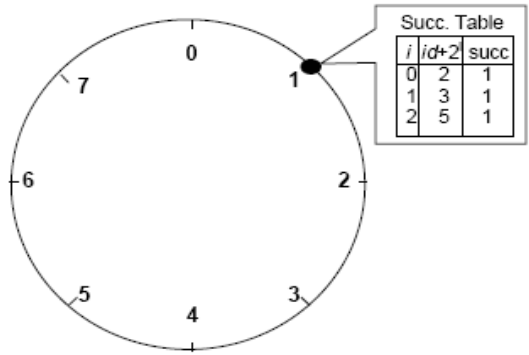

Figure 4 CHORD Topology

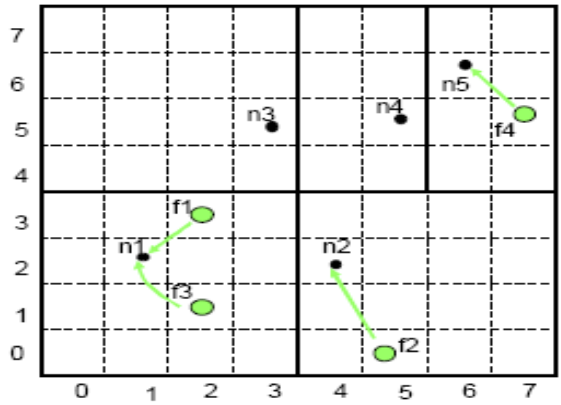

Figure 5 CAN topology

In CAN, it uses d-dimensional Cartesian coordinate space on a d-torus. This coordinate space is dynamically partitioned among all the nodes in the system. Each node is associated with a zone in this coordinate space, and its neighbors are nodes that "own" adjacent zones. Each data element is mapped to a point in this coordinate space using a uniform hash function. The data element is stored at the node that owns the zone in which the point lies.

PEER-TO-PEER AS INFORMATION RESOURCES

The recent development of fast internet access and cheaper data storage has turned peer-to-peer into main information resources for internet users. Internet users can store a lot of files in their personal computers or notebooks as disk storage are getting bigger and cheaper recently (Levin 2006). The users can easily search and download files such as text, audio and images files over the internet.

Earlier, peer-to-peer has started as the platform for information resources through Napster. In this system, users can download music files they want over the internet. KaZaa which uses different approach of peer-to-peer architecture but for the same purpose as Napster offers users to search document or files that available from other users in the internet. Users can easily search and find their intended information or files quickly rather than going to the library to borrow some books or even to go the music store to find new album. The emergence of bigger and better hardware to record music, video or documents into a compact disc (CD) or a DVD has even increased the popularity of peer-to-peer application to find those kinds of files over the internet. Therefore, the need for efficient and effective peer-to-peer search is ever demanding.

Peer-to-peer is dynamic in nature, large in size and is a very complex system. Areas that peer-to-peer system has been actively involved in computer science are not just networking, but also involve database, information retrieval. Peer-to-peer has become the most important of information resources due to its dynamicity, self-organization, less redundancy, availability and fault tolerance.

Efficient information searching and discovery is difficult to implement due to the unavailability of global knowledge about other peers because of the decentralized nature of the peer-to-peer network. The dynamic nature of the peer-to-peer network where peers can join and leave at anytime make the efficient searching is even difficult. The demand for advance searching technique is always there as the Peer-to-peer becoming larger and more complex.

Table 1 Comparison of peer-to-peer topology

Peer-to-peer approach	Example	Advantage	Disadvantage
Pure unstructured	**Gnutella**	**i. Robust and tolerance fault** **ii. Totally unstructured, eliminating possible bottleneck**	**i. Not scalable** **ii. The entire network can be swamped with request, burden the network traffic** **iii. Inefficient lookup which**

		iii. Highly robust	may generate O(N) message per lookup
Super-peer	**(Kwok, Chan et al. 2005)**	i. Maintaining index at super-peer is smaller than centralized approach. ii. Reduce the possibility of excessive network traffic	i. Super-peer could become a single point failure for its cluster and potential bottleneck. ii. Failed super-peers could make all clients become disconnected until new super-peer is selected.
Structured	**CAN, CHORD**	**Efficient lookup; CHORD only needs O(log (N)) messages per lookup, CAN has routing tables of O(d) size and guarantees file is found in $d*n^{1/d}$ steps**	i. Does not support complex metadata sets for data source, only simple filenames ii. Impose restrictions on

			data placement
Centralized	**Napster**	**i. Simple architecture** **ii. Easy to adopt sophisticated search engine on top of the index system because of one centralized server involve for lookup**	**i. Not scalable** **ii. Centralized server could become a single point of failure and potential bottleneck**

SEARCHING IN PEER-TO-PEER NETWORKS

Searching in peer-to-peer networks is the operation that allows complex keyword queries using keywords to search files such as text, video or audio in the network. Peer-to-peer searching approach can be classified based on the architecture or its network topology.

Searching in Unstructured Peer-to-peer Networks

In its early form, searching in unstructured peer-to-peer networks basically involves a simple flood-based approach to search files. The search query is propagated to all the neighbors from the source peer. The query is then replicated and send to other connected peers or their neighboring peers. Flood-based is costly when it is scaled to large number of queries and can only find items within defined radius.

Flooding technique is used in file-sharing peer-to-peer application Gnutella. In this system, each query from a peer will be broadcasted to all the peers in the network but restricted by the TTL (Time to Live) value. Flooding may generate O (N) message where N is the number of node. As a result, the query consumes a great deal of processing resources and excessive network. In a worst case situation such as low bandwidth network, flooding could turned the network become a bottleneck. It is well known for its robustness and simple searching technique but it involves a great deal of communication overhead. Hop number or hop count is also increased exponentially. Some of the messages might visit the same node that has been searched previously. As a result, it involves large communication overhead and becoming the major problem with this approach. Flooding technique can pose a query and this will burden the traffic. These problems has been proven in a number of studies (Ripeanu 2001; Ramanathan, Kalogeraki et al. 2002; Kwok, Chan et al. 2005) .

In the random Breadth-First-Search (BFS) approach (Dimakopolous and Pitoura 2003; Zeinalipour-Yazti, Kalogeraki et al. 2004), each peer forwards a search message to only a fraction of its peers. Each node randomly selects a subset of peers connected to it and then propagates the search message to those peers. The advantage of this technique is that it does not require any global knowledge. Every node is able to make local decision in a quick manner since it needs only small portion of connected peers to route the query. This approach may generate only a fraction of messages used by flooding approach.

In Random walks (Lv, Cao et al. 2002) approach, the requesting peer sends out a number of query messages to an equal number of random neighboring peers. Each of the query messages will follow its own path in which intermediate peers will forward the messages to randomly chosen neighbor. These query messages is known as walkers. The walkers will be terminated on both success and failure occasions. Failure is determined through the use of TTL of the messages or through the use of checking method in which the walker periodically contact the source peer whether the termination condition is met. The approaches achieve message reduction when compared to the flooding approach. However, the success rate and the number of hits depend largely on the network topology and the random choices it made.

Another unstructured peer-to-peer searching approach is the Directed BFS combined with the Most Result in the Past in (Yang and Garcia-Molina 2002). In this approach, a query is defined to be satisfied if X, for some constant X, or more results are returned. A peer forwards a search message to a number of peers which returned the most results for the last M queries. The nature of this approach is it allows peers explore larger network segments and find most stable neighbors.

A content-based searching for peer-to-peer based system is proposed in (Koloniari and Pitoura 2004). In this approach, each peer will have a special index called filters to facilitate query routing only to those that may contain relevant information. Each peer maintains one filter that summarizes all documents that exist locally in the peer, called local filters. A merged filters is the filter that summarizing the document of a set of its neighbors. When a query reaches a peer, the peer will check its local filter and uses the merged filter to route the query to the peers whose filters match the query.

(Zeinalipour-Yazti, Kalogeraki et al. 2004) proposed a searching technique based on the similarity of the query. In this approach, each peer has its own profile table that stores the information they get from peers that answered their queries. The information stored in this table is the query ID, peer ID, and the

query keywords that have been answered and also the query hit. Only the latest peer that answered the query will be kept into the table of a size t. Routing is based on the similarity values of the query word with the keyword from the past queries stored in the profile. Peers that have high similarity with the query will be selected for routing.

Ant Colony optimization is also used in unstructured peer-to-peer search by Michlmayr in (Michlmayr). The approach is called SemAnt where it imitates the nature of ants cooperating between themselves to find food based on the pheromone. In SemAnt, the peers act like an ant, and cooperate in creating pheromone trails which is the probabilistic overlay networks and indicates the most promising path for a given query. As a result, the more popular a query becomes, the better the trail. Her experiments shown that the search algorithm is stable, robust and converges fast whole its performance is pretty much acceptable.

(Ishak and Salim 2008) proposes the combination of the use of query similarity and peer connection stability to determined peer relevance for peer-to-peer search in unstructured peer-to-peer network. The work is based on the use of query similarity and query connection space. Peers that have optimum value in both similarity of past query and current query along with optimum number of hits will be determine as relevance peer. Each peer holds a table which stores past history of query hits, and query similarity along with the answering peer IDs.

In centralized approach, peer-to-peer is based on the use of centralized server, where all peers are connected to one central server. This server will be the single reference point for all of the peers for searching data objects or files within the network. However, there is major drawback created by this approach. Theoretically, (Yang and Molina 2003) stressed out that the cost of housing the centralized index is very high.

Super peer architecture is based on combination of centralized in unstructured approaches. In this approach, there will be a number of peers that will be act as a dedicated peer or super peer. Queries will be forwarded to the super peer, and then the

super peer will submit the queries to the child node that connected to it. Super peer is also connected to other super peer queries will be submitted to other super peers and thus, expanding the search. This approach reduces the network traffic as queries are submitted to fewer peers as opposed to the flood-based query strategy.

This has been proven in (Yang and Molina 2003) where it performs better in terms of reducing load of individual peers (when compared to a centralized approach that uses single centralized server), lower message overhead is needed because query is submit to the super-peer. Then the super-peers will submit this query to its subordinate peers or if no response, it will be send to other super-peers. However, super-peer searching could succumb to the threat of bottleneck that could be caused by excessive access to the central server by large number of peers at the same time which has been mentioned in (Ramanathan, Kalogeraki et al. 2002). Thus it will reduce the performance of the peer-to-peer searching.

Searching in Structured Peer-to-peer Networks

Searching or data lookup in CHORD involves such simple steps. A node forwards a request or query to the neighbor that is closest to the key. This process involves *O (log N)* messages for each lookup in sending the query. The lookup, just like CHORD, is done through forwarding the request to a neighbor that is closest to the key. Thus it guarantees that file could be found in $d*n^{1/d}$ steps where d is the diameter in a routing table size of *O (d)*.

Distributed Hash Table (DHT) based approach is developed to solve the problem caused by previous peer-to-peer approaches; unstructured peer-to-peer that uses flood-based query and centralized approach that uses centralized server to serve all peers in the network, which are used for file sharing purposes. Structured peer-to-peer approaches are primarily based on the use of Distributed Hash table (DHT). DHT-based peer-to-peer provides better retrieval performances compared to flood-based searching and centralized server approach. It is shown through

better computing workload and to a certain extent, better query answering.

