

**Information Retrieval – using Porter Stemming Algorithm**

by

**Zurida Azita binti Zulkifly**

**Dissertation submitted in partial fulfillment of  
the requirements for the  
Bachelor of Technology (Hons)  
(Business Information Systems)**

**OCTOBER 2006**

**Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan**

t

TK

7855

.1141

296

2006

1) Algorithms

2) 7855 -- then

## **CERTIFICATION OF APPROVAL**

**Information Retrieval -- using Porter Stemming Algorithm**

by

Zurida Azita binti Zulkify

Dissertation submitted to the  
Business Information Systems Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
BACHELOR OF TECHNOLOGY (Hons)  
(BUSINESS INFORMATION SYSTEMS)

Approved by,

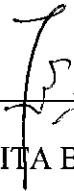
---

(Miss Eliza bt. Mazmee Mazlan)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK  
OCTOBER 2006

## **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 except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

ZURIDA AZITA BINTI ZULKIFLY

## **ABSTRACT**

Stemming is a process of removing or transforming endings (suffixes) when they are found on a word; inflectional endings (-s, -ing, -ed, etc) and derivational endings (-ion, -ative, -ity, -ment, -less, etc) and prefixes (un-, in-, etc). The rationale for using stemming is that similar words usually have similar meanings, so including words that are similar in meaning to those originally contained within it will increase the retrieval process effectiveness. There are many stemming methods that have been developed. However, the main focus of this project is on Porter Stemming Algorithm which has been developed by M.F Porter in 1980. The objective of this project is to develop a system that will demonstrate the information retrieval using Porter Stemming Algorithm. Problem with information retrieval is to get documents that are relevant to users' queries. To measure the performance, there are two measurements, which are precision and recall. The scope of the project is to implement the original Porter Stemming Algorithm in the application to improve the precision and recall in the retrieving documents process. Even though there are many improvements that have been made to the Porter Algorithm, we will focus on the original algorithm in this project. The Porter Stemming algorithm had five phases, which in every phase have its own rules for stripping the suffixes. By implementing the algorithm, it is expected from the application to retrieve only documents that are relevant to the users' queries.

## **ACKNOWLEDGEMENT**

Most grateful to Allah that I finally able to complete my final year project.

I would like to express my gratitude and thanks to my supervisor, Miss Eliza for her advices, helps, coordination and supervision in making this Final Year Project a success. She assisted and guided throughout the research and implementation of the Final Year Project.

I also would like to convey my appreciation to my family and friends for assisted, guided and moral support during carrying out this project.

Many thanks also to all that contribute in completing the final year project. Without the supports, guidance and helps from all, this project would not be like what it is today.

## TABLE OF CONTENTS

|                        |   |           |
|------------------------|---|-----------|
| <b>ABSTRACT</b>        |   | <b>i</b>  |
| <b>ACKNOWLEDGEMENT</b> |   | <b>ii</b> |
| <b>CHAPTER 1:</b>      | <b>INTRODUCTION</b>                         | <b>1</b>  |
|                        | 1.1 Background of Porter Stemming Algorithm | <b>1</b>  |
|                        | 1.2 Problem Statement                       | <b>3</b>  |
|                        | 1.2.1 Problem identification.               | 3         |
|                        | 1.2.2 Significant of the project            | 4         |
|                        | 1.3 Objective                               | 4         |
| <b>CHAPTER 2:</b>      | <b>LITERATURE REVIEW</b>                    | <b>5</b>  |
| <b>CHAPTER 3:</b>      | <b>METHODOLOGY</b>                          | <b>7</b>  |
|                        | 3.1 Methods/Tasks                           | 7         |
|                        | 3.1.1 Planning                              | 8         |
|                        | 3.1.2 Analysis                              | 8         |
|                        | 3.1.3 Design                                | 8         |
|                        | 3.1.4 Implementation                        | 10        |
|                        | 3.2 Tools                                   | 10        |
|                        | 3.3 Expectation of Algorithm                | 11        |
| <b>CHAPTER 4:</b>      | <b>RESULT AND DISCUSSION</b>                | <b>12</b> |
|                        | 4.1 Logical Design                          | 12        |
|                        | 4.1.1 Use Case Diagram                      | 12        |
|                        | 4.1.2 Data Flow Diagram                     | 13        |
|                        | 4.1.3 Entity Relationship Diagram           | 14        |
|                        | 4.2 User Design Interface                   | 15        |
|                        | 4.3 Limitations                             | 17        |
| <b>CHAPTER 5:</b>      | <b>CONCLUSION AND RECOMMENDATION</b>        | <b>18</b> |
|                        | 5.1 Relevancy to the objectives             | 18        |
|                        | 5.2 Recommendation                          | 18        |
| <b>REFERENCES</b>      |   | <b>19</b> |
| <b>BIBLIOGRAPHY</b>    |   | <b>20</b> |
| <b>APPENDICES</b>      |   | <b>21</b> |

