Text Summarization System with Bayesian Theorem on Oil & Gas Drilling Topic

by

Iwan Kurniawan

Dissertation in partial fulfillment of
the requirement for the
Bachelor of Technology (Hons)
(Information & Communication Technology)

JULY 2007

Universiti Teknologi PETRONAS
Bandar Sri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

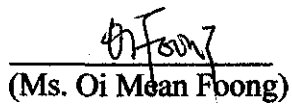
Text Summarization System with Bayesian Theorem on Oil & Gas Drilling Topic

by

Iwan Kurniawan

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,



(Ms. Oi Mean Foong)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own as specified in the references and acknowledgement, and that he original work contained herein have not been undertaken or done by unspecified sources or persons.

IWAN KURNIAWAN

ABSTRACT

Text summarization is the process of identifying the important sentences or words from the article which later to be represented and combined to generate the summary. There exist numerous algorithms to address the need for text summarization including Support Vector Machine, k-nearest neighbor classifier, and decision trees.

In this project, Bayes theorem algorithm is studied and experimented by the implementation of a textual summarizer. This algorithm is used to extract the important points from a lengthy document, by which it classifies each word in the document under its relevant probability of the word's likeliness to be included in the summary given the corpus containing the summary done by the experts as the initial probability. As the application is used and processed, it would learn and keep track of the probability of each keyword so that it would predict the chance of certain keywords to be included in the future summarization.

The objectives of this project are to look at the current situation in the area of text summarization research, to study the statistical approach in automatic text summary generation, and then to create a simple sample of text summarization tool which takes into account the existing research.

Since the area of the application is specific, which is on oil and gas drilling topic, the ready-used corpus on that area is not easy to find. The articles collected are from the journals, news and any other information sources which are related to the discussed topic. Evaluation of the application is carried out against another accompanying system-generated summarizer which is already in the market. Human-made summary are used as the ideal or reference summary in evaluating both performance; the Text Summarization system and the Word Auto Summarizer. Current results show that the Text Summarization system performs better than the Word Auto Summarizer at the compression rate 60% and 70% (2/3 of the articles' length) by 11.31% and 10.80% respectively. Optimum value for overall performance is 85.82%.

ACKNOWLEDGEMENT

First and foremost, I would like to pay my respects to Allah the Almighty for making me capable of undergoing this project and for showering His blessings upon me throughout the execution of the project and especially at times of need.

Thanks to Ms. Vivian Yong Suet Peng, my earlier supervisor, who has provided me with guidance on the basic understanding on the beginning of the project. I really wish you best of luck for your undergoing research and study in New Zealand. The cooperation, help, and support provided by Ms. Oi Mean Foong were a great asset to have during moments of confusion and I truly acknowledge that.

Many thanks have to go to the related academic staffs of UTP; Dr. Chow Weng Sum, Dr. Sony Kurniawan, Ms. Zullina, Mr. Teguh Adji and many other lecturers for their caring hands in contributing to the development, finishing and evaluation phase of the project. Thanks must go also to my country mate master students of Petroleum Engineering, Geosciences and Geology; Mr. Pebriyanto, Mr. Adhi and Ms. Tiulfa for their valuable contributions.

I would like to thank my colleague Nastassja Hong Fang Ying, also for her sharing moments in understanding the different aspects of data mining. I would like to thank my family and close friends for their supports and whose love and care enabled me to ease my mind and soul towards the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

The companies nowadays find out that the data they are producing keep increasing in number. Those data do not always correspond to information since they need to be processed first in order to generate a knowledge or information. As a result, information readers or consumers are bombarded with more facts from more sources than they are capable of taking in.

To encounter this issue, the more information should be made more digestible. There are ways to make it happen. The long information can be made in a shorter form, compressed into a briefer format, so that the users are able to absorb it quickly. It can merely point the user to a fuller account if he or she is interested. It can tip the user to whether a piece of information will be worthwhile. Or, it can pull similar or related sources together into a single summation.

Doing the process manually by having human beings doing the reducing is time-consuming and expensive. However, with the help of current technology, especially Artificial Intelligence, machines or computers have been designed and built and programs are written so that automatic summarization is possible.

Two fundamental approaches are identified to automatic text summarization. These methods represent the endpoints of a continuum. The results of both approaches are in compression of text, with one result is relatively, while the other is deep and complex [5].

Text extraction, the least complex end, creates the summaries by using terms, phrases and sentences pulled directly from the source text using statistical analysis at a surface level. The frequencies of word written are counted and analyzed based on their

occurrences, where they appear and reappear in the source text. This is sometimes referred to as "knowledge-poor" processing and is rooted in the term-weighting algorithms of information retrieval (IR).

On the other hand, more complex and "knowledge-rich" endpoint is summarization through abstracting. The objectives of text abstraction are to have the computer-generated analysis and synthesis of the source text into a completely new text. The new text should be shorter but is still cohesive and intelligible. It should also fulfill the specific information needed by the users. This process is sometimes known as machine understanding, a multidisciplinary endeavor involving information retrieval, linguistics and artificial intelligence.

1.2 Problem Statement

The rapid improvement of technology has made the text summarization an important task to do automatically by the computer. Text summarization process needs to handle, organize as well as analyze the source text in order to deliver the summarized ones.

The ability to analyze and understand huge amount of data of text documents is a challenge across many disciplines. For example, we were given a large data sets of emails, news, articles, technical research or any other important documents, and we want to absorb and understand those information contained in the document fast and accurately. With the limited time, it is a need to do the process as fast as possible.

There is increasing interest in text mining techniques to solve these types of problem. There are also various methods to employ text categorization such as neural networks [1], regression models [17] and decision trees [8]. However, these methods have their own setback which contributes to its poor development of classifiers due to performance variation using different types of data collection [15]. Therefore, numerous researches have been done to further enhance these methods in order to better suit and increase the performance of text categorization.

By having the same amount of time, but with more information we have, text summarization enables us to digest the same amount of information but at a reduced effort. It is when the text summarization comes into its realm. The reasons for the text summarization become more popular are [13]:

1. The user may know which documents to read, which document would provide them the needed information.
2. The user may revise the already-read documents quickly.
3. The user may seek for second opinion on the document or source.

By seeing the backgrounds above that the text summarization offers, we may conclude that the more information we have, the more the need to reduce this information into smaller manageable chunks. These chunks are smaller in size and may reflect how relevant the full information to our needs is.

1.3 Objectives

The objectives of this project are to look at the current situation in the area of text summarization research, to study the statistical approach in automatic text summary generation, and then to create a simple sample of text summarization tool which takes into account the existing research.

1.4 Scope of Study

For this particular project, the text summarization will be able to accommodate documents in English, analyze text files, and perform qualitative measure on the text documents. The type of material which will be summarized (the source document) within this study will be in plain text file without any multi-media material. The source document excludes any tables, graphs or pictorial information. The domain of text will be focused on oil and gas drilling articles.

CHAPTER 2

LITERATURE REVIEW OR THEORY

2.1 Definitions of Automatic Text Summarization

According to Luhn and Salton, automatic text summarization is a technique where a computer automatically creates a summary of one or more text. The initial interest in automatic shortening of text was spawned during the 1960s in American research libraries. During this era, a large amount of scientific papers and books were to be digitally stored and made searchable, but storage capacity was limited in those days. Therefore, in order to cover the issue, only summaries were stored, indexes and made searchable. When there was no ready-made summary of a publication was available, one had to be created. So, basic techniques in summarizing the text were developed and refined [2] [9] [13].

Text summarization would generate the summary of a given text document automatically. Depending on the approach and end-objective of summarization of documents, text summarization being generated would also diverge. For example, it could be indicative of what a certain topic is about, or can be informative about specific niceties of the same. It can differ in being a "generalized summary" of a document as opposed to "query-specific summary". It may be a set of sentences carefully chosen from the document or can be created by synthesizing new sentences on behalf of the information in the papers.

The summary may be categorized by any of the following criteria [11]:

1. Detail: Indicative or Informative
2. Granularity: Specific Events or Overview
3. Technique: Extraction or Abstraction
4. Content: Generalized or Query-based

At the present time, the concern of text summarization has changed. If in the early days, it was needed to save the storage space, currently, it was designed more on to retrieve the data faster and more effectively [6].

Large amount of digital data are available. So in order to avoid from being drowning in it, the data must be filtered and extracted and finally summarized so that one can still accept the required information and not to walk around the whole bunch of data. We have to bare in mind that data are not essentially in correspondence with information. They have to be processed first in order to create information (knowledge).

The overflow of textual information is especially clear on the internet, but also within large companies, government bodies and other organizations. The Internet has come to be of much use mainly because of the support given by Information Retrieval (IR) tools. However with the rapid growth of the information on the Internet, a second level of abstraction of information from the results of the first round of IR becomes obligatory. That is, the great number of documents returned by IR system need to be summarized.

Currently this is the major application of summarization. The many other uses of summarization are almost noticeable: Information extraction, as against document-retrieval, automatic generation of comparison charts, Just-In-Time knowledge acquisition, finding answers to specific questions, a tool for information retrieval in various languages, biographical profiling, etc.

2.2 Approaches to Text Summarization

2.2.1 Abstractive Vs Extractive Methods

The abstracts are different from one person to another and may vary in terms of style, language and detail since abstraction by human is a complex process of modeling information. The process is complex to be mathematically or logically formulated [7]. Some of the tools which considered natural language processing methods have been developed in the last decades to generate abstractions. They extract phrases and lexical

chain from the document, and then later fuse them to be combined to form a summary (abstraction). Sentences from the original input could also be taken and presented in the extractive summary. The approach is called extraction.

Referring to Mani [10], an abstract is a summary where at least some of the material do not appear in the input, whereas, an extract is a summary when it consists entirely of material from the input.

Major problems of the two text summarization methods have been addressed by Ganapathiraju in his paper [4], "Relevance of Cluster size in MMR based Summarizer". Extractive method tends to produce longer than average length of summary sentences. It consumes space since unnecessary portion of the sentences are as well included. Extractive summaries could not capture the fact that important or relevant information is spread across sentences unless the summary is long enough to hold those sentences. Inaccuracy presentation of conflicting information may occur.

On the other side, abstractive summaries suffer from users preferences on extractive summaries over it [3]. The main reason is because the summary presented by extractive method is the as-is information by the author. Sentences synthesis is unavailable yet. Users could read between the lines from the extractive summaries. Hence, incoherent sentences could be produced by the automatic machine generated summaries which occurs only at the border of two sentences.

2.2.2 Surface-Level Approach

In this surface level approach, the system calculates the features of the articles statistically to decide which important key points should be included in the summary [14]. Surface-Level has been the basis of many summarization researches. Some features that could be statistically calculated in surface-level approach are:

1. Location – the position of the terms or sentences in the paragraph or in the whole document.

2. Keywords – the terms occurrences in the articles that could lead to thematic meaning of the whole document [10]. It uses the frequencies of the words in the articles as the key value to determine the importance level of the words. Problem occurs when the documents of the same type are al taken at the same time. The terms which appear too frequently during the period of time may not be worthwhile as the salience measurement of the summary.
3. Heading – the words that appear in the heading or even title of the document are considered having the thematic meaning for the whole document.
4. Cue words and phrases – the pertinence or redundancy of surrounding words and phrases could be seen by determining certain phrases in language.

2.2.3 Entity-Level Approach

Involvement of internal representation for the text plays the role in entity-level approach. Modeling the text entities together with their relationships are done in this approach [14]. This approach tries to represent patterns of terms connectivity to show the importance. The following features exist:

1. Term similarity
2. Word occurrences in common contexts
3. Text unit proximities
4. Logical relations, such as agreement and contradictions
5. Words thesaural relationships

2.3 Graph Theoretic Representation

Finding or identifying the main points addressed to the document is the first process in any text summarization [4]. The availability of the themes identification methods was made possible through passages graph theoretic representation. Sentences are represented as nodes in an undirected graph. This could be achieved, of course, after the preprocessing steps; stop word removal and stemming have been done. A node always exists for every sentence. If two sentences share some common words, their similarity (cosine, or such) are above some threshold, they would be connected with an edge. The visualization of words graph theoretic representation could be further explained by Figure 2.1.

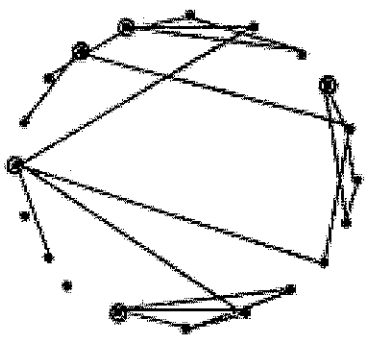


Figure 2.1: Graph Theoretic Representation [4]

A word in the document is represented by a node, based on Figure 2.1. The relationships between words are represented by the edge that connects those nodes. If the similarity between two words is above some threshold, then the edge exists. Important words of the document are the highlighted nodes in the graph.

Two results could be concluded from the representation. The sub-graphs which are unconnected to the other sub-graphs (partition), form distinct topics covered in the documents. It enables the choice of summary coverage. Sentences only from the pertinent sub-graph would be better done for query-specific summary. As for generic summaries, each of the sub-graphs should be analyzed and the representative sentences should be chosen from them.

The important sentences identification in the document is also the result yielded by the graph theoretic representation. High cardinality nodes, the ones with higher number of edges connected to them, are considered the important sentences in the partition. Hence, they carry higher likeliness to be included in the summary. It reflects that those sentences share information with many other sentences in the documents. Visualization of inter-and intra-document similarity could easily adopt the graph theoretic method.

2.4 Key Approaches to Statistical Processing

Within text summarization history, it faced many achievements as well as challenges, ups and downs in terms of research. Text Summarization slowed down considerably in the 1970s and 1980s because the researchers were focusing on the more readily solvable problems. Automatic indexing seemed to be more intense from the investigation area. In the 1990s, IR (information Retrieval) methods could be solved, especially after the improvement on web technology [16].

Other currently promising area including statistical analysis of term clustering, statistically based analysis of text structure, or discourse analysis and training algorithms that use human-generated summaries were added to the term-proximity research building on Edmundson. The human-generated summaries were to determine probabilities that certain source-text sentences should be included in the summary. Altogether, each of the approaches represents a point on the undergoing process towards the full-text understanding.

Some of the key approaches were illustrated in the following recent initiatives examples.

2.4.1 From Automatic Indexing to "On-the-Fly" Summarization.

Gerard Salton of Cornell University and others in 1970s and 1980s [16] evolved the automatic indexing research into statistical processing methods based on the *tf.idf* weighting. It applied significance to a term by counting the occurrences it appears in a

document (*tf*, term frequency) and multiplying the result by the logarithmic calculation of the total number of document in the collection divided by the number of documents containing the target term - inverse document frequency (*idf*). Salton used tf.idf weights to identify the closely related segment within a document, together with other measures derived form indexing research. Comparison on those relationships with those of other documents was next processed in generating automatic hyperlinks when the similarities were close.

Therefore, relatedness could be revealed from the analysis of similarities among various combinations of paragraphs within a document. Relatedness is where a topic is reinforced elsewhere in the document. Un-relatedness suggests the beginning of a new topic or angle. The internal links or relationships enable the overall text structure to be derived without the need of complex linguistic theory. Moreover, the links can be compared to a query and extracted summary constructed at retrieval time (on the fly summarization), depending on user's specific information need. Salton found that measuring the amount of overlap between source documents and abstract, the two were nearly identical. However, it should be noted that Salton used Information Retrieval graduate students, not professional abstract writers, in producing the summaries.

2.4.2 Machine "Learning" from Bayesian Statistics.

Julian Kupiec in 1995 with his partners [16] employed an analysis technique which enables the learning progress of the application by probability recalculation in the area of machine training, now as Bayesian statistics. The probability of the likeness for a sentence to be included in the summary would be calculated based on the frequency of text features. Various categories in matching the source text and summary, including *direct match*, where summary sentence and source sentence are identical and *direct join*, where two source sentences are grouped to form a single summary sentence, contribute in learning analysis. As many as 84 percent of machine summaries overlapped with sentences in the manual summaries at a 25 percent compression rate of the source text, according to the test of Bayesian algorithm. It was double the overlap that Edmundson cited at the same compression rate. Bayesian approach was a language-independent

since there was a follow up with Korean text. The optimal set of features for Kupiec; location, cue phrase and sentence length determined the performance.

2.4.3 Topic-Rich Keywords Point to Useful Sentences.

Eduard Hovy and Chin-Yew Lin in 1996 [16] dealt with best location discovery to pick out worthy sentences to be included in the summary by using an existing concept thesaurus to provide rudimentary sentences interpretation through topic-identification routine. In developing topic-rich keywords from training collection of 13,000 newspaper articles they used *tf.idf* (term frequency and inverse document frequency). They developed ranked lists of sentences which have topical terms. News stories have a more predictable structure as compared to other document types. They normally have the important information at the beginning of the article. This may vary on different editing publications practices.