These approaches have turned searching in the peer-to-peer system to perform more efficiently as queries are routed to destination node with less network workload and results are found in very small number of messages (Crespo and Molina 2003). However, existing file sharing peer-to-peer lacks semantic interpretations in searching and in organizing peers and resources because it uses keyword-based searching and hash algorithm to placed peers into locations in the network based on the hashed value of the peer address.

However, according to (Shu and Yu 2006), even though CHORD did locate peers and data efficiently through exact-key lookup but more complex search are necessary for supporting most real-life applications, e.g. a user wants to retrieve documents containing terms or features that are most relevant to his queries. It means that CHORD lacks in utilizing the semantic of the query that is more complex like the database query to match with the semantic of the document or the data sources. According to Nejdl et al. (Nejdl, Wolpers et al. 2003), CAN did neither address complex metadata sets nor complex queries which is strongly needed in database query and retrieval. DHT-based approach also imposed restriction on data placement, thus peers will have less autonomy on the content that they store.

CONCLUSIONS

This chapter presented an overview of the peer-to-peer system, peer-to-peer information resources and the searching approaches across the peer-to-peer networks. Peer-to-peer can be classified into unstructured and structured architecture. Within the unstructured there are also a few more class of peer-to-peer architecture namely pure unstructured network, centralized and super-peers. The strength and witnesses of each and one of them

has been also discussed. Peer-to-peer as information resources over the Internet has also been discussed. The advancements of the internet technology and cheap disk storage has become the catalyst for the internet users to use peer-to-peer application to search and find data and information over the Internet.

Peer-to-peer system also offers a number of searching technique to cater the large and dynamic system over the internet. The searching techniques can also be classified based on the peer-to-peer architecture. Structured peer-to-peer network impose static and strict policy of searching. The searching is very efficient but it has to pay a price of maintaining the structured overlay. Searching in unstructured peer-to-peer network has only small number of policies to follow and using minimal of data and information. Thus, the search in unstructured peer-to-peer can be inefficient and expensive. However, a number of researchers have proven that searching in unstructured peer-to-peer can be effective, and less expensive.

REFERENCES

Gnutella. www.gnutella.com.

Kazaa. www.kazaa.com.

Napster. www.napster.com.

Crespo, A. and H. G. Molina (2003). "Semantic Overlay Networks for Peer-to-Peer System."

Dimakopolous, V. and E. Pitoura (2003). A Peer-to-peer Approach to Resource Discovery in Multi-Agent Systems. International Workshop Series on Cooperative Information Agents 2003, Helsinki, Finland, LNCS, Springer.

Ishak, I. and N. Salim (2008). Exploiting Query Feedbacks for Efficient Query Routing in Unstructured Peer-to-peer Networks. International Conference on Multimedia, Internet and Web Engineering (MIWE'08), Singapore, World Academy of Science, Engineering and Technology.

Koloniari, G. and E. Pitoura (2004). "Content-Based Routing of Path Queries in Peer-to-peer Systems." Advances in Database Technology **2992**: 29-47.

Koloniari, G. and E. Pitoura (2005). "Peer-to-Peer Management of XML Data: Issues and Research Challenges." SIGMOD Record **34**(2): 6-17.

Kwok, S. H., K. Y. Chan, et al. (2005). "A Server-mediated Peer-to-peer System." ACM SIGecom Exchanges **5**(3): 38-47.

Levin, M. (2006)Storage Management Disciplines are Declining. http://www.computereconomics.com/article.cfm?id=1129.

Lv, Q., P. Cao, et al. (2002). Search and Replication in Unstructured Peer-to-peer Networks. International Conference on Supercomputing 2002, New York, USA, ACM.

Michlmayr, E. Ant Algorithms for Search in Unstructured Peer-to-Peer Networks Ph.D. Workshop,22nd International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia, USA, ACM.

Nejdl, W., M. Wolpers, et al. (2003). Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. WWW2003, Budapest, Hungary, ACM.

Ramanathan, M. K., V. Kalogeraki, et al. (2002). Finding Good Peers in Peer-to-Peer Networks. IPDPS, IEEE Computer Society.

Ratnasamy, S., P. Francis, et al. (2001). A Scalable Content-Addressable Network. SIGCOMM'01, San Diego, California, ACM.

Ripeanu, M. (2001). Peer-to-Peer Architecture Case Study: Gnutella Network. P2P'01, IEEE.

Schmidt, C. and M. Parashar (2005). Peer-to-Peer Information Storage and Discovery Systems. Peer-to-Peer Computing: The Evolution of a Disruptive Technology. R. Subramaniam and B. D. Goodman. Connecticut, USA, Idea Group Publishing**:** 79-112.

Schollmeier, R. (2002). A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. P2P'01, IEEE.

Shu, Y. and B. Yu (2006). Clustering Peers Based on Contents for Efficient Similarity Search. DASFAA 2006, Springer-Verlag Berlin Heidelberg.

Stoica, I., R. Morris, et al. (2001). "Chord: A Scalable Peer-to-peer Lokup Service for Internet Applications." ACM: 149-160.

Yang, B. and H. Garcia-Molina (2001). Comparing Hybrid Peer-to-Peer System. 27th VLDB Conference, Roma Italy, VLDB.

Yang, B. and H. Garcia-Molina (2002). Efficient Search in Peer-to-peer Networks. Proceeding of the International Conference on Distributed Computing System, Vienna, Austria.

Yang, B. and H. G. Molina (2003). Designing a Super-Peer Network. ICDE '03, IEEE.

Zeinalipour-Yazti, D., V. Kalogeraki, et al. (2004). "Exploiting locality for scalable information retrieval in peer-to-peer networks." Information System 30: 277-298.

8
QUERY SIMILARITY BASED SEARCH IN UNSTRUCTURED PEER TO PEER NETWORKS

Iskandar Ishak
Naomie Salim

INTRODUCTION

Peer-to-peer networking has faced rapid development and becoming one of the most popular Internet applications during these recent years. It has gained a tremendous popularity especially on the use of sharing resources between peers in the internet. Peer to peer application in its earlier years was made popular by file sharing applications such as Napster (http://www.napster.com) and Gnutella (http://www.gnuttella). Unstructured peer-to-peer networks (Lv and Cao, 2002) are popular due to its robustness and scalability. Query schemes that are being used in unstructured peer-to-peer such as the flooding and interest-based shortcuts suffer various problems such as using large communication overhead long delay response. The use of routing indices has been a popular approach for peer-to-peer query routing. It helps the query routing processes to learn the routing based on the feedbacks collected. In an unstructured network where there is no global information available, efficient and low cost routing approach is needed for routing efficiency.

In this chapter, we present a decentralized, distributed and cost effective unstructured peer to peer query routing approach. It takes into account the past queries stored and connection

information that will determine the stability of the peers to be routed. Therefore, only selected peers that relevant to the incoming query and also having stable connection will be selected to be routed. Our approach does not acquire global knowledge to determine peers that are relevant to the query. The remainder of this chapter is organized covers reviews on the related work. Explanation and description of the proposed routing technique is given further in this chapter. Simulation and evaluation of the approach were also explained and the results were concluded at the end of the chapter.

RELATED WORK

The earliest technique for peer-to-peer routing is based on the Naïve Breadth-First Search (BFS) algorithm or Flooding. This technique is used in file-sharing peer-to-peer application Gnutella [2]. In this approach, each query from a peer will be broadcasted to all the peers in the network but restricted by the TTL (Time to Live) value. Flooding may generate O(N) message where N is the number of node. As a result, the query consumes a great deal of processing resources and excessive network. In a worst case situation such as low bandwidth network, flooding could make the network become a bottleneck. Although, it is a robust and simple technique for query routing but it involves a great deal of communication overhead, that is, high in number of messages. Hop number or hop count is also increased exponentially. Some of the messages might visit the same node that has been searched previously. Therefore, communication overhead and scalability are the main problems in this approach.

In the random BFS approach (Zeinalipour-Yazti, Kalogeraki and Gunopolus, 2004, Dimakopolous and Pitoura,

2003)], each peer forwards a search message to only a fraction of its peers. Each node randomly selects a subset of peers connected to it and then propagates the search message to those peers. The advantage of this technique is that it does not require any global knowledge. Every node is able to make local decision in a quick manner since it needs only small portion of connected peers to route the query. This approach may generate only a fraction of flooding query messages or log O (N) messages.

Another unstructured peer-to-peer routing approach is the Directed BFS combined with the most result in past by Yang & Molina(Yang and Garcia-Molina, 2002). In this approach, a query is defined to be satisfied if X for some constant X or more results is returned. A peer forwards a search message to a number of peers which returned the most results for the last M queries. The nature of this approach is it allows peers explore larger network segments and find most stable neighbors.

Interest based routing (Sripanidkulchai, Maggs and Zhang, 2003) tries to avoid the blindness of flood-based routing by favoring nodes sharing similar interest in the source. In this approach, nodes which have similar interest is grouped together and the queries are routed to these nodes in hoping that it will shorten the time for the queries to get the answer.

Koloniari et al. (Koloniari and Pitoura, 2004) proposed a content-based routing for peer-to-peer based system. In this approach, each peer will have a special index called filters to facilitate query routing only to those that may contain relevant information. Each peer maintains one filter that summarizes all documents that exist locally in the peer, called local filters. A merged filters is the filter that summarizing the document of a set of its neighbors. When a query reaches a peer, the peer will check its local filter and uses the merged filter to route the query to the peers whose filters match the query.

Zeinalipour-Yazti et. al (Zeinalipour-Yazti, Kalogeraki, and Gunopolus, 2004) proposed a routing technique based on the similarity of the query. In this approach, each peer has its own profile table that stores the information they get from peers that

answered their queries. The information stored in this table is the query ID, peer ID, and the query keywords that have been answered and also the query hit. Only the latest peer that answered the query will be kept into the table of a size t. Routing is based on the similarity values of the query word with the keyword from the past queries stored in the profile. Peers that have high similarity with the query will be selected for routing.

SIMILARITY BASED SEARCH

We start with an overview of our peer similarity-hits graph model followed by details of its construction. Our peer-similarity graph model captures both the peer query hits and peer similarity with corresponding incoming query. We used both information gathered in the profile table which is based on the work done by Zeinalipor. We incorporate both, query content and connection stability information to determine relevant peer to route query. Each peer stores information about past queries and the query hits in a table. There will be no global knowledge shared between all the peers but each peer will also have a list of data collected from the answered query and store it in Neighbor Profile Table (Table 1).
Peers will be ranked based on the relevance value before the query is submitted. The relevance value will be based on two parameters, query hits and the similarity value between the query to be routed and the stored past queries. Query hits determine peer connection stability with the processing peers. The more query hits, the more stable the peer is connected and thus giving the impression of the particular peers connection reliability.