## **LIST OF FIGURES**

- Figure 3-1 Phases in System Development Life Cycle
- Figure 4-1 Use Case Diagram
- Figure 4-2 Data Flow Diagram
- Figure 4-3 Attributes of document
- Figure 4-4 Main screen of the system
- Figure 4-5 Display the relevant documents in space provided
- Figure 4-6: Open the selected document in new form

## **LIST OF APPENDICES**

Appendix 1 Gantt chart

Appendix 2 Pseudo code of Porter's Algorithm

Appendix 3 Steps in Porter Stemming Algorithm



# **CHAPTER 1**

## **INTRODUCTION**

Stemming is a process of removing or transforming endings (suffixes) when they are found on a word; inflectional endings (-s, -ing, -ed, etc) and derivational endings (-ion, -ative, -ity, -ment, -less, etc) and prefixes (un-, in-, etc). Stemming is used to improve precision and recall in information retrieval. The stemming is meant to address problems of matching word in query to variant in document or vice versa, storage overhead by reducing number of distinct index terms and inconsistent use of descriptive vocabulary. The rationale for using stemming is that similar words usually have similar meanings, so including words that are similar in meaning to those originally contained within it will increase the retrieved effectiveness.

### **1.1 Background of Porter Stemming Algorithm**

The task of an Information Retrieval System is to retrieve or texts with information content that is relevant to a user's information need. Document retrieval subsumes two related, but different activities: indexing and searching. Indexing refers to the way documents, i.e. the items in the file, and requests, i.e. expressions of the user's information need, are represented for retrieval purpose. Searching refers to the way the file is examined and the items in it are taken as related to a search query. [1]

In evaluating the retrieval effectiveness in terms of the relevant items that are retrieved, the two most common measures of performance are recall and precision. Recall is the percentage of the relevant items that are retrieved in a search meanwhile precision is the percentage of the items that are retrieved in a search.

There are many techniques in getting relevance document. One of them are using conflation algorithm, a computational procedure that reduces variants of word to a single form. The rationale for such procedure is that similar words generally have similar meanings and, thus, retrieved effectiveness may be increased if the query is expanded by including words that are similar in meaning to those originally contained within it. [1]

The most common conflation procedure is the use of a stemming algorithm. It is an algorithm which reduces all the words with same root to a single form by stripping the root of its derivational and inflectional affixes. Word stemming is easy to implement and provides a highly effective means of conflating morphological variants. [1]

In this project, the focus is on original Porter's Stemming Algorithm, even there are many improvements have been made to the algorithm. There are several reasons for the popularity of the Porter algorithm: it is conceptually very simple; it seems to work at least as well as other, more complex algorithm; and the original paper provides a sufficiently detailed description to enable it to be implemented easily. [1]

Porter Stemming Algorithm will remove various suffixes such as -ED, -ING, -ION, -IONS to leave the word to single stem. For example, words CONNECTED, CONNECTING, CONNECTION, CONNECTIONS will become CONNECT after the suffixes removal. This stripping process will reduce the size and complexity of the data in the system. [2]

Basically, there are five (5) steps in Porter stemming algorithm. In Step1, it deals with plurals and past participles. Step 2, it will get the word in certain suffixes such as -ATIONAL and replace with -ATE, -TIONAL replace with -TION and -ENCI replace

with –ENCE. In step 3, it will look up for another suffix for the word such as –ICATE will transform to –IC, –ATIVE transform to NIL. In step 4, it will check another suffix such as –AL and –ANCE replace to NIL. In step 5, word with end –E will replace NIL and end –LL will be replaced –L. [3]

## **1.2 Problem Statement**

The primary goal of Information Retrieval (IR) is to get all documents which are relevant to user query, while retrieving as few non-relevant documents as possible. The central problem of Information Retrieval is the analysis and measurement of the relevance of the stored information, for example the relation between requested information and retrieved information. It is difficult to extract the information from the text and using it to decide whether each document is relevant or not to particular request. Other problematic areas in information retrieval are closely related with relevance problem involve the measure of precision and recall. Precision is the text that presented as an answer to an inquiry should contain only relevant information. In other hand, recall is all texts that containing relevant information should be found and presented as an answer to an inquiry.