The title (headline) is the optimal place to locate usable terms in technology stories. First sentence of the second paragraph follows. Study on 30,000 general-interest *Wall Street Journal* articles shows that title was optimal, followed by the first paragraph. Journalists, on technology stories, tended to tease a new product announcement in the initial sentences. It is done in with abstract language to reserve the facts for the second paragraph. Different editing standards, in *Wall Street Journals*, resulted in the salient facts being included in the first paragraph.

2.5 Evaluation Criteria

Three criteria for a text summarization could be addressed according to Yang and Chute [17]; important information retention, summary readability and summary compression rate.

Identifying which parts of original text are more important than the rest is crucial. Luhn's technique [17] at word level uses the keywords identification. The keywords could be proper names, or more frequent in the text in the language or average.

Keywords identification is not the only feature, titles and the first lines of paragraphs have higher information values relatively to human than other parts of the text. Specific and special cue words like "in conclusion" states the important section of the text.

Readability of a coherent and cohesive text is another criterion for a good summary [17]. Referring to the Figure 2.2 below, it requires valid semantic links between sentences through pronouns and other markers.

(1) Nick Richman bought General Computhings yesterday.
(2) Many investors want to diversify their portfolios.
(3) Richman sold off Special Stupithings.
(4) He used this cash to pay for his new acquisition.

Figure 2.2: Sentences on Coherent and Cohesive Issue [17]

Incohesive summary could be resulted if sentence (4) is not preceded by a sentence providing the antecedents' for *he, this* and *his*. Incoherent summary could be resulted if sentence (4) is not preceded by the information on the cash and acquisition. Incoherent and incohesive relation is more likely to happen in a more condensed text. Identifying and maintaining the semantic chains are research area which people look up to.

Compression rate shows the relative size of a summary, the percentage of words number in the original text which are left out in the summary. Selecting clauses rather than sentences could be an optimization of a summary size. Linguistic processing is required in analyzing less important relative clauses, appositions or conjuncts to be excluded while others maybe joined by aggregation.

2.6 Issues in Summarization

Some issues regarding the automatic text summarization contribute to the development of the research itself. They (temporal ordering, algorithms and evaluation) provide challenges and problems to the text summarization research.

Chronological order of information must be maintained in a temporal ordering issue. It requires temporal normalization, for example, the word today, next week, Monday, etc., means different date in different articles.

Clustering methods algorithm must be carefully designed around the data sparseness problem. Clustering algorithm often fails due to the high dimensionality and data sparseness in forming meaningful passages or documents clusters [12].

Text summarization evaluation needs to be considered. No standard methodologies apply in evaluating the system since human judgment in evaluating the summary is way subjective which may differ from one another.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Previous chapters have discussed the definition of automatic text summarization, as well as the previous works which have been done and researched on the field. Also, in the literature review chapter, approaches to automatic summarization have been identified; graph theoretic representation has also been discussed in a nutshell. Statistical processing towards text summarization was introduced earlier. Some issues which need to be considered when developing Text Summarization System application were identified. This chapter is focusing on the method which is going to be used for conducting the project.

3.2 Planning

So many applications exist nowadays which involve processing data, especially textual classification such as search engines and text summaries. Most of customers of those applications have limited knowledge on how the applications do the process. The processes going on at the back of the machine (system) which are behind the interface are usually transparent. The intended purpose of this study is to focus on the effectiveness of data mining algorithms, particularly on the probability and statistical techniques to be applied in generating summary of a plain text.

The algorithm used in processing the textual data would be evaluated in terms of the efficiency and performance computation by using a particular testing data. To ensure its effectiveness, the summary generated by the system would be compared with other application which is already available in the market. This comparison process will prove how this algorithm performs.

3.2.1 Aims of Text Summarization System

The objectives of this project are to look at the current situation in the area of text summarization research, to study the statistical approach in automatic text summary generation, and then to create a simple sample of text summarization tool which takes into account the existing research.

The project needs to implement artificial intelligent concept in order to get more precise and convincing summary generated from the text given. Probability and statistical techniques need to be involved in this project. The evaluation of the resulted summary needs to be done in order to know how effective the methods used in generating summaries are. Design section would discuss more on the feature and structure of these programs in detail.

3.2.2 Requirements

The requirements of the system or project delivered are based on workable and satisfactory summarization tool. Therefore, the focus points of the development are on the correct (working) algorithm and auto-generated summary. The summary is comparable to human-generated one, and with a satisfactory evaluation method. A pleasing or user friendly interface with functionalities is another supporting key of the system. The followings are the requirements of the system:

1. The system can read individual text documents
2. The system can statistically analyze the source document
3. The system can produce a generic summary
4. The user can select the summary length being produced
5. The interaction between the user and the system is through Graphical User Interface

3.3 Analysis

Analysis of the steps being involved in the Text Summarization System is very important. Fully understanding of the concept implemented would definitely contribute to the success of the project. The concept of semantic and statistical approach which would be implemented within the project should be fully understood. The project has done the studying on the concept of text summarization and the algorithm meant to be used within the system provided by the past researches. Data and information gathering through internet on the subject was carried out.

Current technology provides various tools to be used for text analysis, some of which are open source software. Information sharing through internet has made it possible to search of the tools and to know how they work. Examples of the tools or resources which would help this project are Protégé tool to develop semantic network, WordNet which provides words bank, as well as Java tool which provides many functions in doing the project.

Many approaches and researches have been done in the text summarization area. Most of the researches were using statistical and linguistic features to rank sentences in the article. An approach of text summarization based on semantic content of the sentence and the relative importance of the content to the semantics of the contents is getting famous [10].

3.4 Design

3.4.1 Design of Text Summarization System

Figure 3.1 below shows the general programs' design and the relationships among them diagrammatically. There are five stages to the overall extraction system engine which is shown in the diagram below accompanied by the explanation of each element in the system architecture.

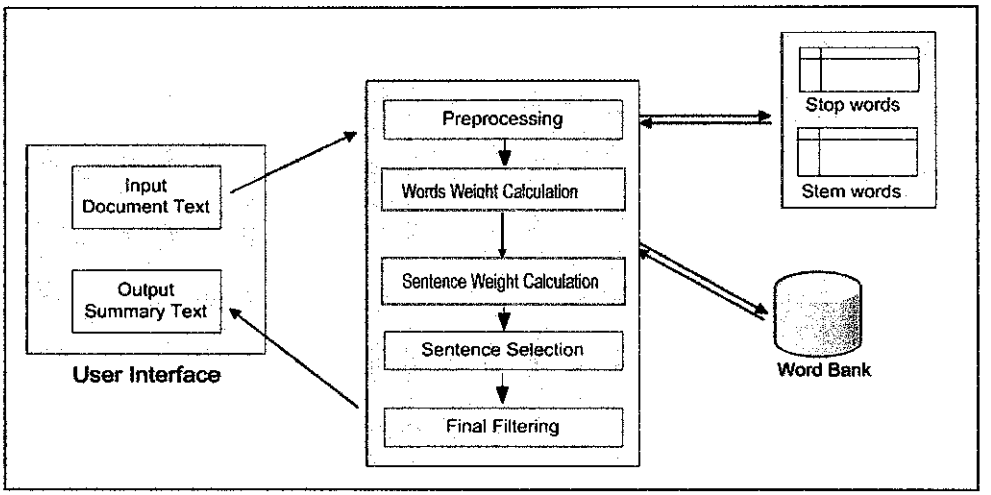


Figure 3.1: System Architecture of the Text Summarization System

The users of the application would load the input article, the one they want to summarize, into the text area provided by the system (as shown in the system architecture in Figure 3.1). As for this specific project, the system would only accept plain text (.txt) with no tabular data representation as well as calculation. When the summarize button is pressed, the whole document will go through the system engine which consists of five stages; namely preprocessing, words weight calculation, sentence weight calculation, sentence selection and final filtering.

Preprocessing stage would break the document into sentence then words. Applying word tagging in the text is essential to pick the correct meaning of the word. Removing stop words and identifying stem words would be processed during this stage. The words and sentence are stored in separate structures. This is done to make the process of learning algorithm of each word easier.

Developing a Bayesian network which is focusing on oil and gas topic would be processed after the preprocessing stage finished. The intention of having Bayesian network in the system is to provide knowledge base on oil and gas to the system. Therefore, the summary resulted later is hoped to be precise and understandable.

Another process within the Bayesian network process is synset ranking stage. It is done basically to rank the sysnsets based on their relevance to the text. Therefore, if a lot of

words in the text correspond to the same synset, it means that the synset or meaning is more relevant to the text, and thus, it must get a higher rank.

The last stage of the algorithm in generating the summary of the text is final filtering. This stage involves the application of simple heuristics to filter out the sentences which have undefined references. Removing sentences which contains words like "He" at the beginning and which begin with quotes are applied.

3.4.2 Database

Database is very important as part of the project. The database would contain a lot of data related to oil and gas topics. The Text Summarization System system going to be developed is trying to focus on specific topic to be fully explored. Oil and gas topic is chosen since it would be very beneficial for oil and gas industry to have Text Summarization System application. Also, by having specific topics to focus on, the development of the database would be more effective and simpler.

The database used by the system contains is represented in a tabular format with rows and columns. The database contains words that are categorized under different parts of speech including verb, noun by which each category has its own columns. Synonym sets data structure would also be included in the database for comparison. The data presented in the database would only be in English text.

3.4.3 Stop Word List

The stop word list is a list of terms to be excluded from the consideration in generating the summary. These words do not contribute to understanding the main idea present in the text. Examples of stop words are 'him', 'her', 'the', and 'it'. Omitting the stop words is an important point when doing the keywords frequency count technique, since the stop words would result in bias towards words which bring little benefit to the whole article's main idea.

However, identifying and removing stop word list is better to be done after we have broken down the sentences into words and found the co-mentioned (co-occurring) words because sometimes, the words which are under the list occur most frequently with a specific word which identifies its importance in the text.

3.4.4 Graphical User Interface

The Graphical User Interface is very important in most of the applications. It supports the users to communicate and do the process of the application in an easier way. The GUI designed for this system is intended to be as user friendly as possible. By having GUI applied to the system, it hides the complexity of the algorithms running behind the machine so that the users are able to deal with the application with fewer problems.

Among the elements needed are text areas to display the original text document, the summary of the document and if possible the details of both input and output documents. A button is needed in order to represent the command of executing the summarization of the given input text document. The menu bar of the application is needed to navigate through the process and / or access some functions of the application.

The layout of the user interface is crucial as it will cater for the ease of the usage or its user friendliness. Human Computer Interaction knowledge should be implemented in designing the layout of the user interface. The user interface is where the documents to be processed are loaded. After doing all the processing, the system will display the generated summary onto the display screen right next to the original document. This will help in comparison process by the users by having both original and generated summary side by side. **Figure 3.2** shows the actual planned user interface for the system.

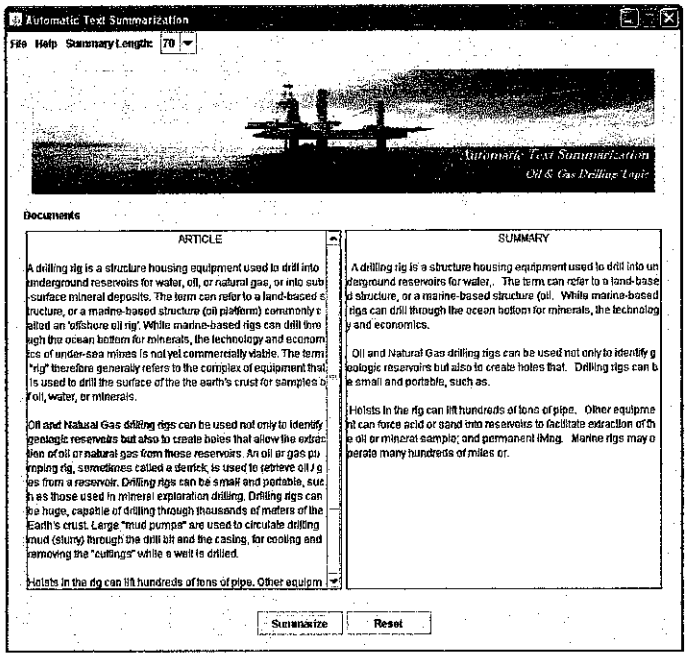


Figure 3.2: The GUI of Text Summarization System

The menu bar provides File menu, Help menu, as well as Summary Length option for the user to choose the compression rate of they summary. The File menu includes the functions in locating and opening the input article which is on text type file. Also, the functions to save or print the summary and to close the application are available under File menu.

The Help menu provides the options for the users to see the information about the application project. The most important part under Help menu is the AutoTextSumm Help which provides the guidelines on how to use the application. Statistics about the input article is also available for the users to choose. It displays the necessary facts on the input article such as the longest and shortest sentence, the number of words in the article and the most important sentence of the article calculated by the application.

The user may select their preference in the summary length (also referred as compression rate) to be processed by the application. The summary length would be dealing with the number of sentences in the input articles with the number of sentences to be displayed in the generated summary.

In addition, the menu bar; File menu, Help menu and the Summary Length, is supported by the tooltips which describe what the menu would function. Tooltips provide helpful information for the users in knowing what they are doing. The figures below show the complete views on the application GUI.

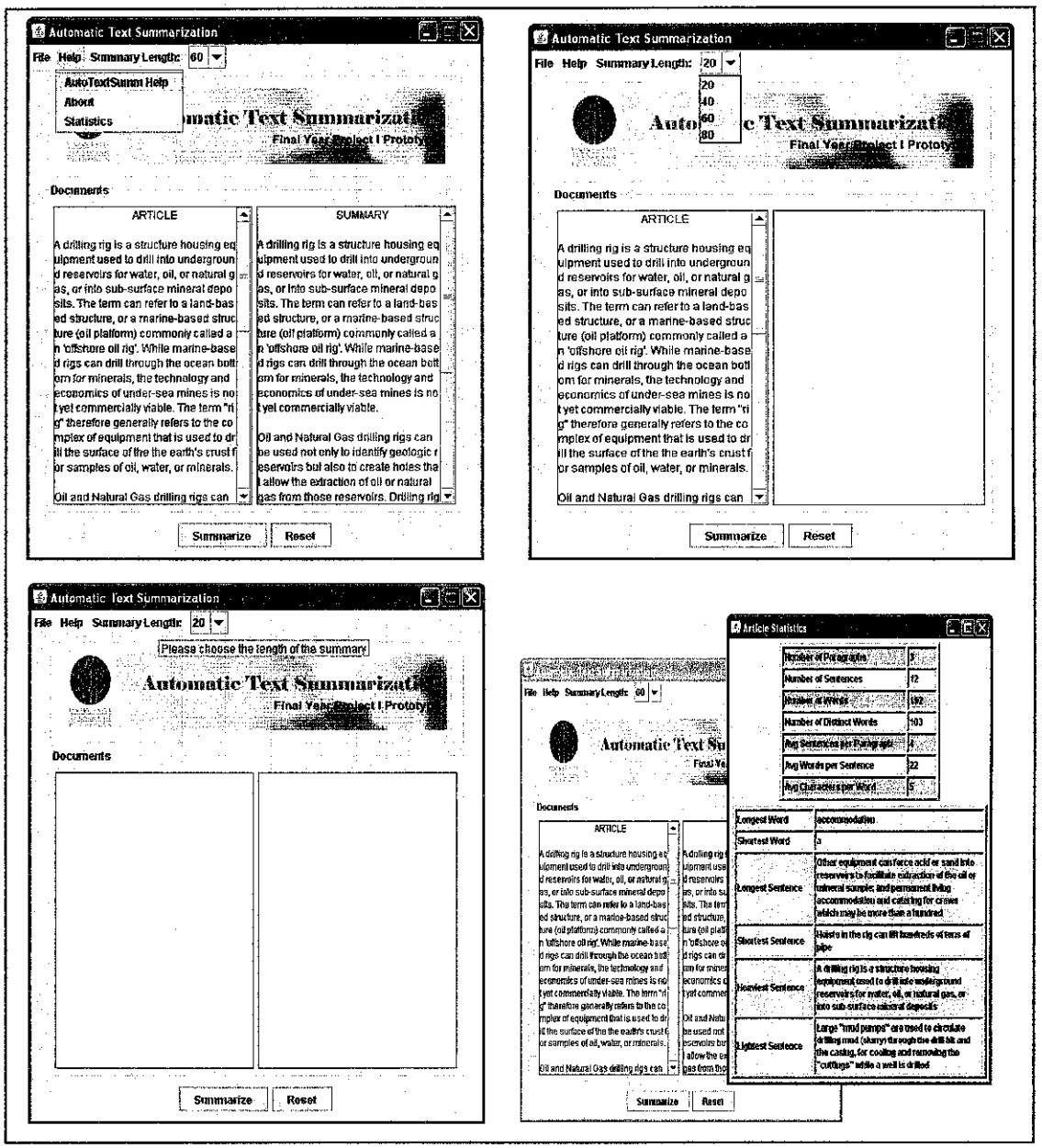


Figure 3.3: The Features GUI of Text Summarization System

The appendix section provides more detail snapshots of the application's graphical user interface.