Table 1 Neighbor Profile Table

Query	ID	Connection and hits	Timestamp
Formula One race	E234	(P1,25), (P3,1),(P5,20)	10123
gulf oil	D233	NULL	10224
Waste disposal	G234	(P11,15), (P13,11),(P15,20)	10979

Similarity value will determine the content that the particular peer has in its storage. As an example, let peer *A* has a list of past queries, *d*. Query *q* is an incoming query and is waiting to be routed. Query *q*, will be compared with all the queries in d. Peers that are associated with queries in list *d*, which are similar with query q, will be selected for routing, based on the relevance value.

Therefore, both parameters are needed to determine the relevance of a peer to be routed. Peers that have higher query hits but less similarity will also be considered to be rank higher. The peer ranking will be based on the relevance value in which the smaller the relevance value the higher possibility the peer will be rank higher and selected for query routing for that particular query.

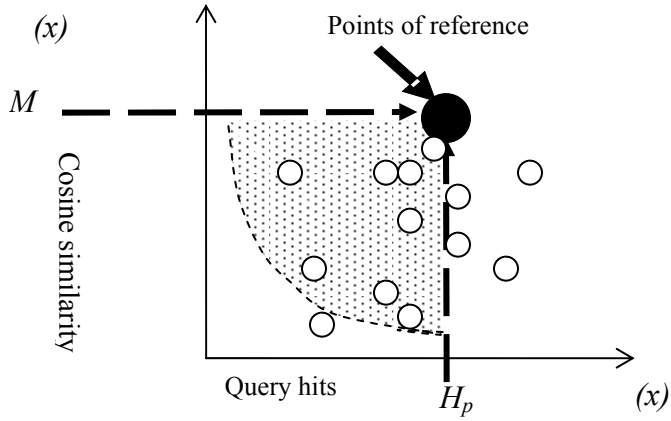

Figure 1 Query Similarity-Peer Stability graph

The Neighbor Profile

The Neighbor Profile or the query feedback table is based on the work done by Zeinalipour-Yazti et. al (Zeinalipour-Yazti, Kalogeraki, and Gunopolus, 2004). The list will contain the ID of the answering peer, connection ID, the query keywords that have been answered by other peers and a timestamp of the returned query. These keywords are actually the words that match the query sent by this peer, and this shows that these words are contained in the peer that answered this query. The list will keep the last M queries and a Least Recently Used (LRU) policy will keep the most recent queries in the table.

Point of Reference

A point of reference is selected to determine a peer's relevance, where we can see in Figure 1. The point is based on the optimal point of both parameters, query hits and query cosine similarity. Maximum point, *M* on the y-axis is the highest cosine similarity value. Similarity point that has similarity value that is near to point *M* is more similar to the incoming query. While, maximum query hits value, H_p on the x-axis is the highest recorded query hits. It will be selected from the list of query hits for all recorded past query. The *max* function (2) selects the highest query hits of a query from the profile table Therefore, the point of reference will be the point that has highest similarity value and highest query hits. The point where Hp and M meets on as shown in Figure 1 is the Point of Reference.

$$H_p = \max(h_i) \qquad (2)$$

Peer Relevance

The peer relevance is determined as follows:

$$R(q, q_i) = \sqrt{\left(\frac{H_p - h_i}{N_p}\right)^2 + \left(M - sim(q, q_i)\right)^2} \qquad (3)$$

M is the maximum cosine value, and we decided to use *M = 1* as a default highest similarity value. h_i is the returned hits values for a particular query, while H_p is the maximum hits retrieved from all *h*

that have been recorded. N_p is the total number of query hits of all peers stored in the Neighbor Profile Table. q is the incoming query while q_i is the stored queries in the neighbor profile table.

SIMULATIONS

We evaluate the performance of our routing approach by extending a peer-to-peer simulator called Peerware (Zeinalipour-Yazti,). 230 nodes are generated and a total of 23336 documents are used. Each node holds random number of documents. Reuters-21578 document collection which appeared on the Reuters newswire in 1987 is used in the simulation. 100 queries are used to query the documents in the simulation and each peer is country based and each node hold documents for just one country. In(Ishak and Salim, 2008), our routing approach performed better than the Breadth-First Search (BFS) routing approach, while in this study we tested our approach with the Most Query Hits (MQH) (Yang and Garcia-Molina, 2002) and the Intelligent Search Mechanism (Zeinalipour-Yazti, Kalogeraki, and Gunopolus, 2004) approach using larger number of queries. In this simulation other approaches also use the query feedback data for routing the query in the unstructured peer-to-peer networks.

We evaluate our routing approach in terms of number of message used and time taken for every query hits on different TTL settings. Network efficiency is the total of query hits over total number of messages for each TTL values (4). The bigger the value means the more efficient the approach is since few number of messages are needed for getting high query hits. We also evaluate time efficiency by calculating the sum of query hits for all queries over the total time for each TTL values (5). The smaller the value meaning that less time is taken to find all the query hits across the network.

$$\text{Time Efficiency} = \frac{\text{QueryHits}}{\text{Query Time (ms)}} \quad (4)$$

$$\text{Network efficiency} = \frac{\text{QueryHits}}{\text{No. of Messages}} \quad (5)$$

RESULTS AND ANALYSIS

In this section we present the results and analysis of our simulation. Figure 2 illustrates that our searching approach recorded higher query hits in all TTL settings than other approaches except when TTL=3, where the ISM approach have a slight 12% advantage over our searching method. On average our approach recorded 44% higher query hit rates than ISM and 60.9% higher than the MQH for all TTL settings. However, in terms of messages used, our approach recorded the highest use of messages on all TTL settings as we can see in Figure 3. Despite the high messages usage, our approach is rather efficient than other approach because we recorded highest efficiency in terms of query hits per message usage as illustrated in Figure 5.

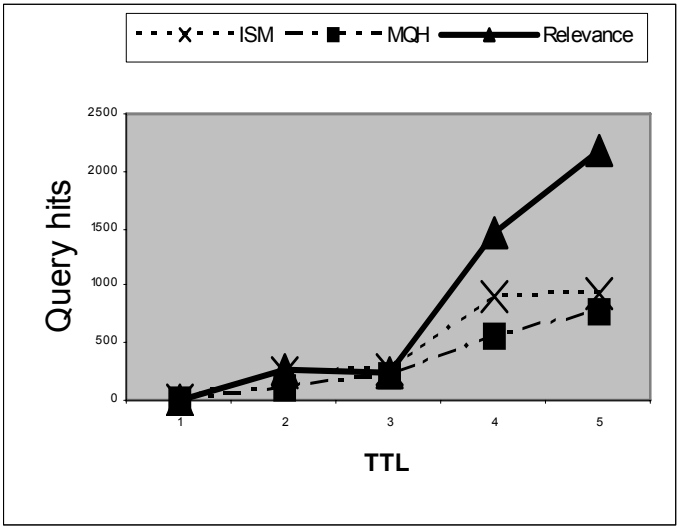

Figure 2 Number of query hits on different TTL settings

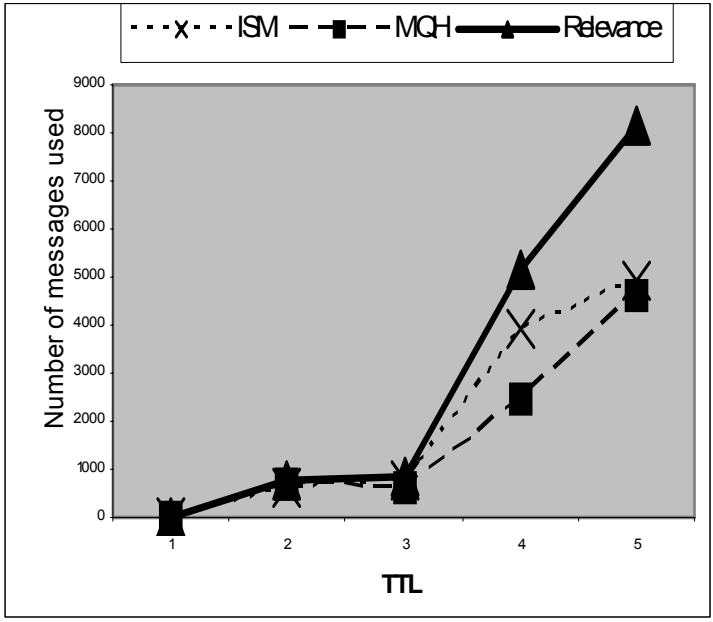

Figure 3 Number of messages used different TTL settings

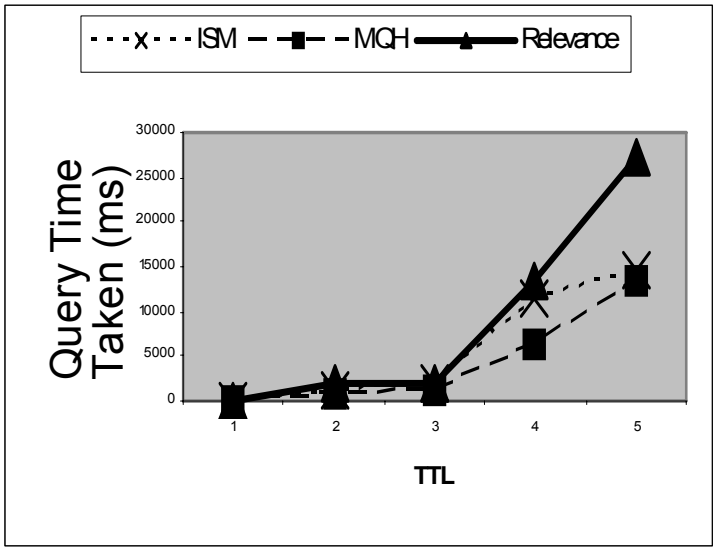

Figure 4 Query time taken based on different TTL settings

Figure 4 illustrate the delay or time taken for each successful search. Our approach topped as the approach with the highest delay for all TTL settings. This is due to the time for processing the similarity of the query, peer ranking and selections for query routing. However, the long delay is justified by the efficiency of our approach that is illustrated in Figure 6. Figure 6 illustrates that our searching approach have among the highest efficiency of successful queries over time taken. We can see that our approach recorded slightly lower time efficiency when TTL is 2 and 3. ISM recorded the highest time efficency when TTL is set to 2 while MQH is the most time efficient searching method when TTL is set

to 3. when TTL is set to 4 and 5, our approach recorded the most efficient approach in terms of time taken for successful search.

In Figure 6, we can see that our approach showed increasing efficiency of query hits over number of messages used. Other search approach showing decrease in efficiency as the TTL is increased and the efficiency of our approach only decrease slowly. When the TTL is set to 4, our approach topped the message efficiency graph by 60% over the MQH approach and 45% over the ISM approach. When the TTL is set to 5, our approach again becomes the most efficient in terms of messages used for successful search by 70% over MQH and 60% over ISM.

Figure 7 illustrate the delay time for each searching approaches. We can see that our approach recorded higher delay in query answering especially when TTL is set to 4 and 5. When TTL=4 in which we have a slight 12% higher than ISM but a 50% higher query time than MQH approach. When TTL=5, our approach recorded 46.9% higher query time than ISM and 50.7% higher query time than MQH. The long query time taken by our approach is justified when we compare it with the number of query hits in which we recorded highest query hits as we explained previously. Formula (5) is used to calculate our time efficiency for each TTL and we can see that our approach recorded high time efficiency especially when TTL=4 and TTL=5.