### **1.2.1 Problem identification**

Achieving recall and precision – it is difficult to decide either the document is related or not with request. The problem involves the precision and recall. Precision is the percentage of items that are retrieved in a search and recall is the percentage of the relevant items that are retrieved in a search.

### **1.2.2 Significant of the project**

Performance – the Porter stemmer works reduces all the words with same root to a single form by stripping the root of words derivational and inflectional affixes. This will reduces the total number of terms in the Information Retrieval system, hence reduce the size and complexity of the data in the system. This will improve the performance of the information retrieval by increasing the precision and recall. This will give the user accurate documents.

### **1.3 Objective**

The objective of this project is to develop an application that demonstrates the information retrieval using the Porter's Stemming Algorithm for English words. This application is able to stem the words to the root word by stripping the suffixes. After the stemming process, the application will retrieve documents that relevant to the user query.

## **CHAPTER 2**

### **LITERATURE REVIEW**

Since 1940s, the problem of information storage and retrieval has attracted increasing attention. As the amount of information increase, it is becoming more difficult to get accurate and speedy access. When high speed computers become available for non-numerical work, many thought computers can 'read' the whole document collection to extract relevant information. However, it is difficult to duplicate human process of 'reading'. 'Reading' involves attempting to extract the information from the text and using it to decide whether each document is relevant or not to a particular request. The main problem rises are how to extract the information and also how to use it to decide relevance. [4]

In order to counter this problem, there are various stemming algorithms have been developed to increase Information Retrieval System's efficiency. The rationale of using stemming is that similar words generally have similar meanings. By including the words that are similar with to those originally contained within the users query, it will increase the retrieval process effectiveness. This project will concentrated on Porter stemming algorithm which is developed by M.F. Porter in 1980. This is the most widely used algorithm in Information Retrieval.

The Porter stemming algorithm is a process for removing the commoner morphological and inflexional endings from words in English.[5] The algorithm will removed suffixes of word to get the conflated it into a single term. By conflated the word into a single term, it will reduce the total number in the Information Retrieval system, and hence

reduce the size and complexity of the data in the system. The algorithm comprises of five rules each dedicated to handling certain kinds of word transformations. A given word's suffix is checked against each rules sequentially until it matches one of the rules, and consequently the conditions in the rule are tested on the word may result in suffix removal or modification. For example, words 'CONNECT', 'CONNECTED', 'CONNECTING', 'CONNECTION', and 'CONNECTIONS' will be reduced to CONNECT by removal of the various suffixes -ED, -ING, -ION, IONS.

The Porter stemming algorithm has long been recognized as a rather simple, computationally inexpensive and successful technique to bring together the words conveying the same or similar meaning and treat them as the same content contributors. However, in some cases, the algorithm did not conflate related words into a same common stem word (i.e. DEEPENINGS conflated to DEEPEN, while DEEP stayed DEEP. Also, RELATEDNESS conflated to RELATED, while RELATED transformed into RELAT).

## CHAPTER 3

### METHODOLOGY

#### 3.1 Methods / Tasks

For this project, the System Development Life Cycle (SDLC) method is used as a guideline to develop the system. Generally, the system development life cycle has four phases. There are: Planning, Analysis, Design and Implementation.