3.5 Implementation

The following section is the algorithm of the Text Summarization System. The program basically consists of five parts; preprocessing of the text, words weigh calculation, sentence weight calculation, sentence selection and final filtering, which are explained in the section below.

3.5.1 Preprocessing of the Text

Preprocessing of the loaded text is intended to break down text into sentences and later into words. It then removes unnecessary objects from the datasets such as initial white space and store words individually as collection. By referring to Figure 3.3, we may know the algorithm for the preprocessing stage.

```
Initialize stop word list, stem word list
Initialize constant variables with probability values
Define collections
Initialize variables
Load source document files, read into buffer
      Contents of buffer changed to lowercase characters
Skip initial white space
      Stop words removed
If a real word
      Text split into sentences collection
      Tex split into words collection
If new paragraph
      Increment number of paragraphs
Calculate document's sentences count
      Total added to total real words in source
```

Figure 3.4: Preprocessing Stage Algorithm

a. Tokenization

The preprocessing phase of the system starts by taking input document (article which needs to be summarized) into buffer reading. The application, before tokenizing the input article into words, must identify the paragraphs and then the sentences of the input articles. It should be able to calculate how many paragraphs, how many sentences in each paragraph, and how many sentences in the whole article. The purpose is to make the later process, which is sentence weight calculation, easier. Also, to be able to cope with the compression rate chosen by the users.

The buffer will then execute and process the input document by separating the input into words. The list of the words from the input document would then be stored in a specific file. This file contains all words (also known as token) without repetition together with the number of occurrences in the article. So, the first step of preprocessing is then known as tokenization.

b. Stem Word Process

The purpose of having stem word process is to have only one root word which is written more than once in different types of format. Different format of word leads to different meaning. However, the root word is the same. Stem word process is also done to avoid having huge amount of repetitive word banks.

For example, the word 'go', 'goes' and 'going' are having one root word which is 'go'. In other words, stem word process is dealing with prefixes and also suffixes of each word. The input of stem word process is taken from the file produced by tokenization process. The stem word process would execute each word and keep the root word in a new file. This new file would contain only 1 occurrence for each word. In the case where the same root word is found, only the first occurrence would be stored.

c. Stop Word Process

After storing all the words (tokens) into a specific file and the stemming process, the next step in preprocessing phase is executing stop word list. This process would take each token (word) from the file produced earlier to be analyzed if it is categorized as a stop word. The process would import another file which contains all the words which fall under stop word list. Stop words are the words which appear frequently in any articles or situation but contribute less meaning in identifying the important content of the article.

Each token would be given an attribute of 'stop word' if it falls under this category. The reason of giving the attribute is to give rank later on to each word. Words which are stop words would have less point as compared to those which are not. The reason of giving point is to rate the importance of each word in contributing the meaning to generate the summary.

Beside the attribute of stop word for each word, tokenization and stop word process would also calculate the frequency of each word within the document. This occurrence rate is very important as it is most likely to contribute in determining the importance of each word to the article.

3.5.2 Bayesian Theorem

A Bayesian network is a directed acyclic graph whose arcs denote a direct causal influence between parent nodes (causes) and children nodes (effects). The nodes can be used to encode any random variable. For example, a person can be ill or well; the car engine can be working normally or having problems, etc.

The intention of having Bayesian network in the system is to provide knowledge base on oil and gas to the system. Therefore, the summary resulted later is hoped to be precise and understandable.

a. Word Weight Calculation

After tokenizing the input article into words, we now have the collection of words appear in the article. The next step is to assign specific weight to each of the words being captured. As discussed earlier, we categorized the words into either stop words, keywords or unknown words (not in the two categories).

In assigning the weight for each word, we consider the type of word (stop word, keyword or unknown word) as well as the corpus that we have. The word weight based on the corpus would be calculated by the following formula:

$$Cw = tf * itf$$

Equation 3.1

Where tf is the Term Frequency; the number of occurrence a specific word appears in the article. The itf is the Inverse Term Frequency; the Log base of number of articles in the corpus (article database) divided by the number of article in the corpus in which the term exists. The following pseudo code would explain how the word weight calculation is done.

```
For each Word in the word list
        Get wordFrequency
        If Word is "stopword"
                Assign wordWeight to 0.1
        Else if Word is "keyword"
                Set wordWeight to 0.1
                For each Article in the corpus
                        If Article contains Word
                                Increment numbOfArticle
                Calculate wordCorpusWeight as
                        [wordFrequency * log(Corpus size/numbOfArticle)/log 2]
                Add wordCorpusWeight to wordWeight
                Update wordWeight based on Synset
        Else
                Word is know as "unknown word"
                Assign wordWeight to 0.3
```

Figure 3.5: Word Weight Calculation Algorithm

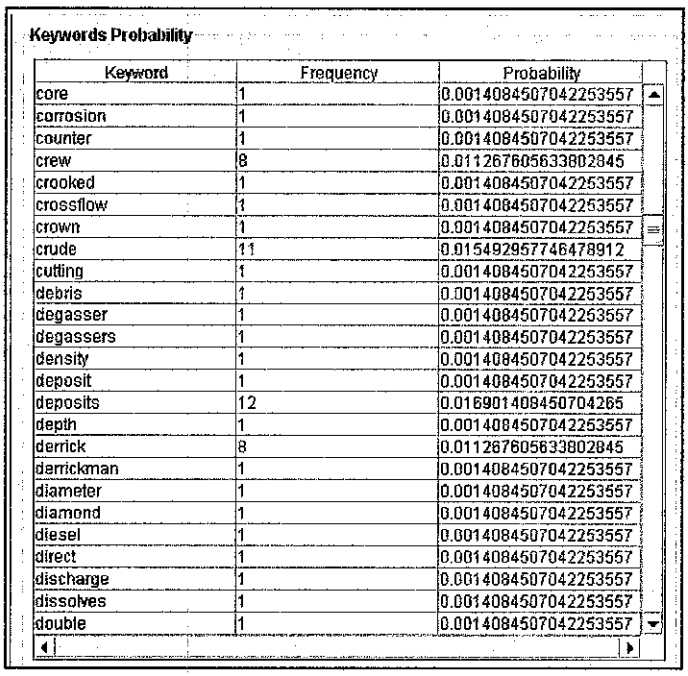
The objective of having synsets ranking is to rank the synsets based on their relevance to the text. If there are so many words are connected to one particular synset, and then the synset is more relevant to the text, then it is given a higher rank. The following figures show the pseudo code and the sample of keywords probability in the Bayes theorem for the application.

```
For each Word in the keyword list
      For each Summary in the summCorpus
            If Summary contains Word
                  Increment numOfSummary
      Calculate wordOccurrence as [numOfSummary/summCorpus size]

For each word in the keyword list
      Add wordOccurrence to totalWordOccurrence

For each word in the keyword list
      Calculate wordProbability as
[wordOccurrence/totalWordOccurrence]
```

Figure 3.6: Word's Probability Calculation Algorithm



Keywords Probability

Keyword	Frequency	Probability
core	1	0.0014084507042253557
corrosion	1	0.0014084507042253557
counter	1	0.0014084507042253557
crew	8	0.011267605633802845
crooked	1	0.0014084507042253557
crossflow	1	0.0014084507042253557
crown	1	0.0014084507042253557
crude	11	0.015492957746478912
cutting	1	0.0014084507042253557
debris	1	0.0014084507042253557
degasser	1	0.0014084507042253557
degassers	1	0.0014084507042253557
density	1	0.0014084507042253557
deposit	1	0.0014084507042253557
deposits	12	0.016901408450704265
depth	1	0.0014084507042253557
derrick	8	0.011267605633802845
derrickman	1	0.0014084507042253557
diameter	1	0.0014084507042253557
diamond	1	0.0014084507042253557
diesel	1	0.0014084507042253557
direct	1	0.0014084507042253557
discharge	1	0.0014084507042253557
dissolves	1	0.0014084507042253557
double	1	0.0014084507042253557

Figure 3.7: Keywords Probability Table

b. Sentence Weight Calculation

After assigning weight to each of the words in the article, the next step is to assign and calculate the weight for each sentence of the input article. The sentence weight is calculated based on the words' weights, as well as based on the position of the sentence.

The sentence weight based on the words would be calculated by adding up the weights of the words that form the sentence divided by the number of the words in that sentence.

Besides, the position of the sentence would also influence the importance level of the sentence to the article content. The first sentence of the article is definitely important to be included in the summary. The first two sentences of each paragraph are also considered important, which lead them to be given higher probability to be included in the summary. Lastly, the last sentence of the article would usually be the conclusion of what the article talks about. By assigning the weight for each of the sentence in the article, we could use it for the next step; sentence selection.

Figure 3.8 in the next page shows the algorithm (pseudo code) of how the sentences' weight calculation is done by considering the words' weights, sentences' position as well as Bayes theorem.

3.5.3 Sentences Selection

So far, we have dealt with the sentence weight calculation. It leads us to the list of the sentences with their entrance reference (appear orderly in the article) and their weights. From here, we could rank them based on the weight. The higher the weight, the more relevant the sentence to the content of the article is.

In ordering the sentences based on the weight, we should keep the entrance number. This process is required, because although the sentences are seen based on their weights, but in displaying the summary, the system should display the sentences based on the entrance.

The compression rate, chosen by the users earlier, would be used as in determining how many sentences to be displayed in the summary. Later, those selected sentences must be displayed according to their order in appearing in the article.

```
For each Sentence in the article
      /*to calculate sentence's weight based on word's weight*/
      For each Word in the Sentence
            Add wordWeight to sentenceWordWeight
            Increment totalWords
      Calculate sentenceWordWeight as [sentenceWordWeight by
totalWords]
      Assign sentenceWordWeight to sentenceCurrentWeight

      /*to calculate sentence's weight based on sentence's
position*/
      For each Paragraph in the article
            Get sentenceCurrentWeight
            If Sentence is 1st two sentences in the Paragraph
                  Calculate sentencePositionWeight as [sentenceCurrentWeight
* 0.5]
If Sentence is last sentence in the article
      Calculate sentencePositionWeight as [sentenceCurrentWeight
* 0.2]

      /*to calculate sentence's weight based on Bayes theorem*/
      For each Word in the sentence
            If Word is "keyword"
                  Add wordProbability to sentenceBayesWeight

      /*to calculate sentence's final weight*/
      Add sentenceCurrentWeight to sentenceFinalWeight
      Add sentencePositionWeight to sentenceFinalWeight
      Add sentenceBayesWeight to sentenceFinalweight
```

Figure 3.8: Sentence's Weight Calculation Algorithm

3.5.4 Final Filtering

After gaining all of the sentences in order based on their relevance to the text, the last stage of the process is to filter those sentences. The final filtering of the sentences would apply to words which do not have defined references. In other words, sentences containing word with undefined references would be filtered out. Removing sentences which contains words like "He", "It", etc at the beginning and removing sentence which begin with quotes would take place. Natural Language Processing (NLP), if time allows, should be implemented in shortening the selected sentences. The technique of NLP in the application should consider the rules which make sentences. The word tagger (categorizing word into noun, verb, adjective, article and / or preposition) should be defined in the application to be later used.

3.6 Tools

3.6.1 Software

JAVA is chosen as the developing language since it is widely used nowadays and the system implemented is intended to be improved by other developer (open source). JAVA also supports the creation of the user interface where the user and the system in communicating through. Microsoft XP is used as the platform for the friendliness reason. To have the application modifiable, UNIX or LINUX platform are better options for an open source application.

Text processing would have the words in particular article to be represented independently, where each of the words could be accessed and given the weight without interrupting or influencing other words objects. Therefore, having the words in the form of rows and columns (matrix representation) in a database would benefit the process. For the time being, the database chosen to represent the words in table format (rows and columns) is Microsoft Access.

Other supporting tools and resources which contribute in helping this project would be Protégé in developing the ontology of the knowledge and WordNet in providing the words meanings.

3.6.2 Hardware

Text Summarization System is an independent application, where only one computer is sufficient to have it run and produce the summary. Therefore, no internet connection is required. Since it is running on a single computer, then the database is also stored in the same computer. The more words stored (information/knowledge) in the database, the better the summary would be. Below are the specifications of the computer:

1. Pentium III processor
2. 512 MB RAM
3. 40GB hard disk space

CHAPTER 4

RESULTS AND DISCUSSION

This section shows and discusses the findings related to the project from the very beginning up to its completion phase. It is not an obvious task to evaluate the quality of summarization.

4.1 Evaluation

Because of the objective of the evaluation process is to actually evaluate the effectiveness of the Text Summarization System system in generating summaries, some measurements to rate against itself are needed. The reference summary, which would be used to rate against the system generated summary, are taken from an existing summarizer which is already in the market, the Microsoft Word AutoSummarizer. Also, the human made summary would contribute in comparing the summaries. The AutoSummarizer and human (expert) would act as a comparison of a summary in an auto-generated format; one which has already evolved and enhanced from time to time.

The "gold-standard" reference summary is taken into account in evaluating all of the summaries (Text Summarization System, Word Auto Summarizer and human-generated ones), to find out the overlapping of sentences appear in both generated summary and the reference summaries. The method used in evaluating the output summary is called an intrinsic method which aims to evaluate the quality of the summaries as compared to other summaries or extracts.

Some computational results, which are achieved by testing common articles (datasets), would be used to represent the application evaluation. Finally, these results would be compared to some baseline summarization procedures or reference summary (manually generated summaries by experts). It would give the qualitative measurements and shows how well the application performs. The experts involved in generating and evaluating the summary would be from the respected area; petroleum engineering

department. Besides, expertise on language, English lectures, humanities department, would also be joining the evaluation process.

In calculating the overall performance of the Text Summarization System, the following information should be considered:

1. The reference summaries' selected sentences (gold-standards)
2. The Text Summarization System's selected sentences
3. The AutoSummarizer's selected sentences
4. The overlap between the Text Summarization System's summary and the reference summaries
5. The overlap between the AutoSummarizer summary and the human generated summary

The criteria which are taken into consideration for the evaluation process would be further discussed by the section below.

4.1.1 Performance Measure

The performance measures used for the evaluation of the summary generated by the application are precision, recall, F-score [6]. Precision measures the percentage of correctness for the total number of summaries judged by the summary assessor to be relevant. Precision also measures the usefulness of the summarizer while recall is a measure of the completeness of the summarizer.

Recall is a measure of how effective the system in including relevant sentences in the summary. It is 100% when all relevant sentences are retrieved. Precision is a measure of how effective the system in excluding irrelevant sentences from the summary. It is 100% when all documents returned to the system's users are relevant to the summary. Meanwhile, F-Score is a composite score that combines the precision and recall measures. The formula 4.1 shows the mathematical distribution of those measures.

$$Precision = \frac{|\{Relevant\ sentences\} \cap \{Retrieved\ sentencs\}|}{|\{Retrieved\ sentences\}|}$$

$$Recall = \frac{|\{Relevant\ sentences\} \cap \{Retrieved\ sentences\}|}{|\{Relevant\ sentences\}|}$$

$$F - Score = \frac{2\ x\ Precision\ x\ Recall}{(Precision + Recall)}$$

4.1.2 Compression Rate

Compression rate measures length of a summary elative to the length of the original full text and is derived from the equation

$$C = \frac{Ns}{Nft}$$

Where C is the compression rate, N_s is the number of sentences in generated summary and N_{ft} is the number of sentences in the original full text. Different degree of compression rate is used as a factor in assessing performance of individual systems. Each application will have to produce summaries with percentage of 10, 20, 30, 40, 50, 60, 70, 80 and 90.

4.1.3 Existing Text Summarizer

The evaluation is done against the currently available application which is the MS Word AutoSummarizer. It is integrated in Microsoft Word which cuts words by counting words and ranking sentences. The most common words are identified; each sentence is given a score based on the frequency of the words in the document, and finally calculates the average score by dividing the total value of the sentence by the number of words within it. The top scoring sentences are compiled to become the summary of related compression rate chosen by the users.

4.1.4 Human-generated Summary

To obtain the results of all performance measures, a reference output should be at hand. This section of evaluation uses a human-generated summary. The individuals involved in this process are the experts in Petroleum Engineering and the experts in English Language. The summary generated by experts would be used as a reference in obtaining the number of relevant sentences in a particular summary. However, the summary generated by the experts is very subjective and produces different results.

4.1.5 ROUGE (Recall Oriented Understudy for Gisting Evaluation)

ROUGE is a widely used evaluation package for text summarization. It has been used by many researchers in order to cut down on testing time. Basically, ROUGE would compare two generated summaries produced by different application. ROUGE will have the precision, recall and the F-Score of both applications as the output.

4.2 Results

The summaries generated by both the Text Summarization System and the Word AutoSummarizer were obtained by summing the sentences for all of the summaries and comparing them with the human-generated summary, which acts as the reference summarizer. All sentences of the generated summaries from both applications were conducted in exactly the same way.