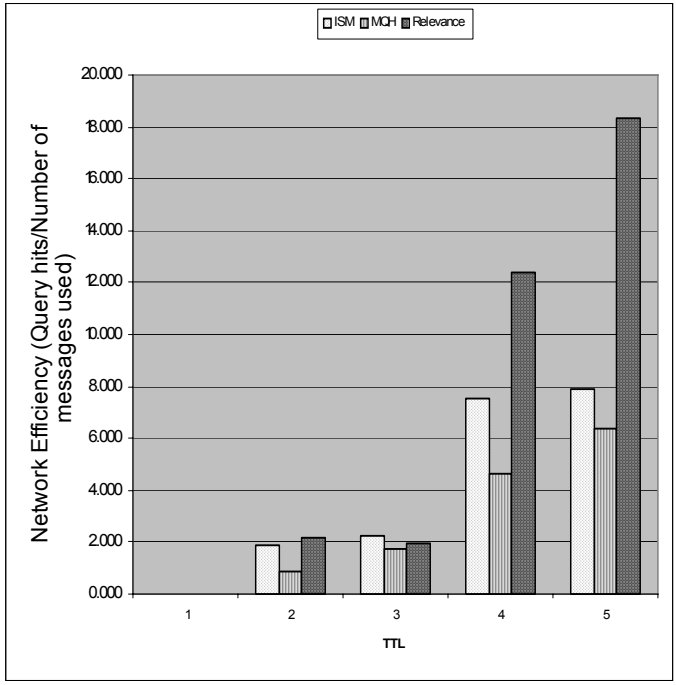

Figure 5 Efficiency of query over number of messages used

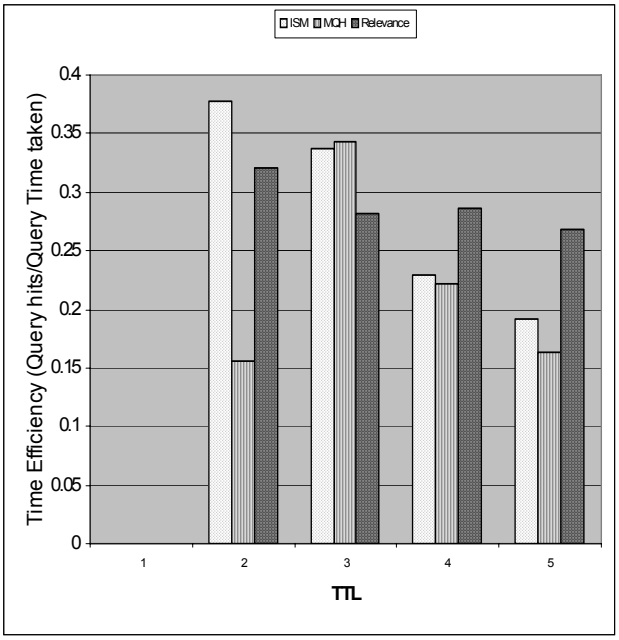

Figure 6 Efficiency of query over query time

CONCLUSION AND FUTURE WORK

The main purpose of this research is the need of selective query routing approach in unstructured peer-to-peer network that consume better network traffic cost and reduced query time. Our simulation test shows that our routing approach performs better than the other two (ISM and MQH) approach in terms of retrieved result, messages used and query time. We showed that by and using minimal information of query hits and query similarity, efficient routing in unstructured peer-to-peer network can be achieved.

REFERENCES

Dimakopolous, V. and Pitoura, E., "A Peer-to-peer Approach to Resource Discovery in Multi-Agent Systems," International Workshop Series on Cooperative Information Agents 2003, Helsinki, Finland, 2003.

Ishak, I. and Salim, N., "Exploiting Query Feedbacks for Efficient Query Routing in Unstructured Peer-to-peer Networks," accepted for publication in proceedings of International Conference on Multimedia, Internet and Web Engineering (MIWE'08), Singapore, 2008.

Koloniari, G. and Pitoura, E., "Content-Based Routing of Path Queries in Peer-to-peer Systems," *Advances in Database Technology*, vol. 2992, pp. 29-47, 2004.

Lv,Q., Cao, P., E. C. A. T. Labs-Research, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-peer Networks," International Conference on Supercomputing 2002, New York, USA, 2002.

Sripanidkulchai, K., Maggs, B. and Zhang, H., "Efficient content location using interest-based locality in peer-to-peer systems," 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03), San Francisco, Californa, USA, 2003.

Yang , B. and Garcia-Molina, H., "Efficient Search in Peer-to-peer Networks," Proceeding of the International Conference on Distributed Computing System, Vienna, Austria, 2002.

Zeinalipour-Yazti, D., Kalogeraki, V. and Gunopolus, D., "Exploiting locality for scalable information retrieval in peer-to-peer networks," Information System, vol. 30, pp. 277-298, 2004.

Zeinalipour-Yazti, D., "Peerware," http://www.cs.ucr.edu/~csyiazti/peerware.html

9
ONTOLOGY EXTRACTION

Saidah Saad
Naomie Salim

INTRODUCTION

Ontology is an important emerging discipline that has the huge potential to improve information organization, management and understanding. Ontology has become an important mean for structuring knowledge and building knowledge-intensive systems. The importance of domain ontologies is widely recognized, particularly in its relation to the expected advent of the Semantic Web. As the term refers to the shared understanding of some domains of interest, which is often conceived as a set of concepts, relations, functions, axioms and instances (Gruber, 1993), the goal of a domain ontology is to reduce the conceptual and terminological confusion among the members of a virtual community of users that need to share electronic documents and information of various kinds.

According to Uschold and Jasper (1999), `An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.'

Gruber (1993) defines an ontology as `the specification of conceptualisations, used to help programs and humans share knowledge' . The conceptualisation is the couching of knowledge

about the world in terms of entities (things, the relationships they hold and the constraints between them). The specification is the representation of this conceptualisation in a concrete form. One step in this specification is the encoding of the conceptualisation in a knowledge representation language.

As this definition suggests, ontologies differ from data models in two significant aspects (Peter and Hans, 2003).

i. Ontologies build upon a shared understanding within a community. This understanding represents an agreement over the concepts and their relationships that are present in a domain.
ii. Ontologies use machine processable, logic-based representations that allow computers to manipulate ontologies. This includes transferring ontologies among computers, storing ontologies, checking the consistency of ontologies, reasoning about ontologies etc.

With the support of the ontology, both user and system can communicate with each other by the shared and common understanding of a domain and they serve as reusable vocabularies that can be shared by human as well as computer system.

ONTOLOGY LEARNING

Ontology learning (OL) is an emerging field aimed at assisting a knowledge engineer in ontology construction and semantic page annotation with the help of machine learning (ML) techniques. OL also refers to extracting conceptual knowledge from input and building an ontology from them automatically. It is because manual building of ontologies is a costly and time-consuming, tedious and error-prone task and manually built ontologies are expensive, biased towards their developer, inflexible and specific to the purpose that motivated their construction. Automation of ontology construction not only reduces costs, but also results in an

ontology that better matches its application. In this sense, ontology learning could be defined as the set of methods and techniques used for building ontology from scratching, enriching, or adapting an existing ontology in a semi-automatic fashion using several sources. (Chen & Wu, 2005).

OL was introduced by Madche and Staab where it can be described as the acquisition of a domain model from data (Madche & Staab, 2000). The vision of OL proposed includes a number of complementary disciplines that feed on different types of unstructured, semistructured, and fully structured data to support (semi) automatic and cooperative ontology engineering.

Ontologies formalize the intensional aspects of a domain, whereas the extensional part is provided by a knowledge base that contains assertions about instances of concepts and relations as defined by the ontology. The process of defining and instantiating a knowledge base is referred to as knowledge markup or ontology population, whereas (semi-)automatic support in ontology development is usually referred to as OL (Buitelaar. et al, 2005).

A framework on the OL development consists of information extraction, ontology discovery and ontology organization (Zhou, 2007). In information extraction a variety of data can be exploited in OL either in form of structured or unstructured. This will converts all structured/unstructured data into the forms that can be used for ontology discovery. In particular, text documents are processed via content analysis by employing a variety of natural language processing techniques, ranging from tokenization, to part-of-speech tagging, phrase structure and/or grammatical function parsing, semantic, discourse analyses and also statistical techniques. After that supervised and unsupervised learning algorithms have been applied to discover ontological concept and relations from extracted concept candidate. The filtering techniques will be used in order to get domain – specific concept. Words/phrases that only perform grammatical functions and words that are unlikely to carry domain-specific meanings are filtered out using information retrieval techniques. Ontology organization is deploys to improve the usability of the

discovered knowledge by using several approaches such as clustering, inverse relation, pattern recognition and others.

APPROACHES FOR ONTOLOGY LEARNING

According to survey done by Gómez-Pérez et.al (2005), Maedche & Staab(2000) distinguish different ontology learning approaches focus on the type of input:

i. Ontology learning from text, it based on the use of text corpora. A corpus of text is a set of text that should be representative of the domain (complete), prepared to be processes by a computer and accepted by the domain expert.
ii. Ontology learning from dictionary, it bases its performance on the use of a machine readable dictionary to extract relevant concepts and relations among them.
iii. Ontology learning from knowledge base, it base aims to learn an ontology using as source existing knowledge bases.
iv. Ontology learning from semi-structured schemata where it looks for eliciting an ontology from sources which have any predefined structure, such as XML schemas.
v. Ontology learning from relational schemata. It aims to learn an ontology extracting relevant concepts and relations from knowledge in databases.

This section, we only discuss on the ontology learning from text where it consists of extracting ontology by applying natural language analysis techniques to texts.

ONTOLOGY LEARNING SYSTEMS

Ontology learning refers to extracting ontological elements (as derived in component of ontology below) from input and building an ontology from them automatically. Ontology learning uses methods from a diverse spectrum of field such as natural language processing, machine learning, information retrieval, database management and information/knowledge acquisition. But in this study, we focused on the natural language processing and statistical technique only. For the comparison of the approaches, we do the analysis with the well known OL systems such as TextToOnto (Maedche & Staab, 2001), ASIUM (Faure & Poibeau, 2000), OntoLearn (Navigli & Velardi , 2004) and DODDLE (Sekiuchi et. all, 1998) (refer table 1) using the methods discussed below.

Prerequisite Process MAIN COMPONENT OF AN ONTOLOGY

Cimiano (2006) divided Ontology learning component into six layers of the different subtasks of learning ontology (refer figure 1). In global, the architecture consists, (i) term extraction, (ii) synonym extraction, (iii) extracting concepts, (iv) extracting relation (taxonomic and non-taxonomic relation) and (v) axiom.

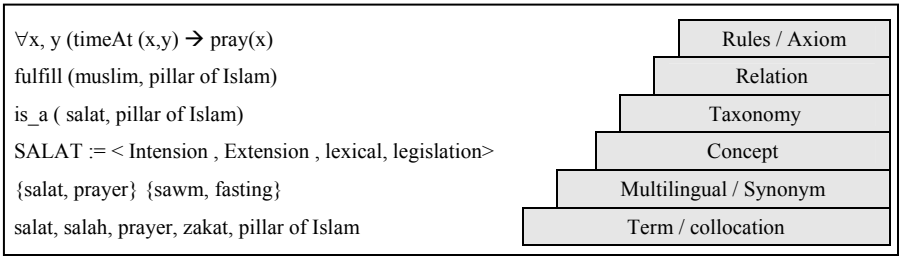

Figure 1 Ontology learning layer cake

The extraction process is preceded by a module for lexical analysis and parsing of the documents.