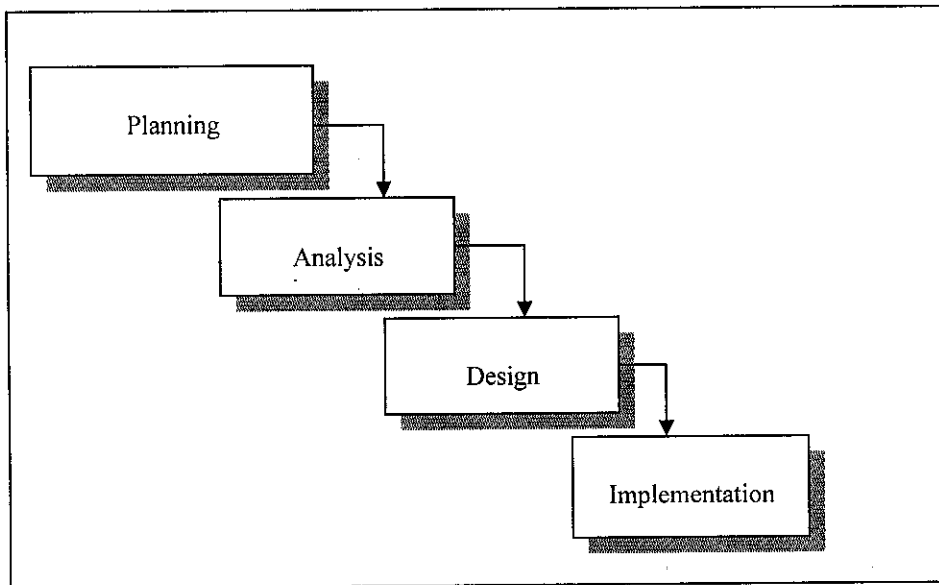


Figure 3-1: Phases in System Development Life Cycle

### **3.1.1 Planning**

In this stage, the title for Final Year Project is identified and proposal is made. The discussion with supervisor has been done in order to get clearer view of the project. The schedule of the project for two semesters has been planned. Please refer the Gantt chart in Appendix 1.

### **3.1.2 Analysis**

For this stage, the initial investigation and requirement gathering has been done. Research and literature review are done in order to understand the Porter's stemming algorithm. The tools that will be used for this project also have been identified. Besides, in this stage, the system flow also has been identified. In this project, there is only one user which is the one who will search for the document. The user will key in the keyword to search the document and click the submit button. By clicking the button, it will trigger background process where the system will use the Porter Stemming algorithm to stem the keyword. After that, the abstract for the documents that related to the keyword will be displayed and the users can choose specific document to review full document.

### **3.1.3 Design**

There are four (4) main designs that have been done in this stage which is logical design, study the pseudo code of the algorithm, Graphical User Interface (GUI) design and physical design.



### *Logical Design*

The modeling model that is use for the logical design is UML Modeling. The use case and data flow diagram was developed based on the system flow. This is to make a clearer presentation of the system flow. Besides, the Entity Relationship Diagram was developed to present the database. In this project, the database was developed using Microsoft Office Access 2003. The document was stored in the database using OLE object.

### *Study Pseudo code for the Algorithm*

The main backbone for the system is the stemming process. In order to develop the system, the pseudo code has been studied to understand the algorithm. There are five stages in the algorithm based on Porter's stemming algorithm. In every phase, the algorithm will remove some of the prefixes based on the rules.

### *Graphical User Interface (GUI) design*

In this system, there is one (1) screen. The screen has the text box and search button. In this screen, the user will key in the keyword for the query and click the search button. Then the system will process the keyword and display the result in the space provided. The list of related document will be displayed. The user has option to open the document by clicking the list.

### *Physical Design*

In this phase, the task is to convert the logical designs to physical designs. The Entity Relationship Diagram was converted into the database. From the pseudo code, the code for the application was applied. The Graphical User Interface is developed.

### **3.1.4 Implementation**

For the last phase, all the designs that have been developed in the design phase were implemented. The database, Graphical User Interface and the real program were developed based on respective design.

## **3.2 Tools**

### **Hardware:**

- Intel® Core™ Duo Processor T2400 - 1.83GHz
- 0.99 GB of RAM
- 80 GB hard disk
- Microsoft Windows XP Home Edition Version 2002 Service Pack 2

### **Software:**

#### **Development tool:**

Microsoft Visual Basic .NET

#### **Database:**

Microsoft Office Access 2003

#### **Other tools:**

Microsoft Office Project 2003

Microsoft Office Word 2003

Microsoft Office Visio 2003

### 3.3 Expectation of Algorithm

As the system using Porter stemming algorithm, it will stem words or query that inserted or requested by the user. From the stem word, it will match against the document title. After relevant document is retrieved, the system will ranked the document based on stem words recall in the document. For example, if in the document A the stem word is recall 5 times, in document B the stem word recall is 15 times, and in document C the stem word recall is 10 times, the list of relevant document will be list as:

1. Document B
2. Document C
3. Document A

## CHAPTER 4

### RESULT AND DISCUSSION

#### 4.1 Logical Design

##### 4.1.1 Use Case Diagram

Use case diagram is designed based on the analysis that has been done in analysis phase. The Figure 4-1 shows the interaction between users and the system. Basically, when the user keys in the query, the system will search relevant document in the database. From the list of document provided, the user can select and open the document.