4.2.1 Tabular Data

Evaluation on: Article 6						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.8733	0.8433	0.858038	0.7992	0.8134	0.806237
20	0.7647	0.6907	0.725819	0.7272	0.7234	0.725295
30	0.80003	0.7357	0.766518	0.7154	0.6879	0.701381
40	0.7955	0.7143	0.752716	0.6942	0.7247	0.709122
50	0.7597	0.7141	0.736195	0.6667	0.7008	0.683325
60	0.8443	0.7462	0.792225	0.6967	0.72307	0.70964
70	0.8041	0.7448	0.773315	0.6566	0.6799	0.668047
80	0.7293	0.6458	0.685015	0.6563	0.626	0.640792
90	0.8009	0.6239	0.701406	0.7216	0.5944	0.651853

Table 4.1: The average precision, recall, and F-Score for Text Summarization System and Word Auto Summarizer using Article 6

4.2.2 Graphical Representation

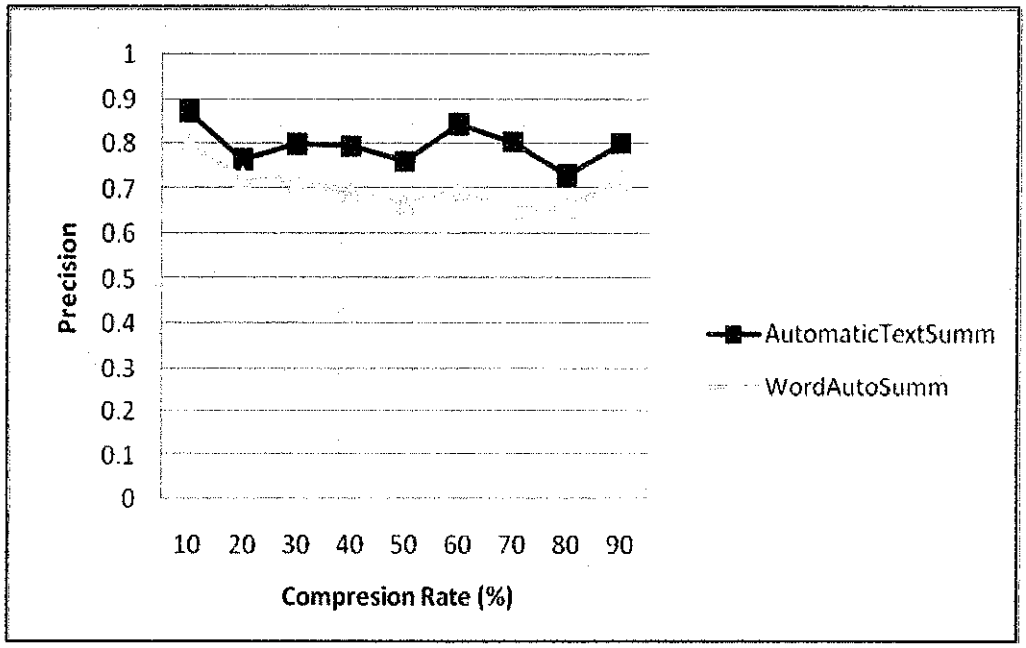


Figure 4.1: The average precision graph for Text Summarization System and Word Auto Summarizer using Article 6

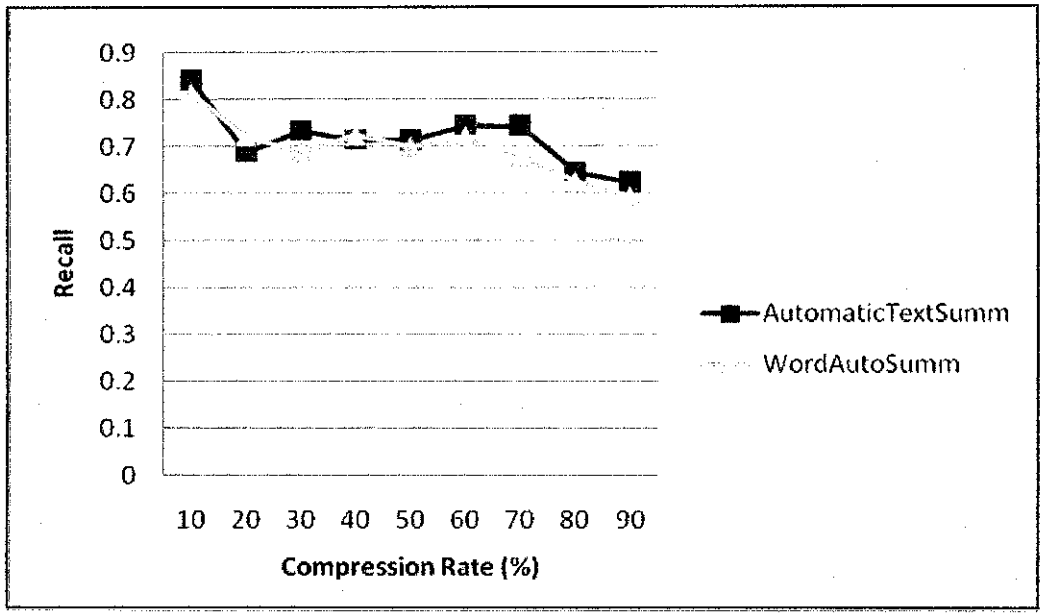


Figure 4.2: The average recall graph for Text Summarization System and Word Auto Summarizer using Article 6

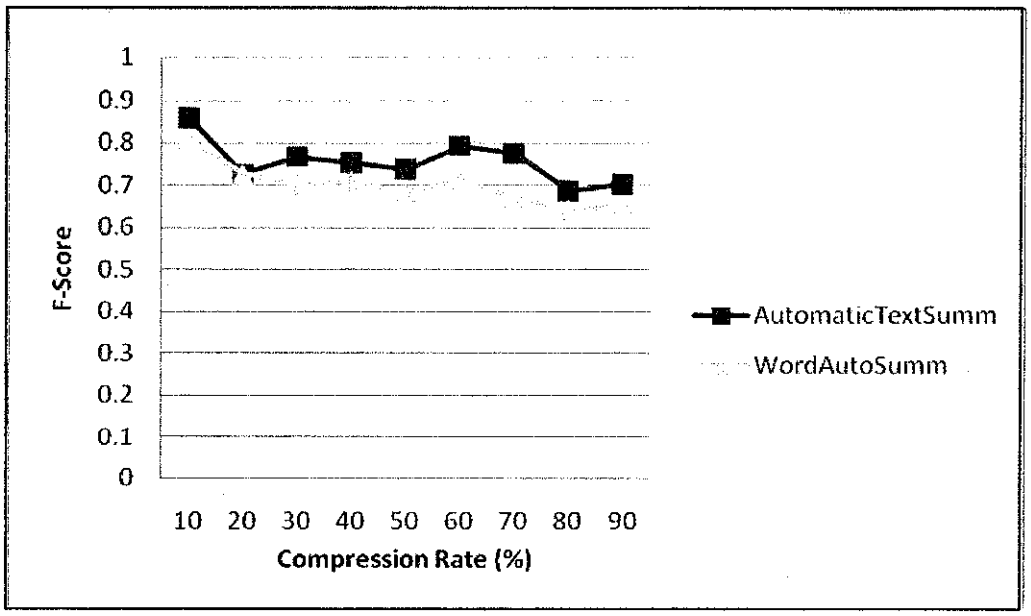


Figure 4.3: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 6

The average precision, recall, and F-Score shown in **Table 4.1**, indicates that Text Summarization System is better as compared to the Word Auto Summarizer at some compression rates. Both summaries were compared to the reference summary, which is generated by the expert (human made summary). However, the Text Summarization System, at some compression rates is left behind by the Word Auto Summarizer as indicated by the graphical representation in **Figures 4.1**, **4.2**, and **4.3**.

Evaluation of all auto-generated summaries by the system was based on the overlapping sentences when compared with established auto-generated summary system. The significant difference resulted by the evaluation was mostly because of the methods used by different system. Text Summarization System uses Bayesian theorem with knowledge base on the topic, whereas the Word Auto Summarizer uses the frequency-based theorem. When summaries generated by both system were compared to human made summary, the summary generated by the system has similar pattern as compared to Word Auto Summarizer's. AutomaticTextSumm has optimum greater value for precision, recall, and F-Score at compression rate 60% and 70% by 11.31% and 10.80% respectively. This is because the human made summary was also considering the knowledge base of the topic of which the article was being evaluated.

The human-generated summaries are most often generated to 2/3 of the original article's length (around 66.67% compression rate). At this compression rate level, AutomaticTextSumm leads the WordAutoSumm by nearly 11% similarity to the human-generated summary. AutoTextSumm has optimum value for precision at 87.33%, recall at 84.33% and F-Score at 85.80%. The current results could be considered satisfactorily.

4.3 Discussion

The project started developing the application by implementing the availability of the graphical user interface which is considered well developed. The development process was done with Java as the core language programming. The next step was the construction of the main function of the system, such as loading the input article, setting the compression rate and other operations. The most challenging part was the

implementation of the overall algorithm which was quite confusing due to unfriendliness of the developer in understanding the fundamental concept of Bayesian theorem.

The prototypes of the application were developed by following the priority of the application's functions. The first prototype was designed to have the basic user interface with the loading article function. The improvement of the prototype was then addressed the issue in processing textual data. As the prototype was improved, it is believed that all essential sub-functions for this Text Summarization System have been met. The development of the code design will continue in order to achieve code efficiency which determines the effectiveness of the application.

4.3.1 Results Evaluation

Some reasons behind the acquisition of the findings (results) exist. First, the combination of features in the system may not be applicable for other corpus. The previous researches used articles with average length much longer than 20 sentences. The system being developed by the project used most of the articles with average length of 15-20 sentences. Difference in articles' length affects the analysis due to the difference on the overall articles' structures. Therefore, the project considered the location or position factor.

The system uses the help provided by the list of stop words and keywords. The lists are used for the first round of weight assigning for each word in the article. The corpus was used in determining the weight for keyword. The system would process the corpus to find out how likely a keyword appears in the article. The corpus focuses specifically on oil and gas drilling topic, as mentioned from the very beginning. The limitation of the scope was aimed for the system to really focus on an in depth knowledge base.

Synonym sets was considered important since a single word could be represented by different words. For example, *offshore* could be replaced by *marine-based* or *sea-based.* These kinds of words should be taken in eh same way since they all represent a single same meaning.

The location or position of the sentence in the article was also taken into consideration in calculating the sentence weight. Sentences in the first two or three sentences of a paragraph are seen to be important. The methods was also used in treating the last first or two sentences of the article, because they most likely to bring conclusion of what the article is about.

Lastly, the system puts consideration on Bayesian theorem. The method processes the keywords with regards to the corpus of articles summaries being generated. The summary corpus was collected from the distribution of the articles to be summarized by experts; in this case, Petroleum Engineering and Linguistics lecturers.

As for finishing phase, the application implements a filtering and shortening algorithm. The filtering process was aimed to keep eh generated summary a cohesive summary. The sentence which begins with words like *it, they,* or *he* should be analyzed, so that the reader of the summary could understand to what a sentence refer to. Shortening process is done in a basic algorithm. When a sentence to be included in the summary is considered very long, the application would analyze the sentence and search for key points which connect two sentences into one. The key points mentioned above are like *e.g.,* and *where.*

It is concluded that the mixture of trying to identify important sentences for a summary from documents in a specific topic by using machine learning algorithm shows a similarity with the summaries generated by the expert (human-generated summaries). Therefore, the conclusion which have arisen from the results, suggest that this technique is suitable for a specific topic corpus and still have a lot of improvements to be made especially in terms of research.

Some constraints which inadvertently influenced the application's performance would have been identified if the project have had put more time and effort in enriching the corpus and deeper studying the algorithm.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Automatic text summarizer's demand is increasing in nowadays high-technology environment. The advanced technology has caused more inventions found and more information shared. Therefore, information overloading has to be faced by users who are more interested in shorter version of lengthy documents. There exist some available text summarizers in the market; *Microsoft Word Auto Summarizer, NetSumm, Pertinence* and *Extractor*. However, rooms for further improvement need to be addressed in order to produce better summaries, which are similar to the human-generated summaries. The evaluation on the summarizer's effectiveness is still a huge area of research.

A summarizer, normally, is an application that reads in a textual document or article, quantifies and classifies important words, removes the unnecessary contents, summarizes it using a certain technique within the chosen summary length. It also evaluates its effectiveness against some pre-defined criteria. Rapid change on technology could be valuable point to be used in betterment of text summarization effectiveness. Comparison between the system generated summary and the human made summary could be used as a technique in evaluating the effectiveness since the human made summary is assumed to be logic by having human brain algorithm in processing the summary.

The reason why the Text Summarization System produced evidently better summaries, i.e. nearer to the ideal standard of human-generated summary, than the Word Auto Summarizer, could be due to the topic specification. The developed system focuses on oil and gas drilling topic with the keywords and corpus as its knowledge base in predicting the likeliness of a sentence to be included in the summary. The summary

generated by the expert is also done by considering the main theme of the article and then applies the experts' knowledge in generating the summary. Considering more features in generating summaries; cue words, sentence position, keywords, and probability assumption, could further enhance the developed application.

5.2 Recommendation

Space for development and further improvement within the research project undeniably exist in order to boost and obtain the expected result, as opposed to the average results obtained from the project. The following states some recommendations for future enhancements.

The most important part of the Text Summarization System application system is to intelligently select the best sentences to be included in the generated summary. Therefore, by enhancing the algorithm in sentences selecting process, the system would be improved. For example, features such as title words, as introduced in a research by Kupiec, et al (1995).

Another way of improvements is to enhance the Natural Language Processing phase during the finishing phase (filtering and shortening summary sentences). This project uses a simple method of NLP in doing both processes. Filtering summary sentences is done by screening the sentences which have "I", "She", "He", "They", etc. Later, those words would be replaced by the reference sentences which appear earlier.

Too long sentences from the articles should be shortened in a better way. As for this project, the shortening phase is done by identifying summary sentences which are having length more than a particular number of words. The algorithm used in shortening the sentences is, so far, by identifying connecting words such as "while", "but", "however", etc. which appear in a single sentence. A semicolon (";") which separates a single sentence is also considered. The sentences are later shortened by truncating them according to the factors above.

The improvement on the corpus of the system would enhance and train the application in processing the probability of a sentence to be included in a summary. It would be a good idea to consider other machine learning techniques such as the decision tree algorithm and the Support Vector Machine (SVM). This is to determine whether other algorithms might be suitable for the features chosen and the corpus used for the evaluation.