Lexical Analysis

Lexical analysis is the process of converting a sequence of characters into a sequence of tokens. This process is also known as tokenization. Tokenization can occur at a number of different levels: a text could be broken up into paragraphs, sentences, words, syllables, or phonemes. And for any given level of tokenization, there are many different algorithms for breaking up the text.

There is two level of tokenization. First, for this development we break it up into sentences before proceed to the next process, to parse the sentence.

Second, the tokenizer is a case-sensitive scanner, which generates tokens by identifying delimiters from text. There are two types of delimiters, boundary delimiters and removal delimiters. The boundary delimiters are rules that rules to detect the noun phrase.

The removal delimiters can be any items which are considered not significant for any noun phrases, and will be discarded by the tokenizer. There are four types of removal delimiters:

1. Stopwords: Articles, pronouns, prepositions (excluding of), conjunctions, some adverbs (e.g. what, when, where, and so forth), number words (e.g. one, ten, hundred), unambiguous verbs that can be used as verb only (e.g. be, expect), those past and passive participles of most frequently used irregular verbs (e.g. known, bought, got), and words about times (e.g. minute, hour, week); stopwords do not include those verbs that can be nouns (e.g. can, make).

2. Numbers: Numeric items, such as currency, percentage, and fractions, are removed.
3. Punctuation: Only few exceptions, like hyphen used in compound words, quotes used in possessives or as an and (e.g. rock `n' roll), and periods used in abbreviations, most others are considered removal delimiters.
4. Formating delimiters: Such as table fields, labeled lines, and section heads are considered removal delimiters.

Parsing Technique

This technique also known as a part-of-speech tagging (POS tagging). POS is the process of marking up the words in a text as corresponding to a particular part of speech (noun, verb, participle, article, pronoun, preposition, adverb and conjunction), based on both its definition, as well as its context—i.e., relationship with adjacent and related words in a phrase, sentence, or paragraph. Figure 2 shows a parse produced by Apple Pie Parser (Sekine, 1996).

```
output >>(S (S (NP (NPL The (ADJP most basic) requirement)
(PP of (SS (VP being (NPL a Muslim))))) (VP is (PP to (NP
(NPL publicly state) (NPL the words))))) -DOPENQ- (S (NPL
There) (VP is (NPL no God))) but (S (NPL Allah and
Muhammad) (VP is (NPL His Prophet))) -DCLOSEQ- -COMMA-
(S (SS (PP with (NPL sincerity))) and (SS (PP without (NPL
any reservations)))) -PERIOD-)
```

Figure 2 : Output generate by Apple Pie Parser

Term Extraction

Term extraction is a prerequisite for all aspects of ontology learning from text. Terms are linguistic realizations of domain-specific concepts. The task here is to find a set relevant terms or sign for concept and relation. Techniques that used to determine most relevant phrases as terms can group into:
Linguistic Methods which used rules or pattern over linguistically analyzed text which can be divided to the following families:

a. Simple term patterns: Adjective-Noun (ADJ-N), Noun-Sequence (NSEQ), Noun-Preposition-Noun (NPREP-N) and Proper Name (PN).
b. Syntactic relations: Verb-Object (VB-OBJ) and Subject-Verb (SUBJ-VB).
c. Semantic relations: IsA and HasA.

Statistical Methods which used co-occurrence (collocation) analysis for term extraction within the corpus and comparison of frequencies between domain and general corpora where computer Terminal will be specific to the Computer domain and dining Table will be less specific to the Computer domain

In statistic analysis, the scores used in term extraction are:

i. **TFIDF** – Term Weighting

   The term frequency in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards

longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j.

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

where $n_{i,j}$ is the number of occurrences of the considered term in document d_j, and the denominator is the number of occurrences of all terms in document d_j.

The inverse document frequency is a measure of the general importance of the term

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

With | D | : total number of documents in the corpus; $|\{d_j : t_i \in d_j\}|$: number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). Then

$$tfidf = tf_{i,j} \cdot idf_i$$

A high weight in tfidf is reached by a high term frequency (in the given document) and a low document frequency of

the term in the whole collection of documents; the weights hence tend to filter out common terms.

ii. **χ2 (Chi-square)** – Cooccurrence Analysis & Term Weighting

The vector vfi contains all the words and their frequencies in the document collection i, and is constituted by pairs ($word_j$, $freq_{i,j}$), that is, one word j and the frequency of the word j in the document collection i. We want to construct another vector vxi with pairs ($word_j$, $w_{i,j}$) where wi,j is the χ2 value for the word j in the document collection i

$$w_{i,j} = \begin{cases} \frac{(freq_{i,j} - m_{i,j})}{m_{i,j}} & if\ freq_{i,j} > m_{i,j} \\ 0 & otherwise \end{cases}$$

Equation below defines $m_{i,j}$, the expected mean of word j in document i. When computing the χ2 values, the frequencies in the target document collection are compared with the rest of the document collection, which we call the contrast set. In this case the contrast set is formed by the other word senses.

$$m_{i,j} = \frac{\sum_i freq_{i,j} \sum_j freq_{i,j}}{\sum_{i,j} freq_{i,j}}$$

Where

$$X^2 = \sum \frac{(obs - exp)^2}{exp}$$

iii. **Domain Relevance & Domain Consensus** (Navigli and Velardi, 2004). Considers term distribution within (DC) and between (DR) corpora

According to Navigli and Velardi (2004), they parsed the available documents in the application domain in order to extract a list Tc of syntactically *plausible* terminological such as noun phrases (NP), e.g. compounds (*credit card*), adjective-NP (*local tourist information ofce*), and prepositional-NP (*board of directors*)

The system they built call OntoLearn uses a novel method for filtering "true" terminology. The method is based on two measures, called *Domain Relevance* and *Domain Consensus.*

Domain Relevance can be given according to the amount of information captured within the target corpus with respect to a larger collection of corpora. By given a set of *n* domains $\{D_1 \ldots\ldots D_n\}$ and related corpora, the domain relevance of a term *t* in class D_k is computed as:

$$DR_{t,k} = \frac{P(t \mid D_k)}{\max_{1 \le j \le n} P(t \mid D_j)} \qquad E(P(t \mid D_k)) = \frac{f_{t,k}}{\sum_{t' \in D_k} f_{t',k}}$$

Where $f_{t,k}$ is the frequency of term *t* in the domain D_k

Domain Consensus measures the distributed use of a term in a domain D_k. The distribution of a term *t* in documents $d \in D_k$ can be taken as a stochastic variable estimated throughout all $d \in D_k$.

$$DC_{t,k} = \sum_{d \in D_k} \left(P_t(d) \log \frac{1}{P_t(d)} \right) \quad E(P_t(d_j)) = \frac{f_{t,j}}{\sum_{d_i \in D_k} f_{t,j}}$$

Filtering non-terminological (or non-domain) candidate terms is performed using a combination of the measures (1) and (2).

$$TW_{t,k} = \alpha DR_{t,k} + \beta DC_{t,k}^{norm}$$

where $\beta DC_{t,k}^{norm}$ a normalized entropy and $\alpha,\beta \in (0,1)$.

α Usually chose 0.9 and β depends upon the number N of documents in the training set of D_k. When N is sufficiently large, “good” values are between 0.35 and 0.25

Hybrid Methods

The hybrid methods are combination of linguistic rules to extract term candidates and statistical (pre- or post-) filtering.

SYNONYM EXTRACTION

Synonym extraction is a process of identify terms that share (some) semantic, i.e. Potentially refer to the same concept. It can be synonym (within languages) or it is a translation (between languages) or it can be an orthography of a language. The techniques used for extraction of synonym or translations are classification (example by extending WordNet where with SynSets corresponding to classes) such as Brasethvik and Gulla (2001) and clustering where it's cluster according to similar distribution (Karoui et all, 2006; Pantel, 2003). In very specific domains, some researchers like Navigli and Velardi (2004); Tucarto et. All (2000), have exploited integrated approaches to synonym discovery. Saidah et. all (2008) discovered the transliteration problem between Arabic language and English Language as a synonym to the concept.

EXTRACTING CONCEPTS

It was terms that indicate a concept within the domain population. Concept formation should ideally provide an intension definition of concepts, their extension and the lexical which are used to refer to them (Cimiano, 2006).

According to Corcho and Gomez-Perez (2000) concept can be abstract or concrete, elementary (electron) or composite (atom), real or fictious. In short, a concept can be anything about which something is said, and, therefore, could also be the description of a task, function, action, strategy, reasoning process, etc.

They may be extracted from input or be created during the ontology refinement from other concepts. In other words they may or may not have corresponding elements in the input. In terminological (or term-based) acquisition of concepts, a concept node will be created corresponding to the extracted term which may be natural-language words or phrases, while in conceptual (or semantic-based) concept creation, which is usually done in the refinement phase, the concept will be built according to its features (attributes/values), its functionality and so on and hence may have no corresponding input (no corresponding word or phrase in the input text). Some ontology learning systems use an existing ontology and just populate it by instances of classes and relations. Most of these systems do not learn new concepts (classes) and just learn instances of existing classes (Shamfard & Barforoush, 2003).

Intension is an extraction of a definition for a concept from text. It can be either informal definition or formal definition. For informal definition OntoLearn (Navigli & Velardi, 2004) uses natural language generation to compositionally build up a WordNet gloss for automatically extracted concepts such as 'Integration Strategy' : "strategy for the integration of …". A logical form that defines all formal constraints on class membership can be use to define formal definition. It can be construct by using Inductive Logic Programming or/and Formal Concept Analysis.

Extension is an extraction of Instances for a Concept from Text. It commonly referred to as Ontology Population. Extension can be relates to Knowledge Markup (Semantic Metadata). Instances can be names for objects, e.g. Person, Organization, Country or City and it can be event instances (with participant and property instances), such as Football Match (with Teams, Players, Officials, ...) Disease (with Patient-Name, Symptoms, Date, …) etc.

Named-Entity Recognition and Information Extraction can be use to extraction the extension from the text.

Lexicon can be extract by using Synonyms and translations for a Concept from Text.

Saidah & Naomie (2008) extended the attribute of the concept according to Islamic Knowledge requirement where they add legislation/reference in order to define an idle concept for domain Islamic Knowledge.

EXTRACTING CONCEPTUAL RELATION

The conceptual relation can be taxonomic or non-taxonomic relation.

Conceptual relations are important because they allow to structure information into categories, thus fostering its search and reuse. Further, they allow formulating rules as well as relations in an abstract and concise way, facilitating the development, refinement and reuse of a knowledge-base (Cimiano et. al 2005). According to Shamfard and Barforoush (2003), relations may be studied in two ways:

i. A relation is a node in the ontology, so it is a concept and may be learned like other concepts.
ii. A relation relates two or more concepts and so it should be learned as a subset of a product of n concepts (for $n > 1$).