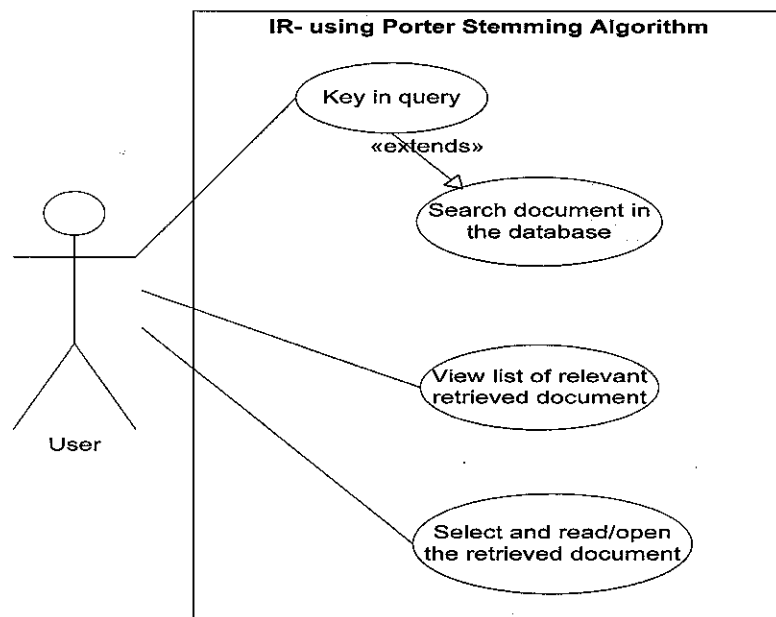


Figure 4-1: Use Case Diagram

Actor: Users of the system

Description: The users of the system will key in the query. By clicking a search button, the system will stem the query and match the word with the title of document in the database. Then it will display the list of document that satisfies the user's query. The users can click at the title of document to open or view the document.

#### 4.1.2 Data Flow Diagram

Figure 4-2 illustrates the flow of data through the system. In data flow diagram, basically there are three things; the input, process and output. The input is the user's query. The process is the system read the query. Then the system will stem the query using Porter Stemming Algorithm and retrieve the relevant document. The output is the system will list the relevant documents to users.

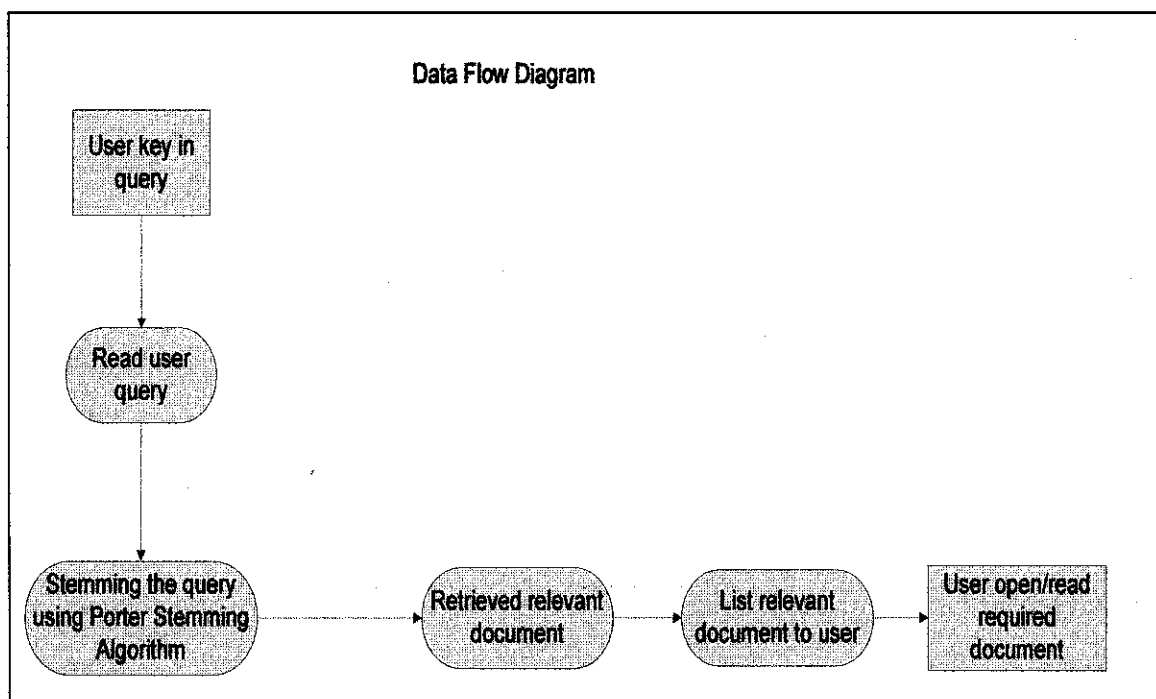


Figure 4-2: Data Flow Diagram

### 4.1.3 Entity Relationship Diagram

In this system, documents will be stored in the database. The Figure 4-3 below shows the attributes of the documents.