REFERENCES

[1] E.Qwiener, J.O. Pederson, and A.S.Weigned, "A neural network approach to topic spotting", In Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrievel (SDAIR'95), 1995

[2] Edmunson H.P., "New Methods in Automatic Extraction", Journal of the ACM 16(2) pp 264 – 285, 1969

[3] Endres-Niggemeyer, B., "Human-Style Www Summarization", Department of Information and Communication, Fachhochschule Hannover University for Applied Sciences, Hanover Germany, 2000

[4] Ganapathiraju M.K., "Relevance of Cluster Size in MMR-based Summarizer: A Report", Sel-paced lab in Information Retrieval, 2002

[5] Hassel, M and H. Dalianis, "Generation of Reference Summaries", In Proceedings of the 2nd Language and technology Conference on Human Language Technology as a Challenge for Computer Science and Linguistics, 2005

[6] Hassel, Martin. "Evaluation of automatic text summarization - a practical implementation", Licentiate thesis, Stockholm, NADA-KTH, 2004

[7] Jing, H. and K.R. McKeown, "The Decomposition of Human-Written Summary Sentences", Department of Computer Science Columbia University, New York, 1999

[8] Joachims, T., "Text Categorization with SupportVector Machins: Learning with Many Relevant Features", In European Conference on Machine Learning (ECML), 1998

[9] Luhn H.P., "The Automatic Creation of Literature Abstract", IBM Journal of Research and Development pp 159 – 165, 1959

[10] Mani. I., and M.T. Maybury (eds.), "Advances in automatic text summarization", pp. 111-121. Cambridge, Massachusetts: MIT Press, 1999

[11] Mani, I., "Automatic Summarization". 2001: John Benjamin's Publishing Company

[12] Saggio, Horacio., "Automatic Text Summarization: Past, Present and Future", Department of Computer Science University of Sheffield, England, United Kingdom, 2004

[13] Salton G., "Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer", Adison Wesley Publishing Company, 1989

[14] Teufel S and Moens M., "Sentence extraction as a classification task", In ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization, Madrid, Spain, 1997

[15] Tsuruoka, Y., Kawaguchi-shi, Tsujii, J., "Journal of Biomedical Informatics archive", Vol.37(6), pp. 461-470, 2004

[16] Victoria, M., "Statistical Approaches to Automatic Text Summarization", Buletin of the American Society for Information Science and Technology, Vol3(4), April/May 2004

[17] Y.Yang, C.G.Chute, "An example-based mapping method for text categorization and retrievel", ACM Transaction on Information Systems (TOIS), 12(3):252-277, 1994

APPENDICES

APPENDIX A

SYSTEM'S GRAPHICAL USER INTERFACE

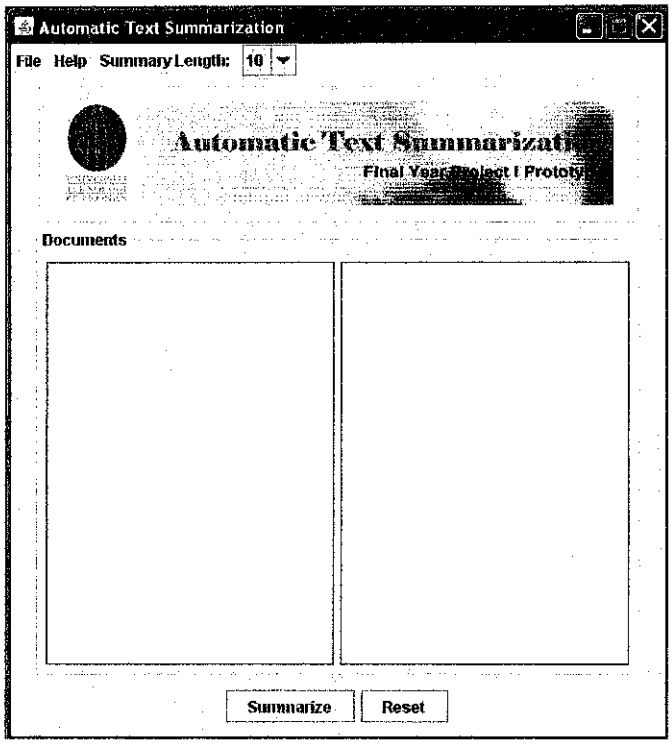


Figure 3.9: System's Main GUI

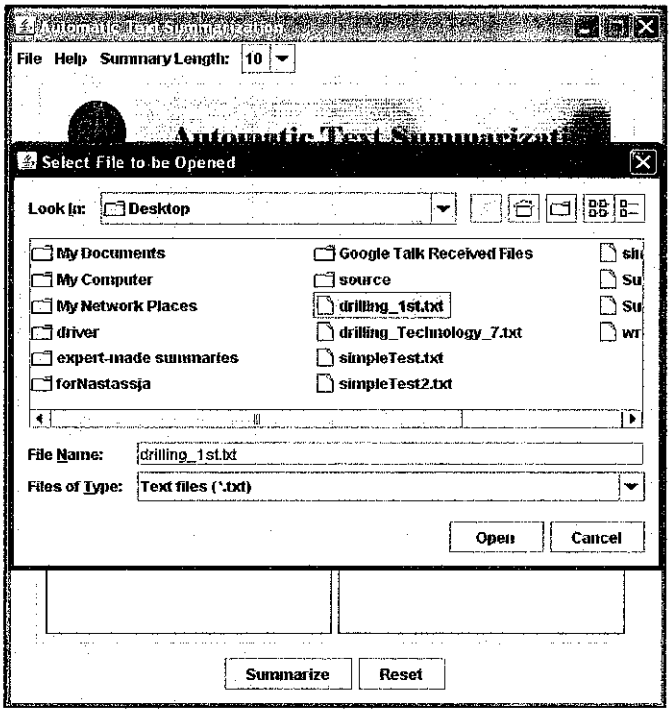


Figure 3.10: Open Document Function

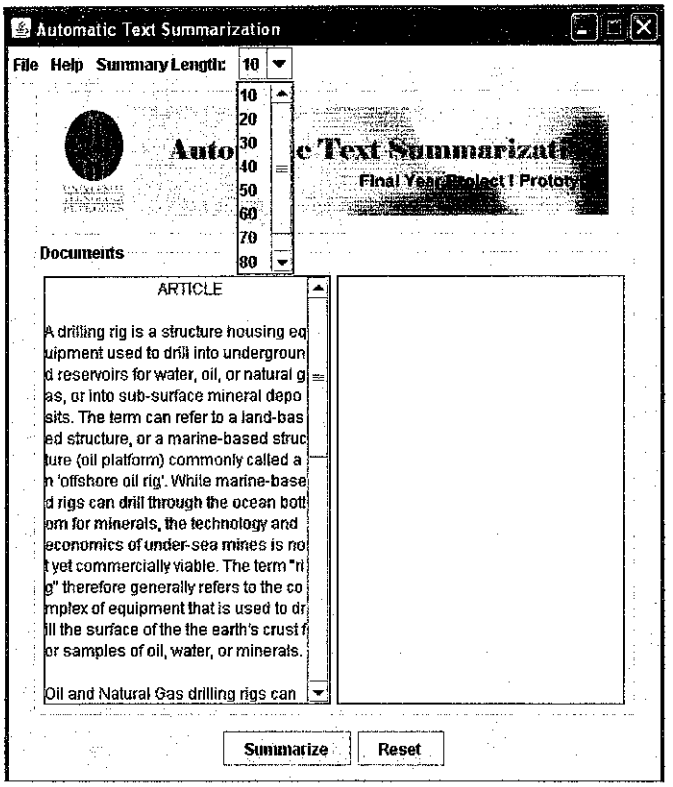


Figure 3.11: Setting the Summary Length Function

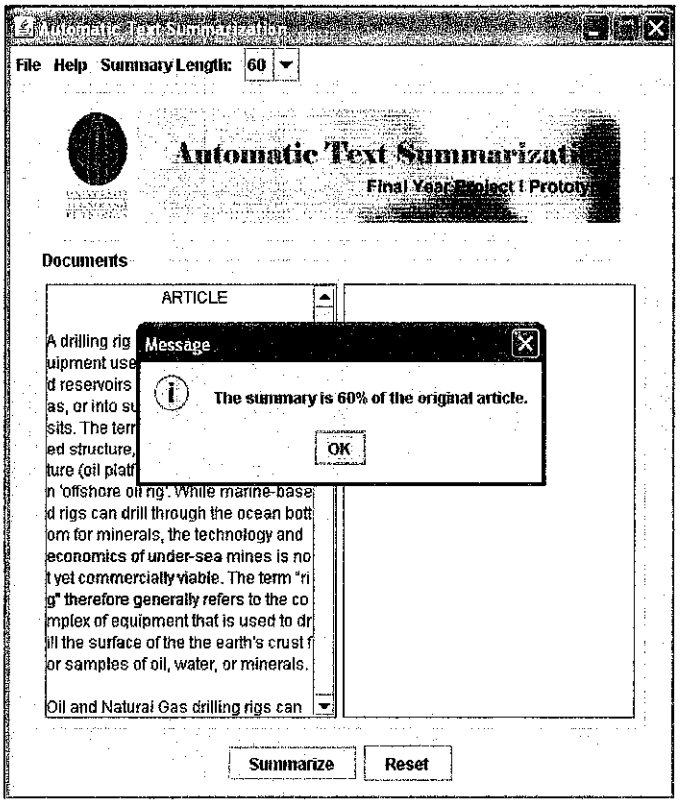


Figure 3.12: Confirmation of Summary Length

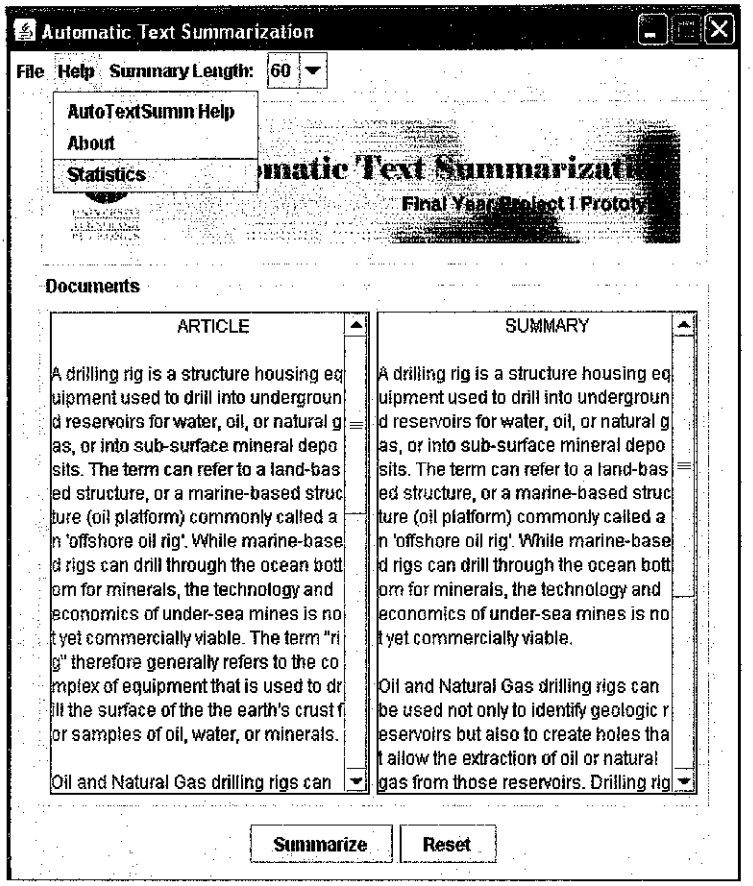


Figure 3.13: Help-Statistics Function

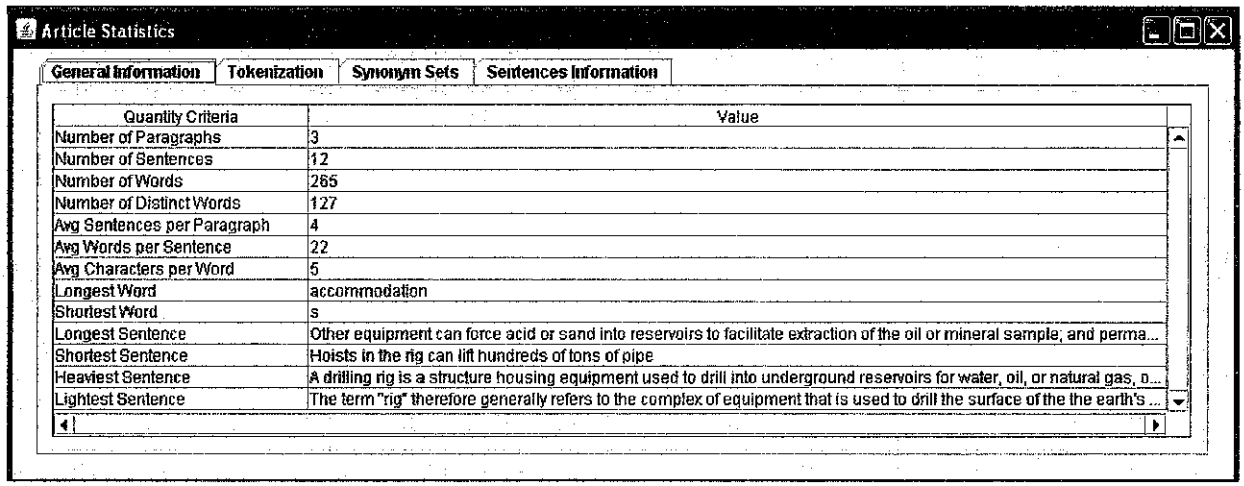
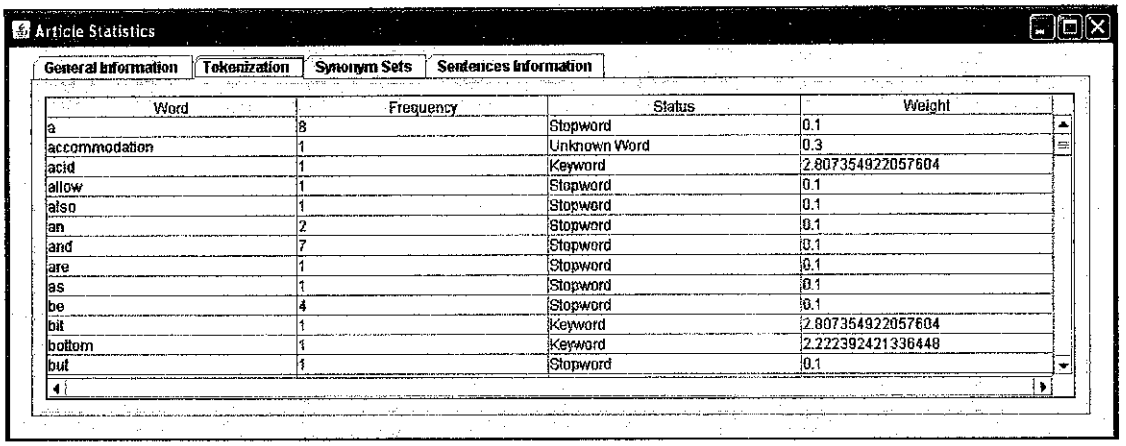
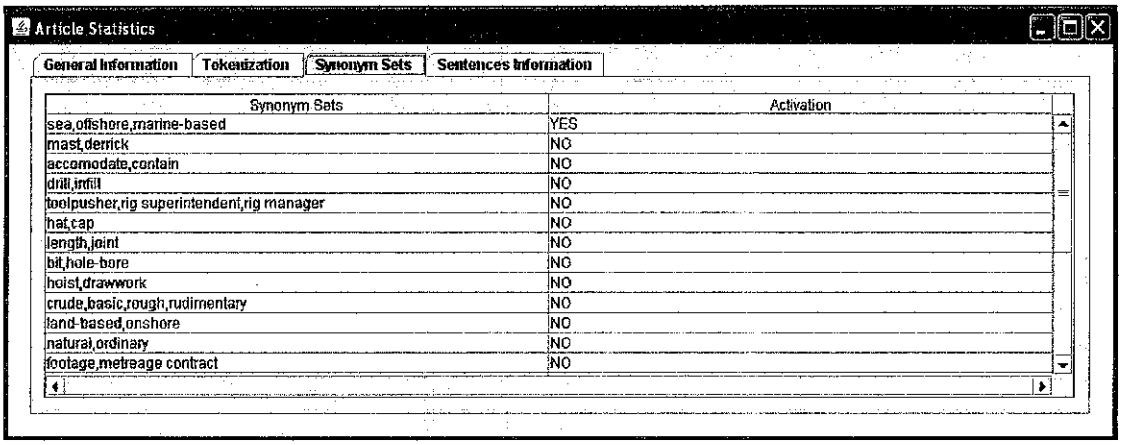


Figure 3.14: Statistics General Information



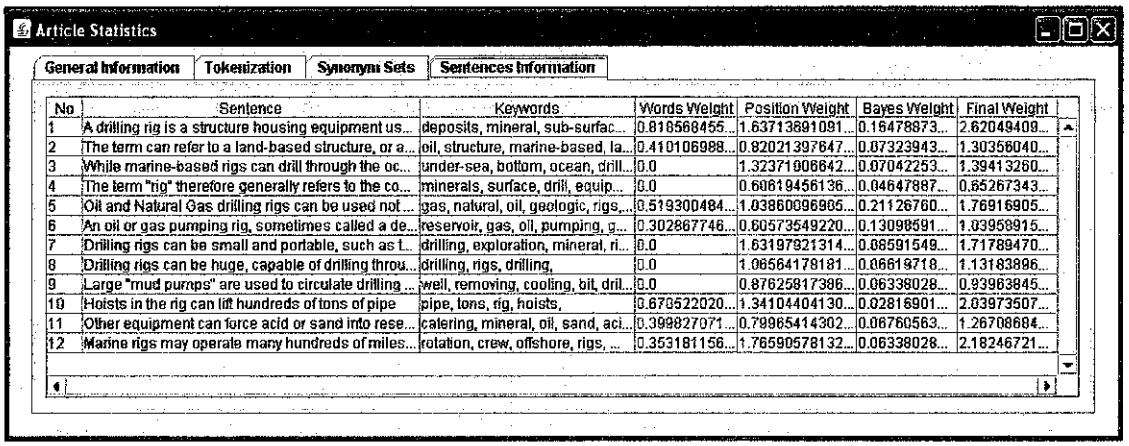
Word	Frequency	Status	Weight
a	8	Stopword	0.1
accommodation	1	Unknown Word	0.3
acid	1	Keyword	2.807354922057604
allow	1	Stopword	0.1
also	1	Stopword	0.1
an	2	Stopword	0.1
and	7	Stopword	0.1
are	1	Stopword	0.1
as	1	Stopword	0.1
be	4	Stopword	0.1
bit	1	Keyword	2.807354922057604
bottom	1	Keyword	2.222392421336448
but	1	Stopword	0.1

Figure 3.15: Statistics Tokenization



Synonym Sets	Activation
sea,offshore,marine-based	YES
mast,derrick	NO
accomodate,contain	NO
drill,infill	NO
toolpusher,rig superintendent,rig manager	NO
hat,cap	NO
length,joint	NO
bit,hole-bore	NO
hoist,drawwork	NO
crude,basic,rough,rudimentary	NO
land-based,onshore	NO
natural,ordinary	NO
footage,metreage contract	NO

Figure 3.16: Statistics Synonym Sets



No	Sentence	Keywords	Words Weight	Position Weight	Bayes Weight	Final Weight
1	A drilling rig is a structure housing equipment us...	deposits, mineral, sub-surfac...	0.818568455...	1.63713691091...	0.16478873...	2.62049409...
2	The term can refer to a land-based structure, or a...	oil, structure, marine-based, la...	0.410106988...	0.82021397647...	0.07323943...	1.30356040...
3	While marine-based rigs can drill through the oc...	under-sea, bottom, ocean, drill...	0.0	1.32371906642...	0.07042253...	1.39413260...
4	The term "rig" therefore generally refers to the co...	minerals, surface, drill, equip...	0.0	0.60619456136...	0.04647887...	0.65267343...
5	Oil and Natural Gas drilling rigs can be used not ...	gas, natural, oil, geologic, rigs,...	0.519300484...	1.03860096965...	0.21126760...	1.76916905...
6	An oil or gas pumping rig, sometimes called a de...	reservoir, gas, oil, pumping, g...	0.302867746...	0.60573549220...	0.13098591...	1.03958915...
7	Drilling rigs can be small and portable, such as t...	drilling, exploration, mineral, ri...	0.0	1.63197921314...	0.08591549...	1.71789470...
8	Drilling rigs can be huge, capable of drilling throu...	drilling, rigs, drilling,	0.0	1.06564178181...	0.06619718...	1.13183896...
9	Large "mud pumps" are used to circulate drilling ...	well, removing, cooling, bit, dril...	0.0	0.87625817386...	0.06338028...	0.93963845...
10	Hoists in the rig can lift hundreds of tons of pipe	pipe, tons, rig, hoists,	0.670522020...	1.34104404130...	0.02816901...	2.03973507...
11	Other equipment can force acid or sand into rese...	catering, mineral, oil, sand, aci...	0.399827071...	0.79965414302...	0.06760563...	1.26708684...
12	Marine rigs may operate many hundreds of miles...	rotation, crew, offshore, rigs, ...	0.353181156...	1.76590578132...	0.06338028...	2.18246721...