Taxonomic relations

Taxonomies are widely used to organise ontological knowledge using generalisation/specialisation relationships through which simple/multiple inheritance can be applied (Corcho & Gomez-Perez, 2000). This methods can be develop using clustering

technique. Although some references (Cimiano et. all. 2005) refer to hyponymy and meronymy relations as taxonomic relations (e.g. provided by WordNet), most taxonomic relation learning systems just learn the hyponymy relations (the IS-A hierarchy of concepts). The taxonomic relation also can be extract using pattern based extraction for example using Hearst patterns (1998). The other approach is using syntactic relation expressed by the head of the noun phrase and its subclasses that can be derived from a combination of the head and its modifiers.

Non-taxonomic relations

According to Shamfard and Barforoush (2003) in their , non-taxonomic conceptual relations refer to any relation between concepts except the IS-A relations, such as synonymy, meronymy, antonymy, attribute-of, possession, causality and other relations (learned by systems such as Shamsfard & Barforoush,2002; Maedche & Staab, 2001; Agirre et al., 2000), knowledge about specific words' syntactic categories and thematic roles such as learning subjects and objects of verbs (Cimiano et. all 2005), discovering verb frames (Faure & Poibeau, 2000), classifying adjectives (Assadi, 1997) and nouns (Chalendar & Grau, 2000) and identifying names (Bikel et al., 1999).

AXIOM

Axioms are used to model sentences that are always true. This construct in form of rules. They can be included in ontology for several purposes, such as constraining the information contained in the ontology, verifying its correctness or deducing new information (Farquhar *et al*., 1996).

Axiom is declaratively and rigorously represented knowledge which has to be accepted without proof. In predicate logic case, a formal inference engine is implicitly assumed to exist.

But, one seldom mentions it. Axioms have two roles as follows in ontology description:

- To represent the (partial) meaning of concepts rigorously.
- Within the scope of the knowledge represented declaratively, to answer the questions on the capability of the ontology and things built using the concepts in the ontology.

CONCLUSION

There are varieties of techniques and approaches have been proposed recently either in natural language processing approach or in machine learning approach. In table 1, is a summary of techniques and methods that have been proposed and introduced by researcher based on ontology learning layer that we discuss before. But, that table also summaries the ontology representation, tool associated with the development on ontology also the reusable of other ontology such as WordNet and lexicon database. The most important thing is evaluation or measurement of the quality and the correctness of the ontology where most of the researches are using either domain experts, empirical measure or/and gold standard as their evaluation.

Table 1 : Ontology Learning Layers (Learning From Domain Text)

Organisation/ Researcher	**AIFB**	**ECAI2000**	**University Di Roma**	**Keio University**	**UTM**
Tool or system associated	TextToOnto	ASIUM	OntoLearn	DODDLE	**ISKnow**
Domain Text		Cooking recipe corpus and Pascal Corpus (INIST)	Economy and Tourism	CISG and Business	Islamic Knowledge
Source Data (tagged)	• Free text natural language document from the web	• Domain free text • Semantic domain with thematic units • Annotated text • Primitive concepts from the human	• Online corpora • Documents • glossaries	• Machine readable dictionar (WordNet) • Domain specific text corpus	• Free text natural language document

			expert		(expert input)	
Terms/phrase		Simple nouns, compound nouns, adjective nouns, prepositional nouns, dependency relations	Verbs and their arguments (subject, object) and adjunts(place, time, means)	Simple nouns, compound nouns, adjective nouns, prepositional nouns		Noun phrase Heuristics and pattern based
Term filtering		• TFIDF based pruning • Frequency propogation • Term frequency comparison with generic doc. collection	• Manual assisted by automated means	• Domain Relevance score • Domain consensus		Linguistic and heuristic filter
Synonyms		clusters	Clusters		Similar pairs	Transliteration of the concept, Translation between language
Concept formation	**Units Concept**	Intension, Extension and (Lexical)	Clusters (Intension and Extension)			Intension, Extension, Lexical and Legislation
	Extraction Method	• Tokenzier • morphological analysis • name entities recognition • part-of-speech tagging • chunk parser	• category of nouns • conceptual clustering & induction • shallow natural language processing with frame • POS tagging (contextual and lexical • rules)	• Tokenzier • POS tagging • Phrase boundry detection combined with syntactic info • category of nouns • filtered using natural language processing and statistical techniques • Semantic interpretation		• Tokenzier • morphological analysis • part-of-speech tagging • chunk parser • Pattern
Taxonomy/ non taxonomy		• co-occurrence • hierarchical clustering of concepts • mediator (proposition, verb) • heuristic rules based on the linguistic dependency relations • general association rules by machine learning	selection preferences of verbs (minimum description length with a threshold)	WordNet & linguistic property of compound nouns and a rule-based inductive-learning	Wordnet and support and confident (based on certain threshold)	• Hearst Pattern • linguistic property of compound nouns

Axiom	Non	Non	Non		Non
Ontology representation	XML	• Conceptual hierarchy • Description logic		• Conceptual hierarchy	XML
Tool Associated	• Statistical and Syntactic processing tools • LoPar & Tree Tagger	•	• Statistical and Syntactic processing tools • Tree Tagger		• Statistical and Syntactic processing tools
Ontology Reuse	Yes (lexicon)		Yes (WordNet)	Yes (wordNet)	Al-Quran and Hadith Indexes
Evaluation	Empirical Measure and Gold Standard		Empirical Measure and By Experts	Empirical Measure and By Experts	Empirical Measure, By Experts and Gold Standard

REFERENCES

Agirre, Ansa, Hovy, Martinez: 2000. Enriching very large ontologies using the WWW. In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00, Berlin, Germany, August 20-25,

Assadi, H. 1997, Knowledge Acquisition from Texts: Using an Automatic Clustering Method Based on Noun-Modifier Relationship, Proceedings of 35th Annual Meeting of the Association for Computational Linguistics (ACL'97), Madrid, Spain, 1997.

Bikel, D. A., Schwartz, R., and Weischedel, R., An Algorithm that Learns What's in a Name, Machine Learning, 34, 211-231, 1999.

Brasethvik and Gulla 2001. Natural language analysis for semantic document modeling. Data & Knowledge Engineering. Volume 38, Issue 1 , July 2001, Pages 45-62.

Buitelaar P., Cimiano P., Grobelnik M., Sintek M. (2005). Ontology Learning from Text. Tutorial at ECML/PKDD, Oct. 2005, Porto, Portugal.

Chalendar, G., Grau, B. 2000, SVETLAN' A System to Classify Nouns in Context, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000

Chen E. and Wu G. 2005. Seventh IEEE International Symposium on Multimedia (ISM 2005), 12-14 December 2005, Irvine, CA, USA. IEEE Computer Society 2005.

Cimiano P, Hotho A, Staab S. 2006. Learning Concept Hierarchies from Text Corpora using Formal Concept Anaylsis Journal of Artificial Intelligence Research 24. August 2005.

Cimiano. P. 2006. Ontology Learning and Population from Text: Algorithms, Evaluation and Applications Springer. November 2006.

Corcho Ó., Gómez-Pérez A. 2000. A Roadmap to Ontology Specification Languages. EKAW 2000: 80-96

Farquhar, A., Fikes, R., Rice, J. 1996. The Ontolingua Server: A Tool for Collaborative Ontology Construction, Proceedings of KAW96, Banff, Canada, 1996.

Faure D., and Poibeau, T., 2000. First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

Gómez-Pérez, A., Manzano-Macho, D. 2005: An overview of methods and tools for ontology learning from texts. Knowledge Engineering Review 19 (2005) 187–212

Gruber T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In Formal Ontology in Conceptual Analysis and Knowledge Representation, edited by Nicola Guarino and Roberto Poli, Kluwer Academic Publishers, in press. Substantial revision of presented at the International Workshop on Formal Ontology, March, 1993, Padova, Italy. Available as Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

Karoui, L. Aufaure, M.-A. Bennacer, N. 2006, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)

Maedche A., Staab S.. 2000. Semi-automatic Engineering of Ontologies from Text. In: Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering, Chicago, 2000.

Maedche, A., and Staab, S., 2001, Ontology learning for the Semantic Web, IEEE journal on Intelligent Systems, Vol. 16, No. 2, 72-79, 2001

Navigli R. and Velardi P.. 2004. Learning Domain Ontologies from Document Warehouses and Dedicated Websites. Computational Linguistics, 30(2), 2004.

Pantel P., 2003. Ontology-based Text Clustering. PhD Thesis, University Of Alberta.

Peter M.and Hans A., 2003. Analysis of the State-of-the-Art in Ontology-based Knowledge Management. Semantic Web and Peer-to-Peer Project Report.

Saidah S., Naomie S., 2008, Methodology of Ontology Extraction for Islamic Knowledge Text. Postgraduate Annual Research Seminar 2008, UTM, Skudai.

Saidah S., Naomie S., Nazlia O., 2008. Extracting Keyword and Keyphrase as a Concept for Islamic Knowledge Ontology, IEEE International Symposium on Information Technology 2008, Kuala Lumpur.

Sekiuchi R., Aoki C., Kurematsu M. and Yamaguchi T.: DODDLE: A Domain Ontology Rapid Development Environment, the Fifth Pacific Rim International Conference on Artificial Intelligence, LNAI1531, Springer-Verlag, pp.194-204 (1998)

Shamsfard M., and Barforoush, A. A., (2002) An Introduction to HASTI: An Ontology Learning System, Proceedings of 6th Conference on Artificial Intelligence and Soft Computing (ASC'2002), Banff, Canada, June, 2002.

Shamsfard M., Barforoush A. A, The State of the Art in Ontology Learning: a Framework to Compare , The Knowledge Engineering Review Journal, vol.18, Dec. 2003.

Turcato D., Popowich F., Toole J., Fass D., Nicholson D., and Tisher G. (2000). Adapting a synonym database to specific domains. In ACL'2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval, 2000.

Uschold, M. and Jasper, R. (1999). A framework for understanding and classifying ontology applications. In Benjamins, V., anf A. Gomez-Perez, B. C., Guarino, N., and Uschold, M., editors, Proceedings of the IJCAI-99 workshop on ontologies and Problem-Solving Methods (KRR5), volume 18, pages 1-11, Stockholm, Sweden. CEUR-WS.

Zhou, L. (2007), Ontology Learning: State of the Art and Open Issues, Information Technology and Management, 8(3), 241-252

10
XML AND RELATIONAL DATA

INTEGRATION WITH CMW

Wan Mohd Hafiz Mohd Nasir
Nor Hawaniah Zakaria
Shamsul Sahibuddin

INTRODUCTION

The demand for rapid data integration is getting higher as more and more information sources appear in modern enterprises. Extensible Mark-up Language (XML) is fast becoming the new standard for data representation and exchange on the World Wide Web, e.g., in B2B e-commerce, making it necessary for data analysis tools to handle XML data as well as traditional data formats.