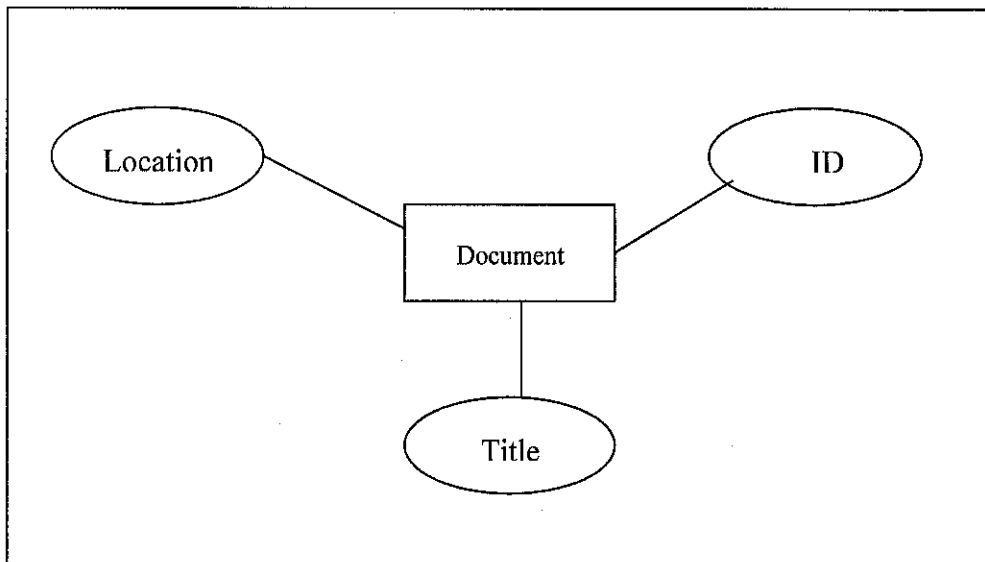


Figure 4-3: Attributes of document

#### Attributes:

**ID:** the ID for every document. It will help in searching the document faster as it is unique.

**Title:** the title of every document. The system will compare the stem word with the title to find the relevant documents.

**Location:** the location of the document. It is for the system to identify the location of the document before opening the document for the users.

## 4.2 User Design Interface

Figure 4-4 below is the print screen of the system. To find relevant document, the user needs to key in the query in the box or space provided. Next, the user clicks the 'Search' button. The system will stem user query. Then it will match the stem word with title in the database. After retrieve the relevant documents, it will display the list in the list view, as shown in Figure 4-5. When the user double clicks the selected item, the document will be opened in new form as shown in Figure 4-6.