Figure 3.17: Statistics Sentence Information

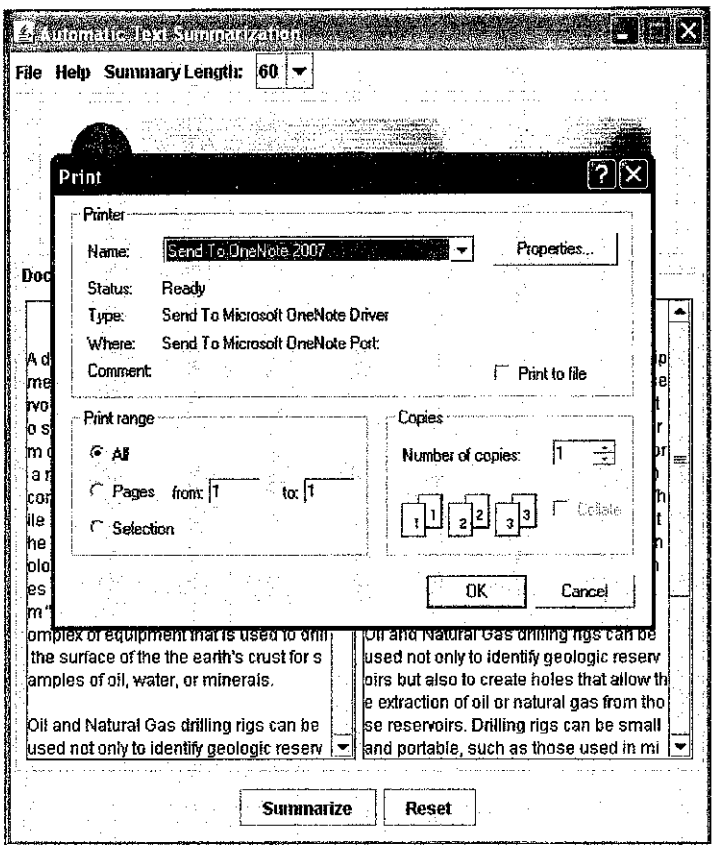
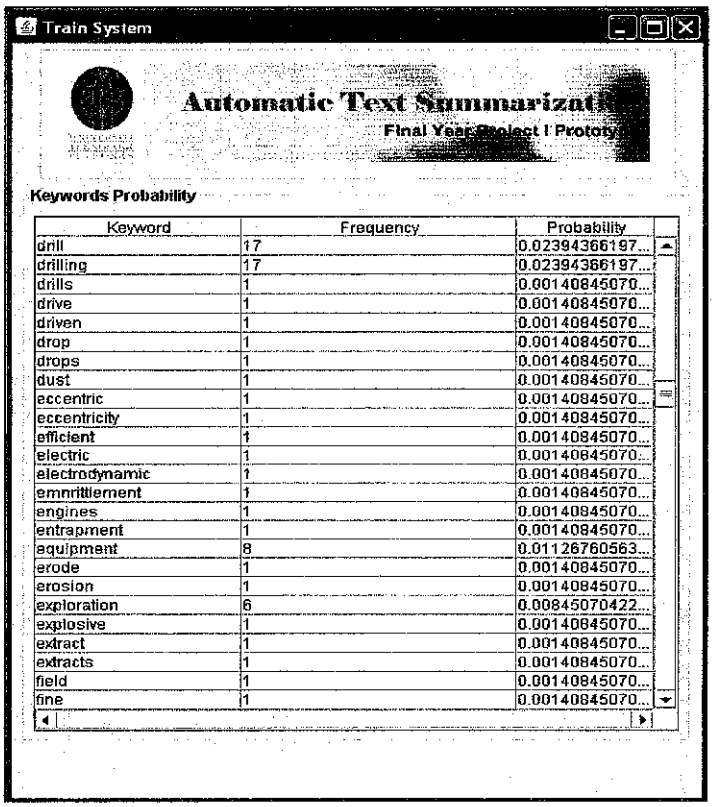


Figure 3.18: Print Summary Function



Keyword	Frequency	Probability
drill	17	0.02394366197...
drilling	17	0.02394366197...
drills	1	0.00140845070...
drive	1	0.00140845070...
driven	1	0.00140845070...
drop	1	0.00140845070...
drops	1	0.00140845070...
dust	1	0.00140845070...
eccentric	1	0.00140845070...
eccentricity	1	0.00140845070...
efficient	1	0.00140845070...
electric	1	0.00140845070...
electrodynamic	1	0.00140845070...
emnrittlement	1	0.00140845070...
engines	1	0.00140845070...
entrapment	1	0.00140845070...
equipment	8	0.01126760563...
erode	1	0.00140845070...
erosion	1	0.00140845070...
exploration	6	0.00845070422...
explosive	1	0.00140845070...
extract	1	0.00140845070...
extracts	1	0.00140845070...
field	1	0.00140845070...
fine	1	0.00140845070...

Figure 3.19: Keywords' Probability Table

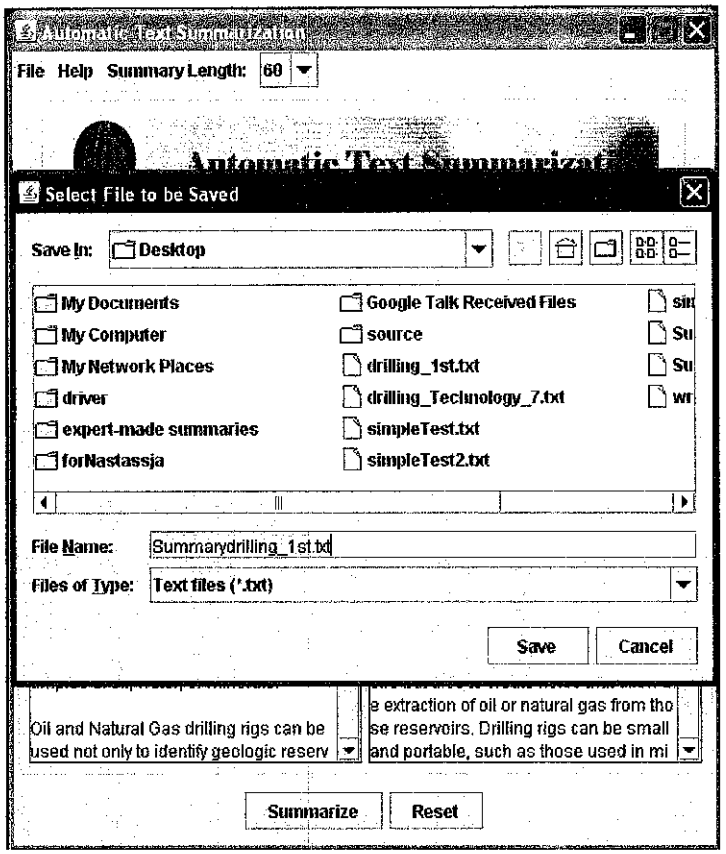


Figure 3.20: Save Summary Function

APPENDIX B

SYSTEM'S HELP FILE

Introduction

Text Summarization System is a tool to summarize the articles focusing on oil & gas drilling topic. The system implements the Bayesian network as the knowledge base in generating the output summary. Text Summarization System with Bayesian Network on Oil & Gas Drilling Topic is the final year project done by Iwan Kurniawan, Information & Communication Technology student of Universiti Teknologi PETRONAS.

Under the supervision of Ms. Vivian Yong Suet Peng and Ms. Amy Foong, the system tries to implement Bayesian network in generating the summary. The tool used to develop the bayesian network is called Weka, a published tool for data mining processing. Thanks to Ian Wite, the developer of Weka.

How it Works

The Text Summarization System consists of several steps in order to come up with the summary. The input article (which is in .txt file), would be analyzed by first taking all the words exist in the article. The system will identify the words which are categorized under stop word list. Stop words are the words commonly appear in any article which do not give meaningful information or do not carry the main topic of the article.

Stemming algorithm is applied in order to get the root form of words. This is done to avoid the repetition in analyzing or assigning weight of particular words. Example, word 'going', 'go', and 'goes' are basically having single meaning; 'go'.

Next step is the keywords analysis. The system should be able to detect words fall under keyword in the database. This will be a comparison process with the knowledge base developed before. The weight of each keyword could be different, depending on the training sets of the knowledge base. Other words which are not under stopwords or keywords would be treated as unknown words, given the weight in the middle range between the stop words and the keywords.

Sentences would be assigned weight of importance level. The sentence weight calculation would consider each word's weight. The higher the weight of a sentence, the more important it would be seen by the system. The summary is generated based on the weight score of the sentence and also the preferred summary length chosen by the users. The system should be able to display summary containing most important sentences but displaying them based on the sentences order of the article.

How to use

1. Open the desired text article which is going to be summarized.
   File > Open Document

2. Set the Summary length to be displayed.
   (10%, 20%, 30, 40%, 50%, 60%, 70%, 80% or 90%)

3. Click the 'Summarize'.

4. The users may save the summary output.
   File > Save Summary as...

5. The users may print the input article and summary output.
   File > Print...

6. The users may view the statistics of the input article.
   Help > Statistics

   The statistics function is to display the necessary information of the input article statistics. The statistics includes the following:

Number of Paragraphs	Longest Word
Number of Sentences	Shortest Word
Number of Words	Longest Sentence
Number of Distinct Words	Shortest Sentence
Avg sentences/Paragraph	Heaviest Sentence
Avg Words/Sentence	Lightest Sentence
Avg characters/Word	

7. The users may find the information about the system.
   Help > About

8. The users may open the help document on how to use the system.
   Help > AutoTextSumm Help

9. Quits the application.
   File > Quit

APPENDIX C

EXPERIMENTAL RESULT OF AUTO-GENERATED SUMMARIES (AutomaticTextSumm & WordAutoSumm) USING ARTICLE 01

Evaluation on: Article 01						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.177916	0.216421	0.195289	0.726253	0.420711	0.532785
20	0.311708	0.452873	0.369259	0.726232	0.420703	0.532773
30	0.55586	1.251544	0.769815	0.726252	0.42071	0.532784
40	0.469625	0.885457	0.613738	1.028791	0.507096	0.679341
50	0.394575	0.651731	0.491551	1.0654	0.515832	0.695113
60	0.203516	0.255518	0.226572	1.212887	0.548102	0.755014
70	0.332637	0.498435	0.398998	2.208209	0.6883	1.049477
80	0.164321	0.196632	0.17903	0.980329	0.495033	0.657866
90	0.164321	0.196632	0.17903	0.980391	0.495049	0.657894

Table 4.2: The precision, recall and F-score for AutomaticTextSumm and WordAutoSumm using article 01

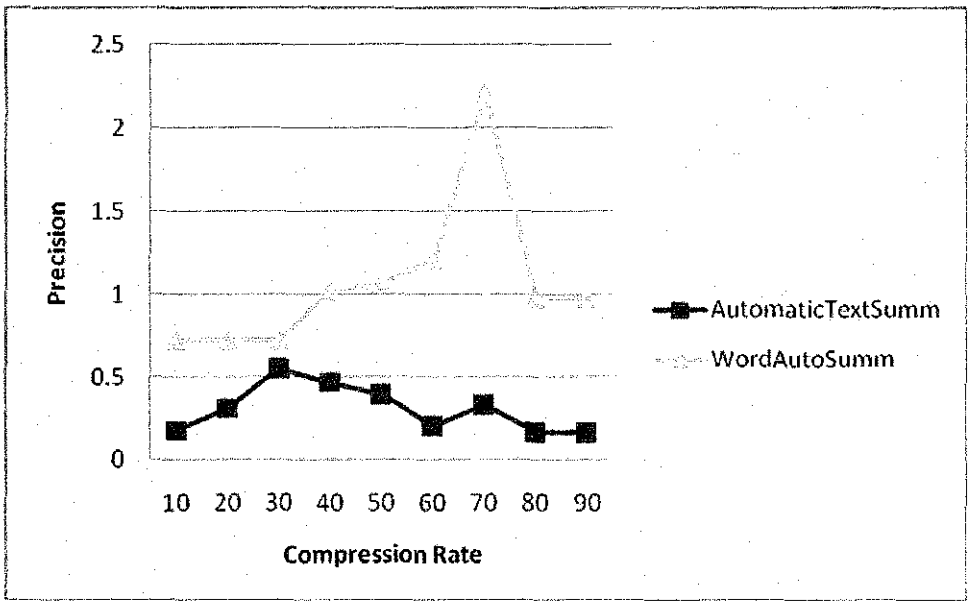


Figure 4.4: The average precision graph for Text Summarization System and Word Auto Summarizer using Article 01

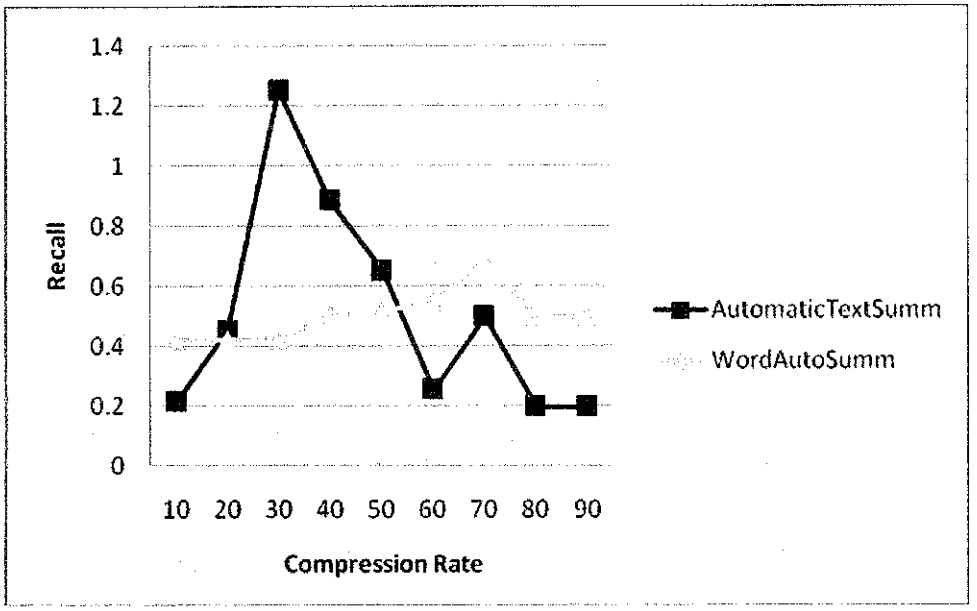


Figure 4.5: The average recall graph for Text Summarization System and Word Auto Summarizer using Article 01