Data integration is a complex activity that involves different levels of data, the data model, the data schema and the instances. Integration of XML data sources requires more than a tool to allow data to be arranged in the same syntax. Adapting to different data model requires the use a common data model to map information from different data sources. Once a common data model has been agreed upon, adapting to the different representation of a same entity is the next issue. For example, two different sources may be using two different names to represent the same thing (e.g. “cost” and “price”) or one same name to represent

two different concepts (Bertino and Ferrari, 2001). The main key to data integration is the use of one language to determine the semantic of data content (Bertino and Ferrari, 2001). The goal is to determine the architecture, models, and standard for semantic description. Several issues that require further investigation are development of a formal base for standard metadata, development of tools and techniques to produce, extract and store metadata, investigation on framework interoperability environment for semantic, development of semantic base tools for knowledge exploration, and development of integration tool for XML.

Metadata management and integration is the number one problem for integration in data warehouse and business intelligence because it requires the use of multiple tools and products, where each tool and product has its own metadata definition and format (Do and Rahm, 2000). Thus, production, sharing and management of metadata for the tools and products require time and can lead to problems. There are several approaches for data integration used today to solve these problems (Minno, 2000). However, all the approaches use their own metadata. Without the use of a standard metadata management, other problems occur, such as data losses that raise the questions of data consistency.

This chapter is discussing an enhance technique for XML data integration into relational data to solve integration problems such as missing data. Integration of distributed data sources is becoming increasingly important as more business relevant data appear on the web, e.g. on B2B marketplaces, and enterprises cooperate more tightly with their partners, creating a need for integrating the information from several enterprises. The data warehousing approach dictates a physical integration of data, mapping data from different information sources into a common multidimensional database schema. This enables fast evaluation of complex queries, but demands great effort in keeping the data warehouse up to date, e.g. when data passes from the sources of the application-oriented operational environment to the data warehouse, inconsistencies and redundancies must be resolved, so

the data warehouse provides an integrated and reconciled view of the data of the organization (Jensen *et al*., 2001a).

BACKGROUND

On-line Analytical Processing (OLAP) is a category of business software tools that enables decision support based on multidimensional analysis of data. OLAP data, typically drawn from physical integration of transactional databases, is organized in multidimensional data models, categorizing data as either measurable facts (measures) or hierarchically organized dimensions characterizing the facts. Features like automatic aggregation (Rafanelli, 1990) and visual querying (Thomsen, 1997) supported by OLAP tools ease the process of decision support compared to traditional DBMSs (Lenz, 1997).

XML is a meta language used to describe the structure and content of documents. XML, although originally a document mark-up language, is increasingly used for data exchange on the Web. The application of XML as a standard exchange format for data available on the Web makes it attractive to use in conjunction with OLAP tools. Previous approaches for integrating web-based data, particularly in XML format, have focused almost exclusively on data integration at the logical level of the data mode, creating a need for techniques that are usable at the conceptual level which is more suitable for use by system designers and end users. The most wide-spread conceptual model is the Unified Modeling Language (UML) (OMG, 2001a).

In current data warehouse environment there is either no or only insufficient support for a consistent and comprehensive metadata management. Typically, a multitude of largely autonomous and heterogeneously organized repositories coexist. Do and Rahm (2000) categorize the major metadata types and their interdependencies within a three-dimensional classification approach and then investigated how interoperability and integration of metadata can be achieved based on a federated

metadata architecture and standardization efforts such as Common Warehouse Metamodel (CWM). They also examined synchronization alternatives to keep replicated metadata consistent and gave an overview of currently available commercial repositories and discussed interoperability issues to couple data warehouse with information portals.

The need for data movement and data integration solutions is driven by the fact that data is everywhere underneath business applications. The same applies for metadata; metadata is also underneath the data and object modelling tools, as well as within the repositories of the ETL, Data Warehouse, Enterprise Application Integration, and Business Intelligence development tools. An adequate metadata management should be focused on storing all the metadata in one central repository to avoid redundancy and keep the metadata consistent. In this study, we proposed architecture for XML based data and metadata integration in data warehousing system with CWM as a standard for modelling and exchanging metadata.

METADATA

Metadata is becoming more and more important in areas like data warehousing, knowledge management, enterprise application integration and e-business (Agosta, 2001). Although the concept of metadata is not new at all there is still no single, accepted definition but the little useful ‘metadata is data about data’. In order to have a sound foundation for further argumentation, we will provide a definition of metadata in the context of data warehousing.

For the context of data warehousing we define metadata as data that answers questions about all the object data in a data warehouse, transformations of object data and underlying data flows, and finally the technical and conceptual system architecture. Referring to this definition, with the help o metadata a user should be able to locate the proper object data for this task as well to

understand and interpret usage, meaning, sources, creation, structure, quality, and topically of the object data he is dealing with (Auth and Eitil, 2002).

In a DW system almost every software component produces and consumes metadata. For example the system tables of the central warehouse database store the description of the warehouse data model, which is actually metadata. To make further use of this metadata a software structure for this purpose containing interfaces, and data store and access components must be implemented. Furthermore, metadata is consumed and produces by the whole range of the data warehouse users starting from developers to end users (Auth and Eitil, 2002).

CWM FOR METADATA INTERCHANGE

CWM is a standard for describing technical and business metadata occurring from data warehousing and business intelligence. CWM is hosted by industry consortium Object Management Group (OMG) (OMG, 2001a). Although the main purpose of CWM is designed for metadata interchange between different tools and repositories it can also be used for building active objects models for storing and maintaining metadata (Poole, 2000). CWM is founded on the UML metamodel and extends it with specific meta-classes and meta-relationships for modelling data lineages found in the warehousing domain. Thus, it provides a complete specification of syntax and semantics necessary or interchanging shared metadata.

META INTEGRATION

Meta Integration Technology, Inc. is a Silicon Valley, California based software vendor specialized in tools for the integration and

management of metadata across tools from multiple vendors, and multiple purposes including data and object modelling tools, data Extraction, Transformation, and Load (ETL) tools, Business Intelligence (BI) tools, and so on. The need for data movement and data integration solutions is driven by the fact that data is everywhere underneath business applications. The same applies for metadata: metadata is also everywhere underneath the data and object modelling tools, as well as within the repositories of the ETL, DW, Enterprise Application Integration, and BI development tools (Bremeau, 2001).

Meta Integration Works (MIW) is a complete metadata management solution with sophisticated functionalities such as the Model Browser, the Model Bridges, the Model Comparator, the Model Integrator, and the Model Mapper all integrated around a powerful metadata version and configuration management.

Meta Integration Model Bridge (MIMB) is a utility for legacy model migration and metadata integration. MIMB also operates as an add-in integrated inside popular modelling, ETL and BI tools. With over 40 bridges, MIMB is the most complete metadata movement solution on the market. MIMB supports most popular standards and the market leading tool vendors.

XML DATA AND METADATA INTEGRATION ARCHITECTURE: JENSEN MODEL

Jensen *et al.* (2001a) introduced an architecture for XML and relational data integration as shown in Figure 1. Based on the figure, Jensen's contribution is to allow OLAP to directly analyzed the XML data using existing OLAP tools. It is done through logical integration. One of the methods used to obtain information is to retrieve the whole XML document from the web. The weakness of this method is that it could take quite a long time. The other method is to consider the XML document logical structure

represented by DTD. DTD retrieval time is less because of it smaller size compared to XML document. Potential problem that may happen is in the form of complexity that can make it difficult for the designer to understand the XML data structure. According to *Jensen et al.* (2001a), what is needed here is a way to communicate the structure in a simple and quick manner to the designer.

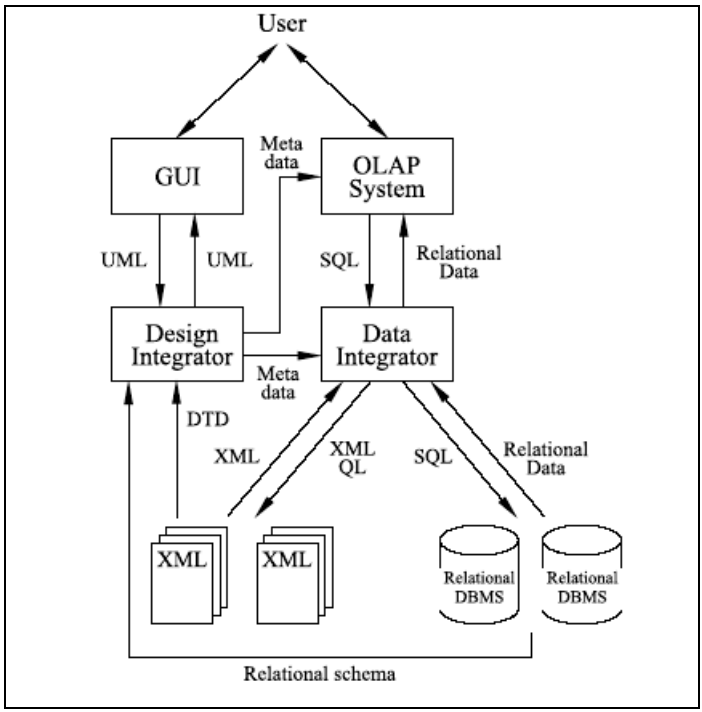

Figure 1 Jensen Architecture (Jensen *et al.*,2001a)

To obtain the information, DTD is transformed into a Unified Modelling Language (UML) diagram. The graphic visualization of the XML document structure will make it easy for

the designer to understand compared to the context-free grammar of DTD. *Jensen et al.* (2001) assumed that the XML document logical structure defined by DTD and the XML document is valid. XML schema can also be used to defined XML document and is more powerful, but DTD is used because of the recommendation from W3C.

Two type of data source is integrated with XML data disguising as relational data. The entity centre in Jensen *et al.* (2001a) architecture is the Data Integrator that allows XML data and/or relational data to be used by OLAP tools. The middle step needed when building the relational structure is the construction of Snowflake UML diagram by the designer using the GUI and generated using Design Integrator. The Snowflake UML diagrams are transformed into the relational structure using Data Integrator and thus make it ready for the OLAP system.

Users will make the query to the OLAP system using OLAP query statement such as MultiDimensional DN_XSS_NEUTRALIZE_eXpression (MDX) used by Microsoft SQL Serer 2000 Analysis Services. Data Integrator will test the SQL query received from the OLAP system and transformed it into a series of queries. The series of queries is in the form of XML or SQL queries, depending on the source of data, whether the source is XML or relational data.

One of the problem in determining the attributes information type is that XML 1.0 is a type-less language, where no data type declaration exist in DTD and also in XML document (W3C, 2001). When the data is received, issues such as missing data, wrong data, non-existent of reference integrity, or server that is not functioning, can occur. When retrieving data, data element can be lost or be duplicated. The designer in Jensen et al. (2001a) model will have to work harder in order to ascertain that missing data is eliminated or reduced by putting base value in the element. The problem is, is the designer able to make certain that all data element contains base value in the presence of large number of XML document.

The problem with mistaken data is also similar with missing data because the value of wrong data can be judged to be the same as the value of missing data. The designer need to solve this problem whether by using fixed value, ignoring data element, or terminating the process. Another problem in Jensen *et al.*(2001a) model is that the server is not functioning during access to the XML data on the web. This research is focused on the aspect of missing data because it is difficult for the designer to fix or set the value of each attribute. The research will look at the aspect of missing data from the Jensen *et al.* (2001a) perspective in order to compare the results of missing data using the improved architecture proposed in this research.