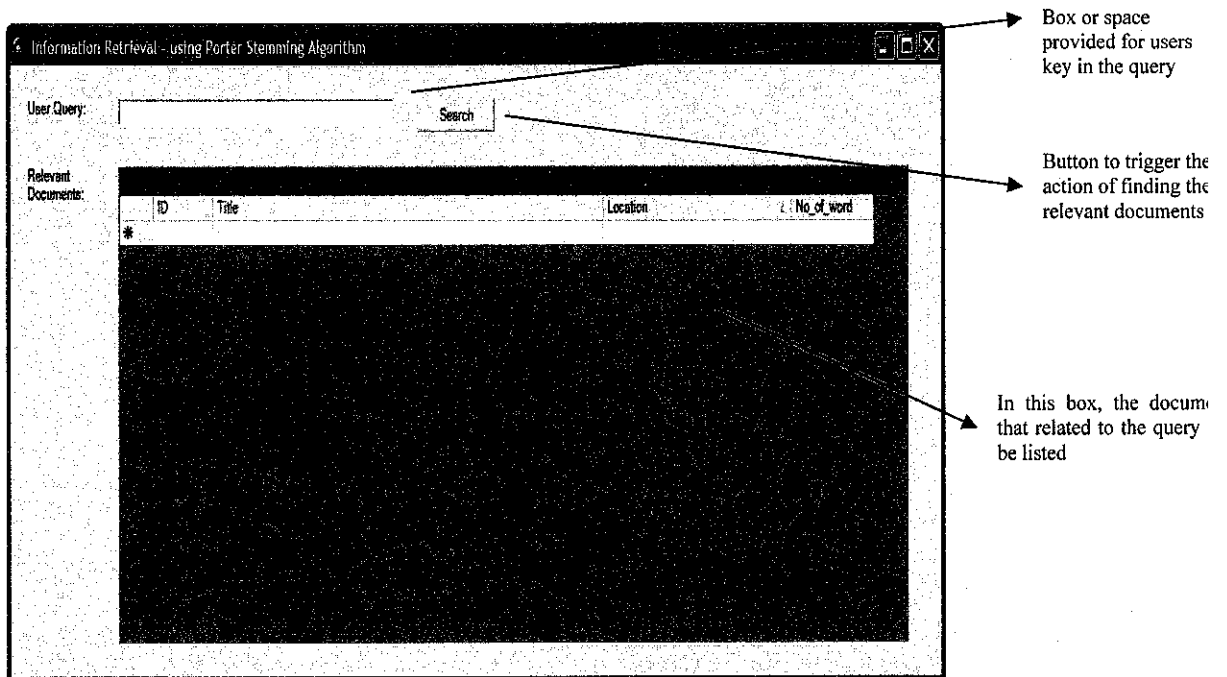


Figure 4-4: Main screen of the system

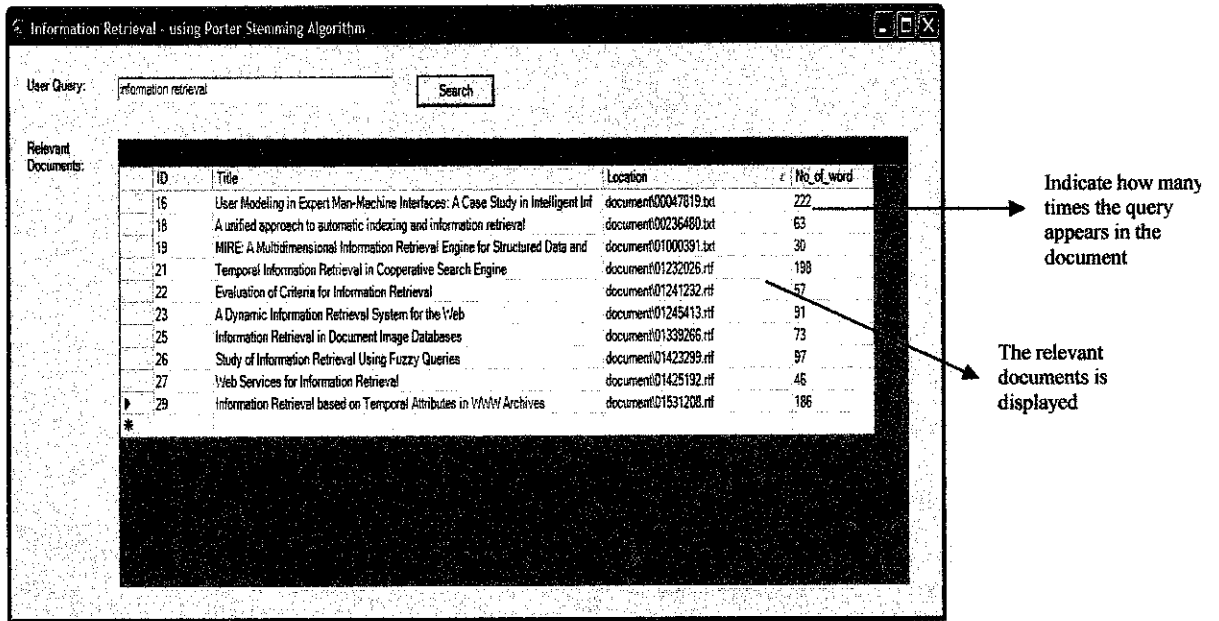


Figure 4-5: Display the relevant documents in space provided

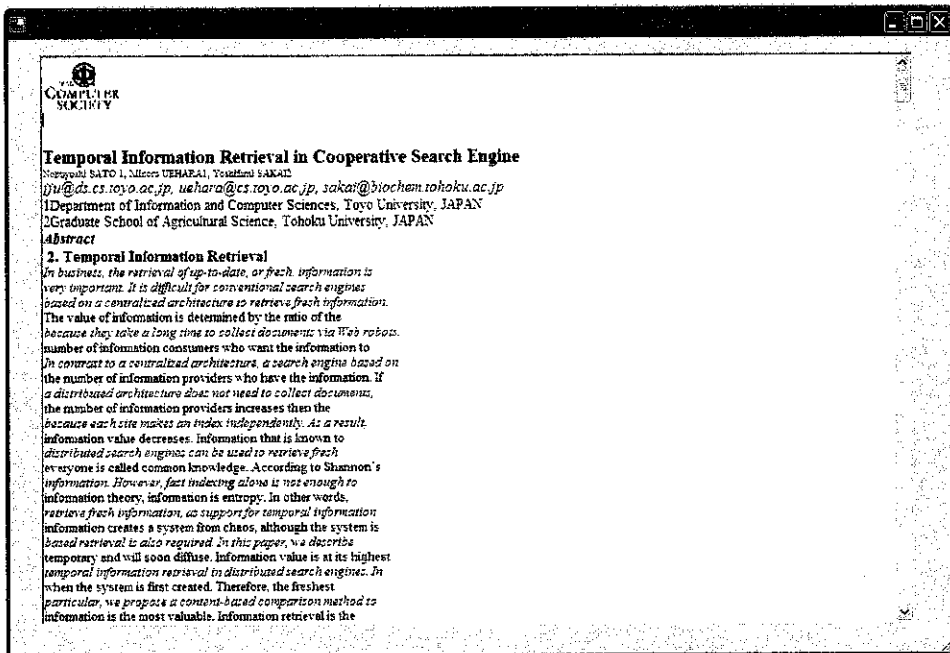


Figure 4-6: Open the selected document in new form



Steps in searching relevant documents:

1. The user key in the keyword or query in the box or space provided. For example the user key in “Information retrieval”.
2. Press ‘Search’ button. The system will stem the query using Porter Stemming Algorithm. In this case, the stem word is ‘inform’ and ‘retriev’. Using the stem word, the system will find relevant documents from the database.
3. The system will list the relevant documents in the box provided. The user can double click at the document’s name to open the file.
4. The document will be opened in another form.

### **4.3 Limitations**

In this system, as it is developed using Microsoft Visual Basic .NET and Microsoft Access, there are some limitation in the system. In terms of document, the document that can be opened by the system is only file with .txt and .rtf extension. Besides, there are only 30 documents in the database; therefore the effectiveness of the system in term of speed cannot be measure for large amount of documents.

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1 Relevancy to the objectives**

Referring to objective of this project, it can be concluded that the system that have been developed have meet the objectives. This system have demonstrates the information retrieval using the Porter's Stemming Algorithm for English words. This system is able to stem the query inserted by the user to the root word by stripping the suffixes. After the stemming process, this system has retrieved the relevant documents to the user query. The users also have option to view the documents which in .txt or .rtf extension files.