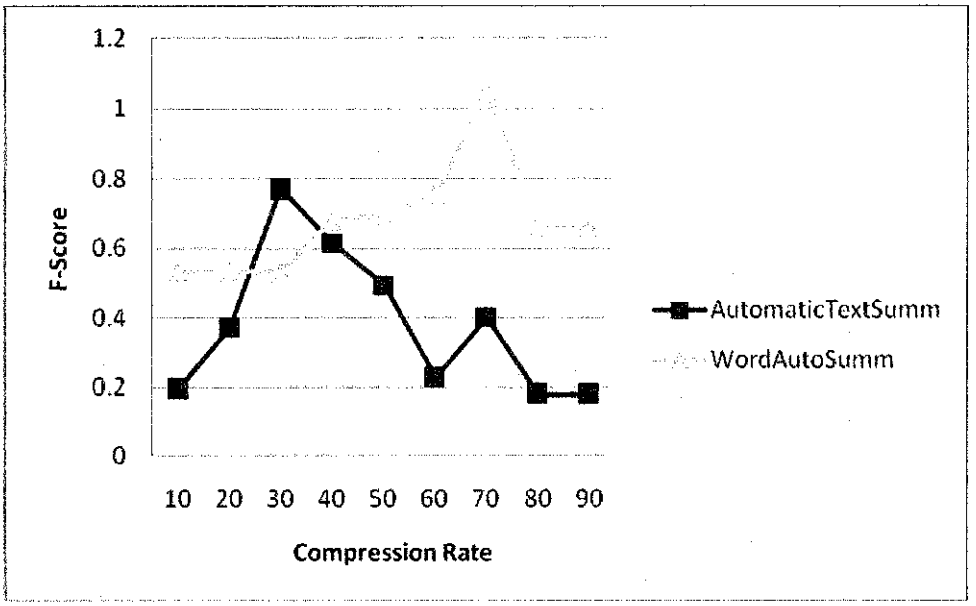


Figure 4.6: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 01

APPENDIX D

EXPERIMENTAL RESULT OF AUTO-GENERATED SUMMARIES (AutomaticTextSumm & WordAutoSumm) USING ARTICLE 02

Evaluation on: Article 02						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.499454	0.99782	0.665697	0.367938	0.582122	0.450887
20	0.36792	0.582078	0.45086	0.625698	0.38488	0.476596
30	0.865452	6.43231	1.525634	0.214912	0.176895	0.194059
40	0.950699	19.28341	1.81206	0.515084	1.062212	0.693755
50	0.890459	8.128964	1.605093	0.231519	0.301268	0.261828
60	0.890459	8.128964	1.605093	0.231519	0.301268	0.261828
70	0.701329	2.34817	1.080073	1.800554	0.642928	0.947522
80	0.230411	0.299395	0.260412	1.800554	0.642928	0.947522
90	0.230411	0.299395	0.260412	1.172414	0.539683	0.73913

Table 4.3: The precision, recall and F-score for AutomaticTextSumm and WordAutoSumm using article 02

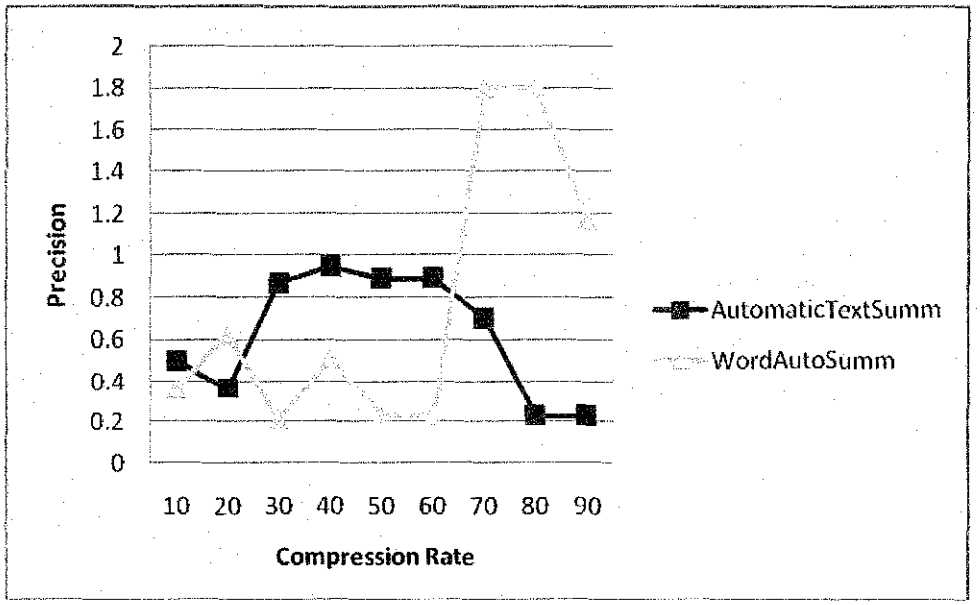


Figure 4.7: The average Precision graph for Text Summarization System and Word Auto Summarizer using Article 02

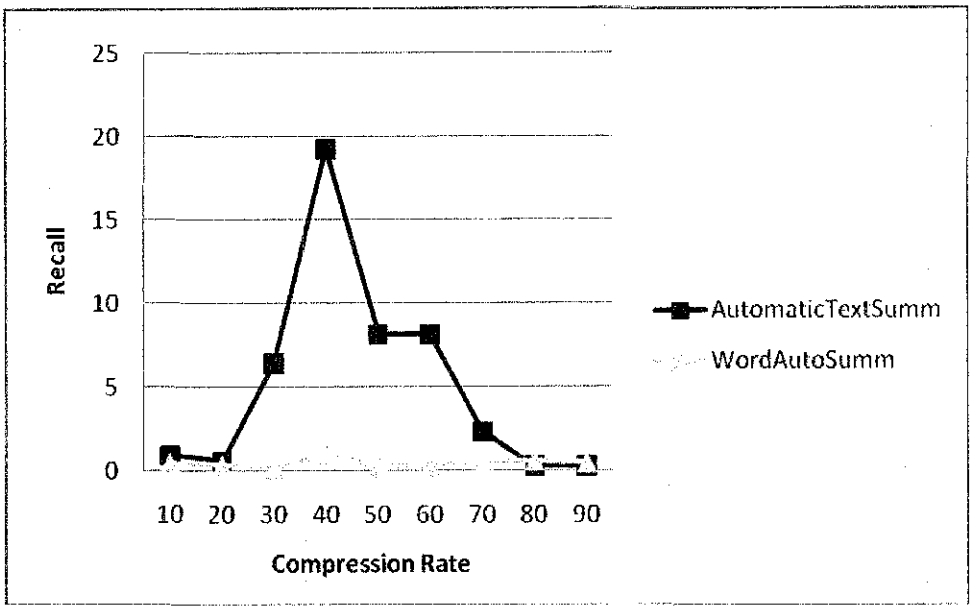


Figure 4.8: The average Recall graph for Text Summarization System and Word Auto Summarizer using Article 02

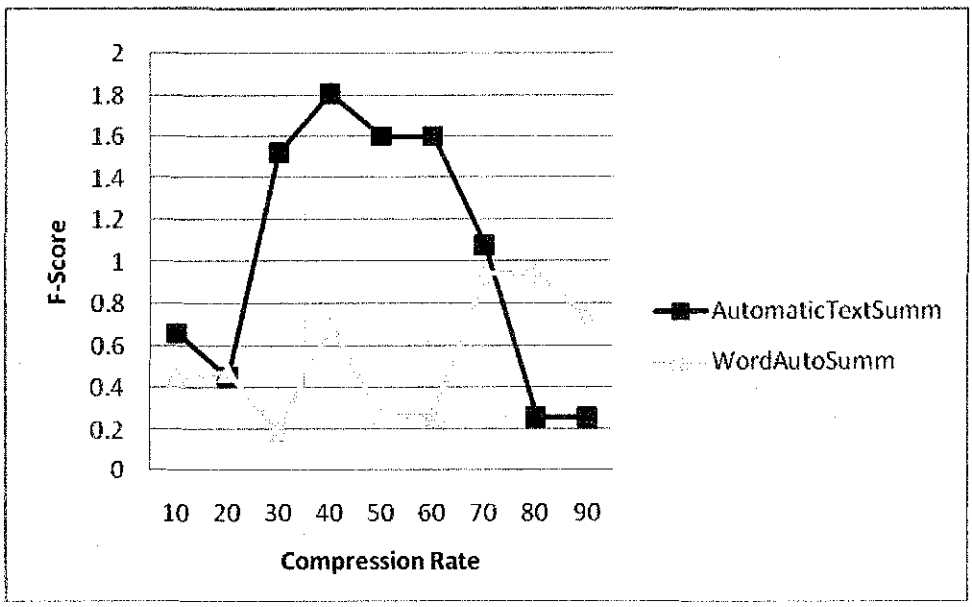


Figure 4.9: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 02

APPENDIX E

EXPERIMENTAL RESULT OF AUTO-GENERATED SUMMARIES (AutomaticTextSumm & WordAutoSumm) USING ARTICLE 14

Evaluation on: Article 14						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.649020721	1.849171	0.960815	0.498788	0.332794	0.399224
20	0.75620323	3.101777	1.215959	0.213302	0.271136	0.238767
30	0.756213732	3.101954	1.215987	0.21342	0.271326	0.238914
40	0.956653789	22.07007	1.833818	0.91467	10.71922	1.685515
50	0.935196911	14.43136	1.756563	0.57401	1.347474	0.805069
60	0.956144358	0.48879	0.646886	0.213891	0.176203	0.193226
70	0.895212796	8.54315	1.620607	0.636506	1.751079	0.93364
80	0.965148153	27.69288	1.865288	0.635992	1.747191	0.932534
90	0.900513322	9.051597	1.638061	0.815503	4.420147	1.376961

Table 4.4: The precision, recall and F-score for AutomaticTextSumm and WordAutoSumm using article 13

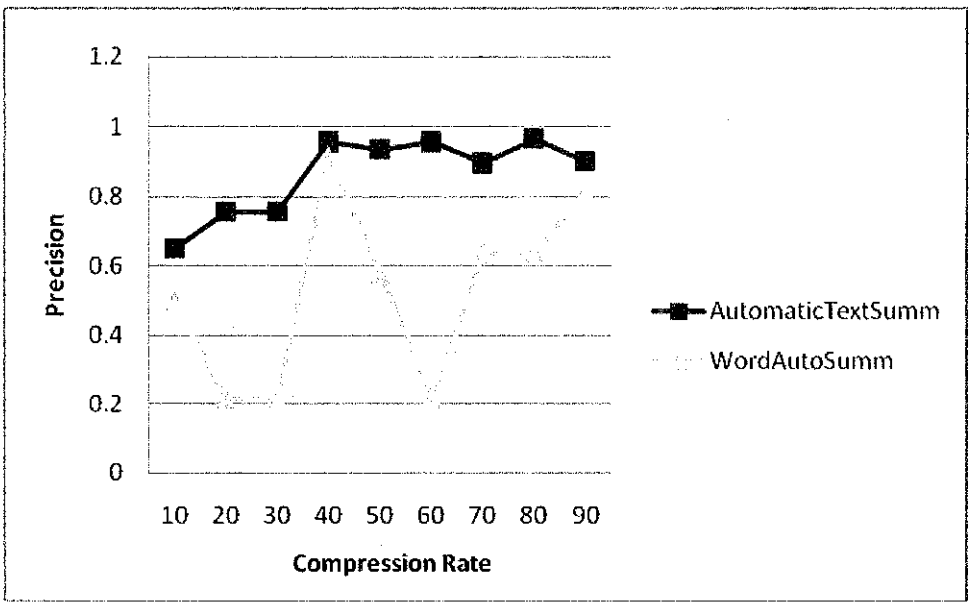


Figure 4.10: The average Precision graph for Text Summarization System and Word Auto Summarizer using Article 13

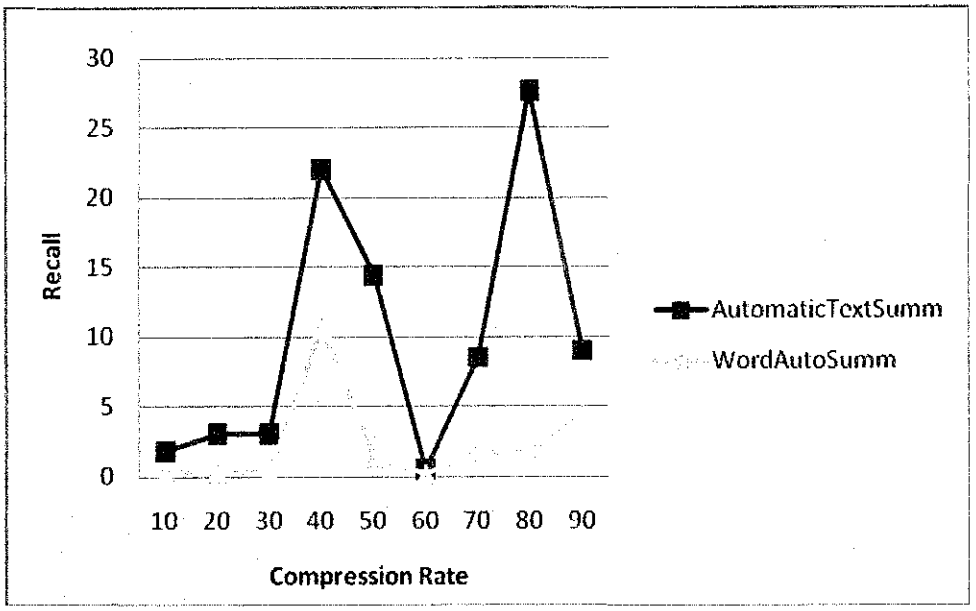


Figure 4.11: The average Recall graph for Text Summarization System and Word Auto Summarizer using Article 13

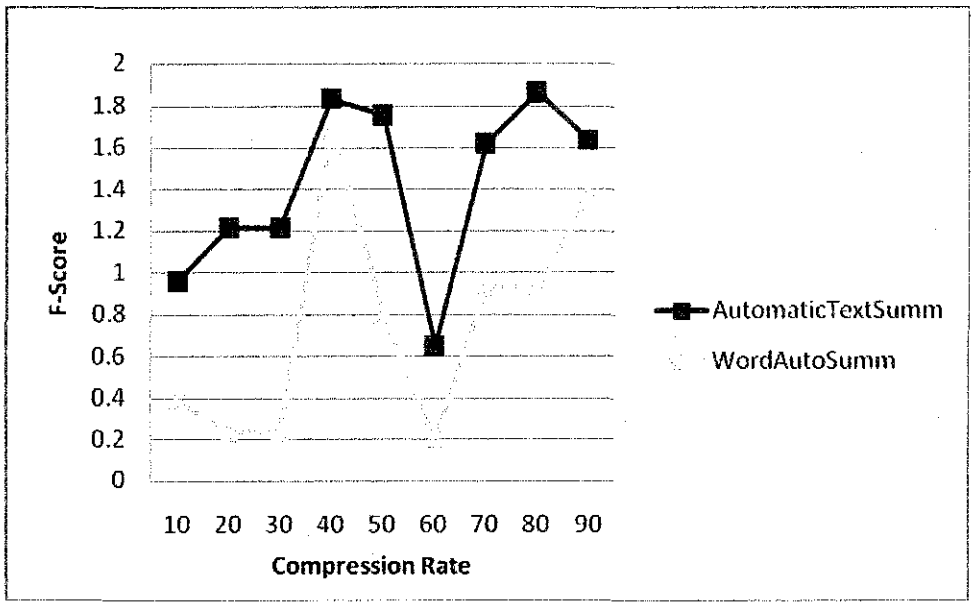


Figure 4.12: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 13

APPENDIX F

EXPERIMENTAL RESULT OF AUTO-GENERATED SUMMARIES (AutomaticTextSumm & WordAutoSumm) USING ARTICLE 7

Evaluation on: Article 7						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.8733	0.7433	0.803073	0.7992	0.7134	0.753867
20	0.5	0.6907	0.580079	0.5	0.6234	0.554923
30	0.80003	0.7357	0.766518	0.7154	0.6879	0.701381
40	0.8	0.7143	0.754725	0.8	0.7247	0.760491
50	0.7597	0.7341	0.746681	0.6667	0.7008	0.683325
60	0.75	0.7462	0.748095	0.75	0.72307	0.736289
70	0.8041	0.7448	0.773315	0.6566	0.6799	0.668047
80	0.9	0.8458	0.872059	0.9	0.7626	0.825623
90	1	0.9239	0.960445	0.9091	0.7944	0.847889

Table 4.5: The precision, recall and F-score for AutomaticTextSumm and WordAutoSumm using article 7