XML DATA AND METADATA INTEGRATION ARCHITECTURE WITH CWM

Our method proposed is to apply CWM for metadata interchange and metadata management that incorporates a common shared metamodel for metadata syntax and semantics.

There are several types of data sources that need to be integrated such as relational data and XML data. In this chapter, the focus is on the way XML data can be integrated. A solution is proposed for XML data movement from XML DTD to the data warehouse and CWM. Previous research did not concentrate on how to integrate metadata that cannot be extracted from DWH components but resides in various other sources such as from e-business application. This chapter focused on how to integrate XML documents and XML DTD as a metadata in the data warehouse. This architecture used a central metadata repository that implements a common meta-model based on OMG's CMW and serves as a hub for metadata interchange. Metadata and meta-models are exchanged between CWM repository and local metadata from data warehouse stores utilizing XML as a standard interchange.

Figure 2 illustrates our proposed general architecture for data and metadata integration which emphasis on the XML document and relational data in the data warehouse environment. The main difference of this architecture with other architectures that involves data migration and transformation is the use of the metadata management and transformation, i.e. the Common Warehouse Metamodel (CWM).

There are two data sources used in the architecture represented in Figure 2, which are the relational database and the XML data. The metadata used from the XML data source is the Document Type Definition (DTD) file since it describes the structure of the content of the XML file, while the relational database uses the ERwin data model. The ERwin data model is transformed through the reengineering process using a bridge as transformation tool. In this architecture the Meta Integration Model Bridge (MIMB) acts as the bridge to perform the reengineering process of data model or metadata.

The Meta Integration Works (MIW) is used as the tool in the loading, mapping and transformation processes of the metadata and data model for the whole integration process. The DTD metadata and CWM model data are loaded into MIW for these processes.

XML DTD metadata file is used as the XML file model involved in the data movement because it describes the structure of the file. The DTD file are loaded into MIW as the model where the XML elements are presented as the classes, XML attributes as the attributes and the relational primary key or foreign key represent the tree structure (parents and child). With the models loaded into MIW, the process of comparison, mapping, and transformation model are then executed.

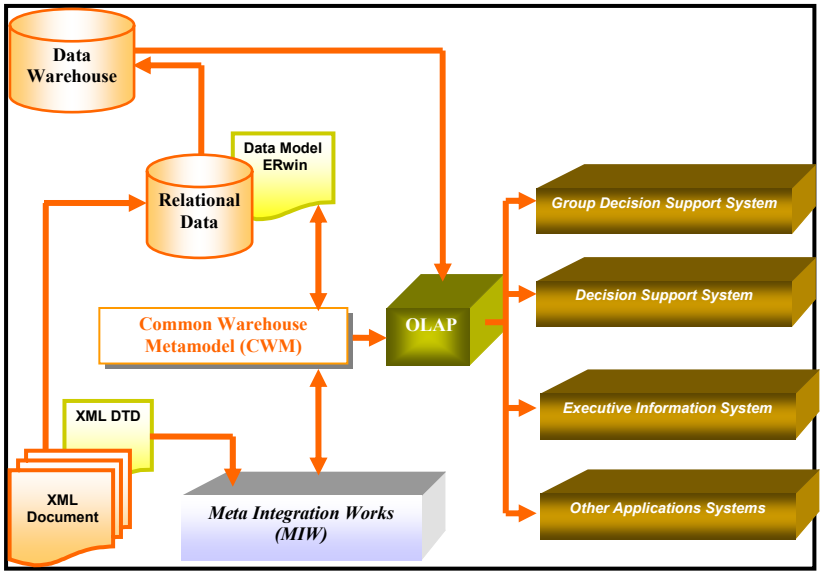

Figure 2 Metadata and Data Integration Architecture With CWM

EXPERIMENT AND ANALYSIS

An experiment was conducted to examine the effectiveness of our proposed architecture. We compared our proposed architecture with the architecture of Jensen's, where in our proposed architecture, the integration considers CWM as the standard, while in Jensen's, the integration does not consider CWM as a standard in the integration process. We tested our integration architecture by first developing a system that performed the data and metadata integration and data migration process. Screenshots of our system are shown in Figure 3 and Figure 4.

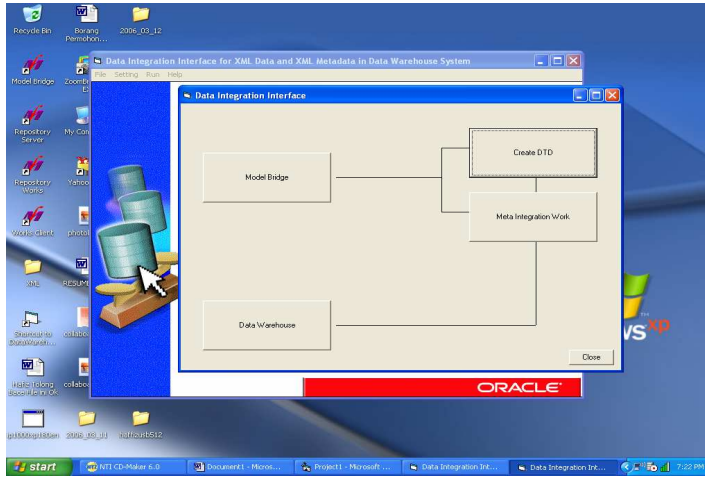

Figure 3 Data and Metadata Integration System Interface

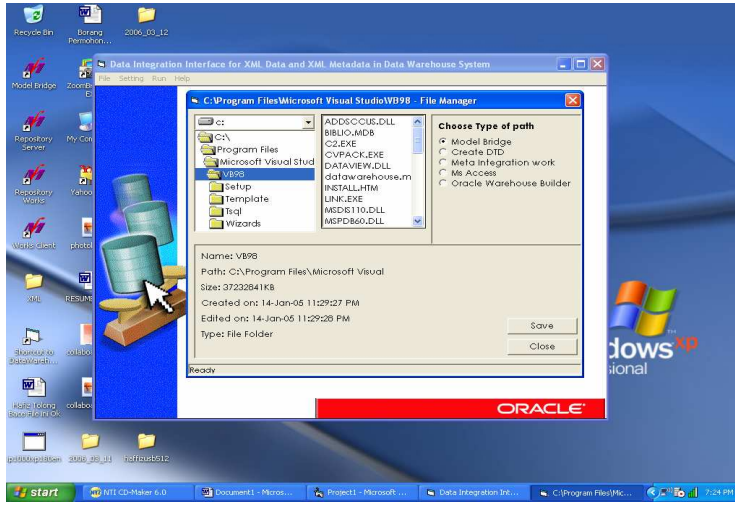

Figure 4 Data Integration Interface Setting Manager Setup

Next, the results from the data and metadata integration and data migration are tested to determine its correctness. This is done by running several queries and observing possibility of data losses occurence as shown in the result of the queries executed. We prepared several sets of data and queries based on two case studies; the first case study was e-Business and the second was Human Resource. The number of data used in both case studies was 1000.

For the first case study, the results of our experiment are shown in Table 1 for integration without CWM and in Table 2 for integration with CWM (i.e. our proposed approach). Both reflect the data loss frequency from the execution of four queries (P1, P2, P3, and P4) on five sets of data (Data1, Data2, Data3, Data4, and Data5).

Table 1 Data Losses For Integration Without CWM

Queries	Data 1	Data 2	Data 3	Data 4	Data 5	Average
P1	2	4	4	6	8	4.8
P2	0	4	2	3	4	2.6
P3	0	1	1	1	1	0.8
P4	4	4	6	8	8	6

Table 2 Data Losses For Integration With CWM

Queries	Data 1	Data 2	Data 3	Data 4	Data 5	Average
P1	0	0	0	0	0	**0**
P2	1	0	0	1	0	**0.4**
P3	0	0	0	0	0	**0**
P4	0	0	0	0	0	**0**
Average	**0.25**	**0**	**0**	**0.25**	**0**	**0.1**

Both Table 1 and 2 show a significant difference in data losses, where it is shown that data losses is reduced significantly for the case of integration with CWM. The percentage data losses reduction can be seen in Table 3. The overall percentage data loss reduction is shown in Table 4 for both case studies used in our experiment. This reduction in data losses is contributed by the fact that both metadata from the source and destinationare are sharing the same syntax because CWM is also written based on XML syntax. The matching process performed before integration improvement requires several transformation on the DTD itself. This syntax and semantic map reduces data loss when CWM metadata is used in the integration.

Table 3 Comparison on Data Losses Percentage

Data Losses	P1	P2	P3	P4
Without CWM	4.8	2.6	0.8	6
With CWM	0	0.4	0	0
Percent Decrease	**4.80%**	**2.20%**	**0.80%**	**6.00%**

Table 4 Overall Percent Decreases For Case Study 1 and 2

Data Losses	Case Study I	Case Study II	Overall
Without CWM	14.2	5.4	19.6
With CWM	0.4	0.4	0.8
Percent Decreases	**13.80%**	**5.00%**	**18.80%**

CONCLUSION

Our proposed architecture, which includes CWM as one of the component, has managed to reduce significantly the data losses problem when integration of data and metadata involving different format compared to previous architectures that uses individual and un-uniformed metadata. This is because by using CWM, the problem of different data semantic, syntax, and attribute is resolved by matching the different data models. When these differences are resolved in the two different data models that are to be integrated, the percentage of data losses can be reduced or avoided totally.

REFERENCES

Agosta, L. (2001). Reports of the demise of metadata are premature. DM Review 3.

Auth, G. dan Eitil, V.M. (2002). A Software Architecture for XML-based Metadata Interchange in Data Warehouse Systems.

Bertino E. dan Ferrari E. (2001). XML and Data Integration. IEEE Internet Computing.

Bremeau C. (2001). XML Data Movement Components for Teradata. www.metaintegration.com, July 2004.

Do, H. H., Rahm, E. (2000). On metadata interoperability in data warehouses. Technical Report 1-2000, Institute Informatics, University Leipzig.

Lenz, H., (1997). Summarizability in OLAP and Statistical Databases, *Proceedings of the Ninth International Conference on Statistical and Scientific Database Management*, 39-48.

Jensen, M. R., Møller, T.H., dan Pedersen, T.B. (2001a). Specifying OLAP Cubes On XML Data. *Tech Report R-01-*

5003, Department Of Computer Science, Aalborg University.

OMG(2001a) OMG: OMG Specifications.

URL:http://www.omg.org/technology/documents/specifications.htm. 22/09/1003.

Poole, J. (2000). The Common Warehouse Metamodel as a Foundation for Active Object Models in the Data Warehouse Environment. Position to ECOOP 2000 workshop on Metadata and Active Object-Model Pattern Mining – Cannes, France.

Rafanelli, M. (1990). STORM: A Statistical Object Representation Model, *Proceedings of the Fifth Conference on Statistical and Scientific Database Management*, p14-29.

Thomsen, E., (1997). *OLAP Solutions: Building Multidimensional Information Systems*, John Wiley & Sons, Inc.

INDEX

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X