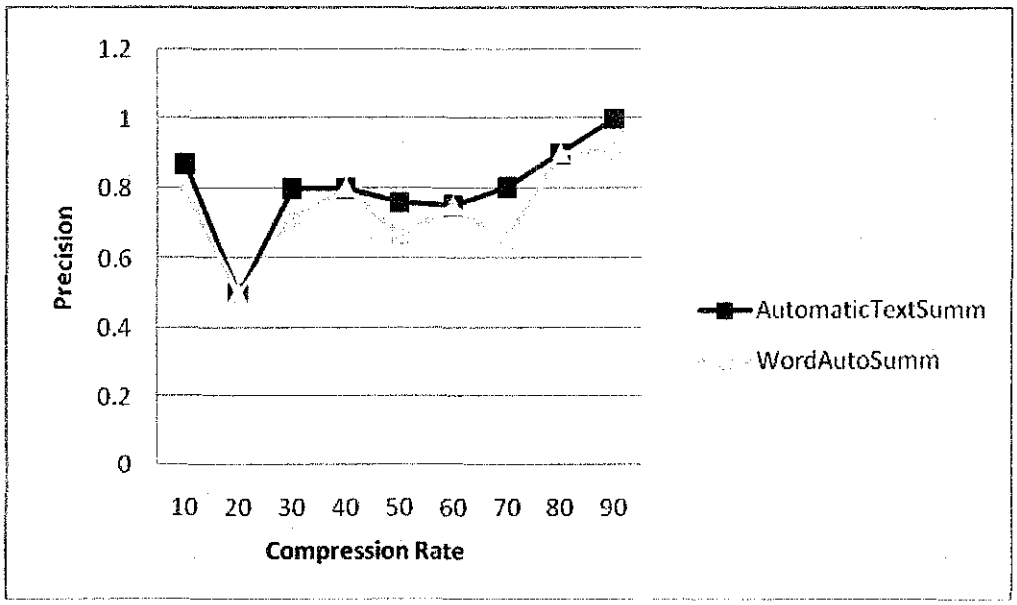


Figure 4.13: The average Precision graph for Text Summarization System and Word Auto Summarizer using Article 7

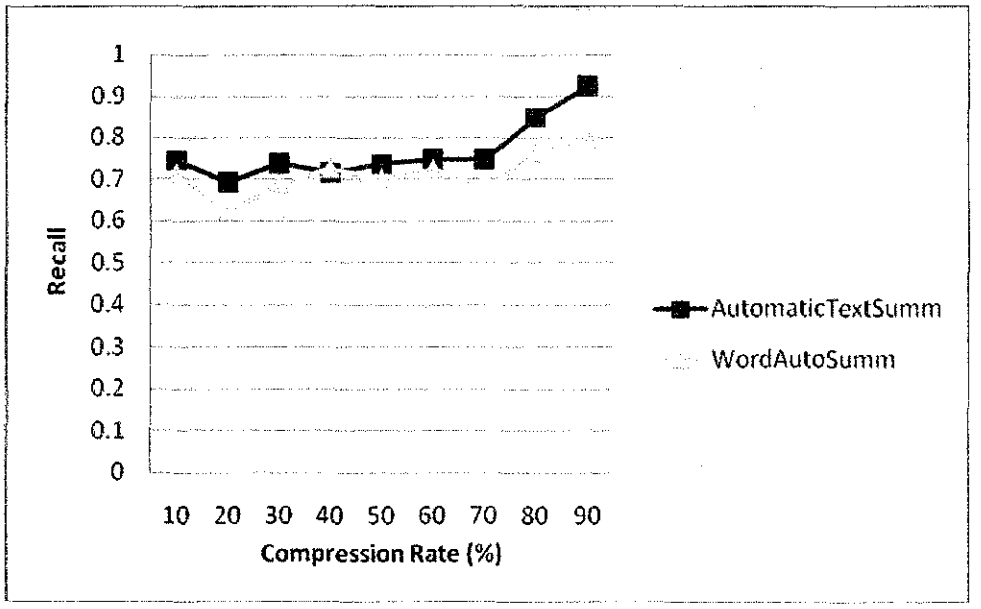


Figure 4.14: The average Recall graph for Text Summarization System and Word Auto Summarizer using Article 7

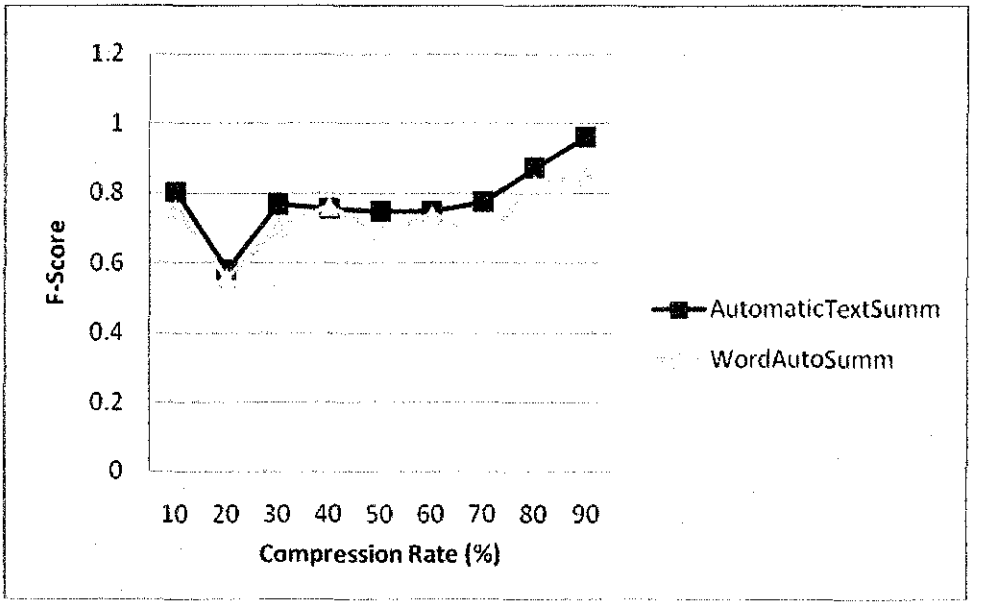


Figure 4.15: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 7

APPENDIX G

EXPERIMENTAL RESULT OF AUTO-GENERATED SUMMARIES (AutomaticTextSumm & WordAutoSumm) USING ARTICLE 11

Evaluation on: Article 11						
Compression Rate (%)	AutomaticTextSumm			WordAutoSumm		
	Precision	Recall	F-Score	Precision	Recall	F-Score
10	0.942	0.9256	0.933728	0.8412	0.8646	0.85274
20	0.8334	0.773	0.802064	0.7692	0.7746	0.771891
30	0.8734	0.818	0.844793	0.7574	0.7391	0.748138
40	0.8642	0.7966	0.829024	0.7362	0.7759	0.755529
50	0.8284	0.7964	0.812085	0.7087	0.752	0.729708
60	0.913	0.8285	0.8687	0.6792	0.7223	0.700087
70	0.8728	0.8271	0.849336	0.6427	0.7311	0.684056
80	0.798	0.7281	0.761449	0.6983	0.6772	0.687588
90	0.8696	0.7062	0.779428	0.7636	0.6456	0.69966

Table 4.6: The precision, recall and F-score for AutomaticTextSumm and WordAutoSumm using article 11

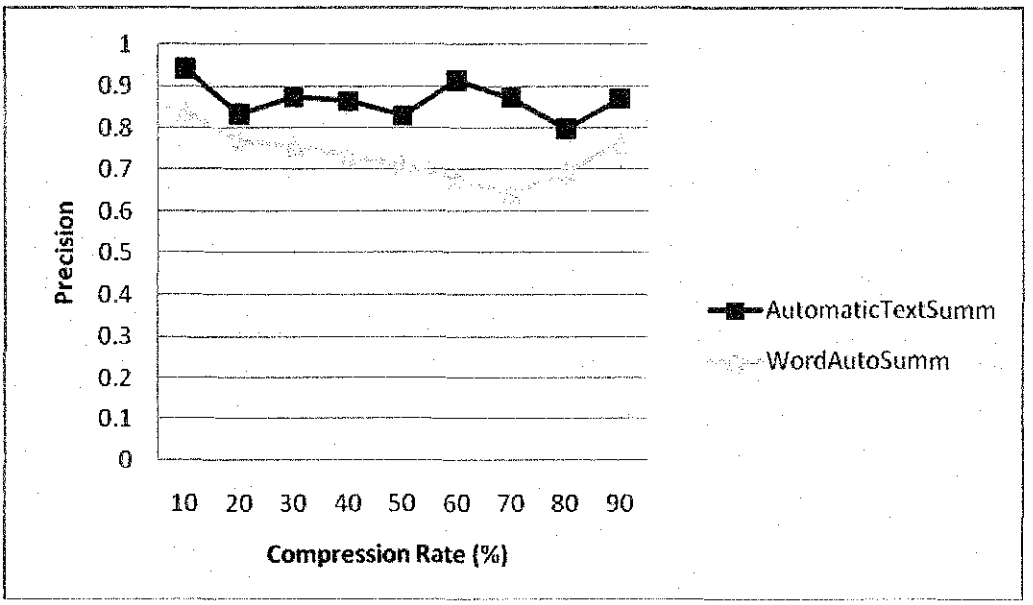


Figure 4.16: The average Precision graph for Text Summarization System and Word Auto Summarizer using Article 11

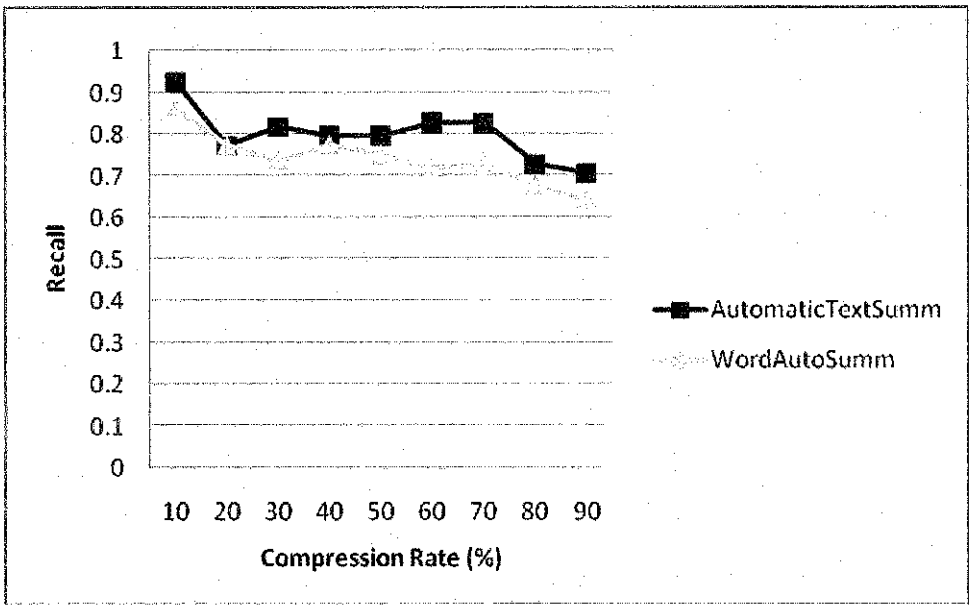


Figure 4.17: The average Recall graph for Text Summarization System and Word Auto Summarizer using Article 11

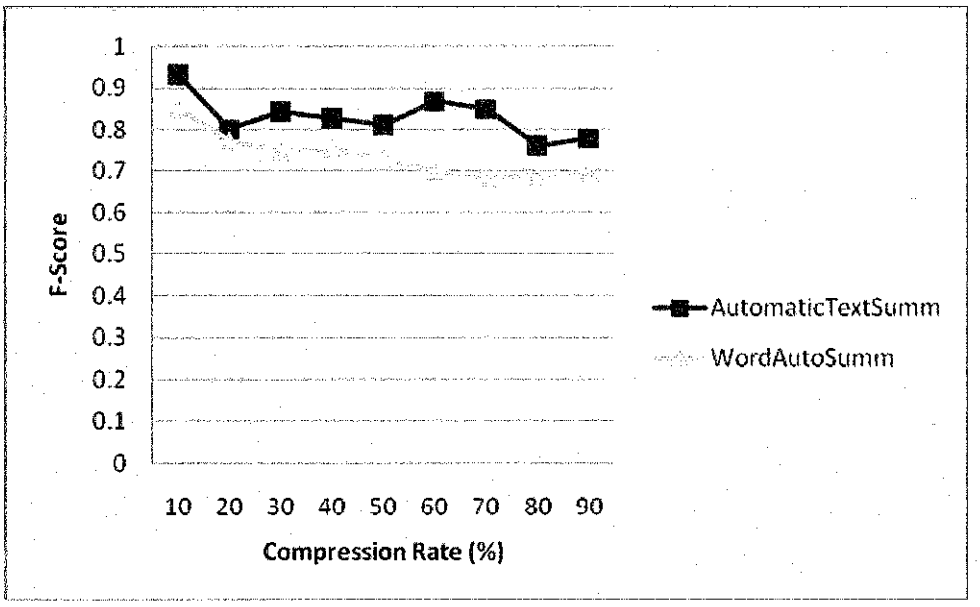
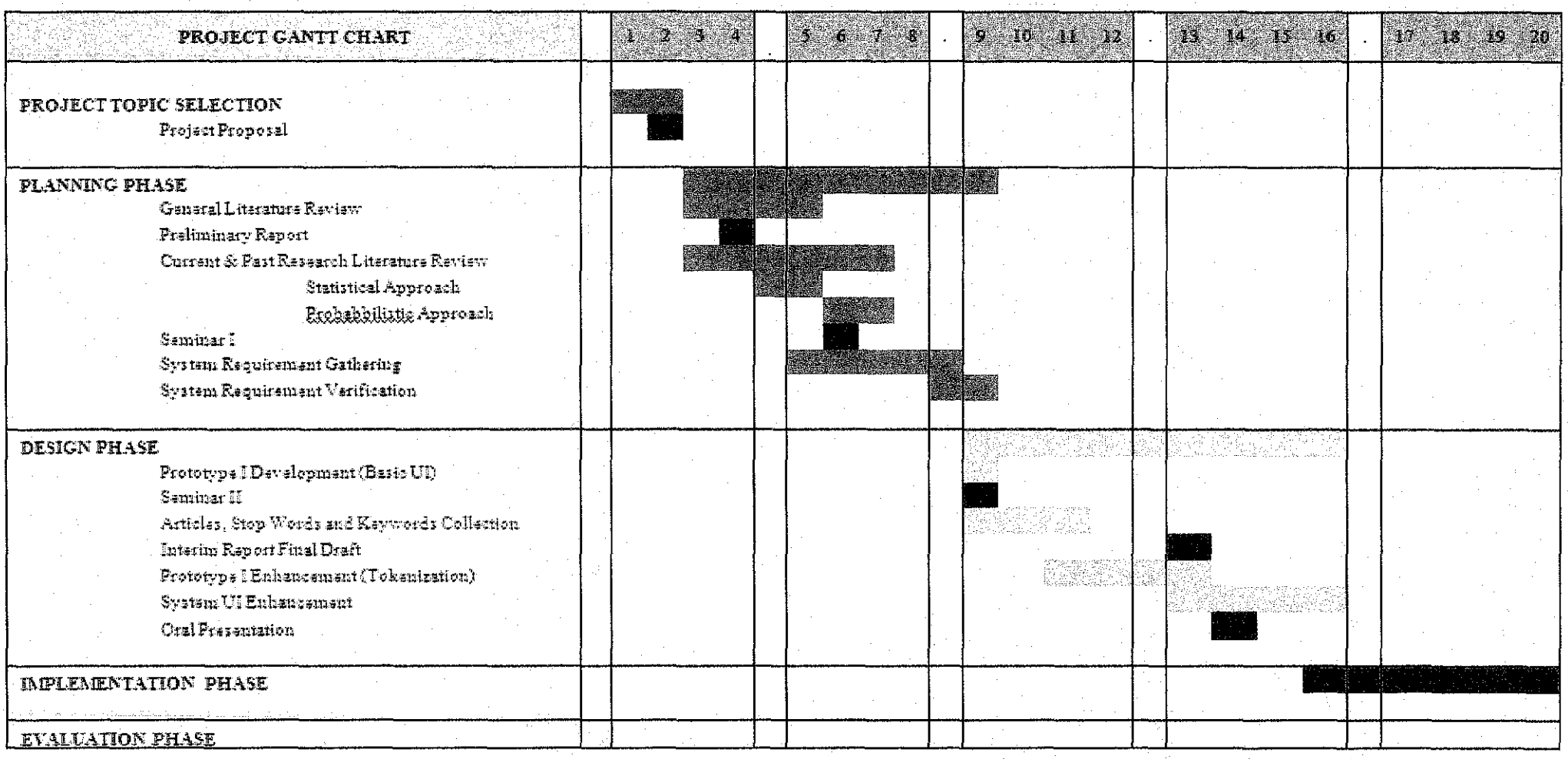


Figure 4.18: The average F-Score graph for Text Summarization System and Word Auto Summarizer using Article 11

APPENDIX H

PROJECT GANTT CHART: PART I



PROJECT GANTT CHART: PART II