#### **5.2 Recommendation**

As in current system, the file with .txt and .rtf extension only can be opened. For future development, the application can be developed using another platform where it can opened PDF file. Besides, the system can be enhanced by increased the number of the documents in the database to check the speed of the system.

## REFERENCES

- [1] Karen Spark Jones & Peter Willet, *Readings in Information Retrieval*, Morgan Kaufmann Publishers Inc., 1997
- [2] M.F. Porter, 1997, *An algorithm for suffix stripping*, Computer Laboratory, Corn Exchange Street, Cambridge
- [3] Term Processing & Normalization  
Available Online:  
[http://66.102.7.104/search?q=cache:MK5ImavIE8sJ:porta.informatik.uni-freiburg.de/lectures/InformationRetrieval/2006SS/Slides/02\\_2\\_Basics-2.ppt+phases+of+Porter+algorithm&hl=en&gl=my&ct=clnk&cd=2](http://66.102.7.104/search?q=cache:MK5ImavIE8sJ:porta.informatik.uni-freiburg.de/lectures/InformationRetrieval/2006SS/Slides/02_2_Basics-2.ppt+phases+of+Porter+algorithm&hl=en&gl=my&ct=clnk&cd=2)
- [4] Information Retrieval  
Available Online: <http://www.dcs.gla.ac.uk/Keith/Chapter.1/Ch.1.html>
- [5] The Porter Stemming Algorithm  
Available Online: <http://www.tartarus.org/martin/PorterStemmer/>

## BIBLIOGRAPHY

*CSc352 Information Retrieval*, Lancaster University

Available Online:

<http://www.comp.lancs.ac.uk/computing/users/paul/CSc352/slides/lecture14and15%20Word%20Conflation%206%20UP.pdf>

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley Longman

Available Online:

[www.sims.berkeley.edu/~hearst/irbook/porter.html](http://www.sims.berkeley.edu/~hearst/irbook/porter.html)

Jeffrey J. Tsay, 2004, Second Edition. *Visual Basic.NET Programming. Business Applications with a Design Perspective*, New Jersey, Prentice Hall, Inc

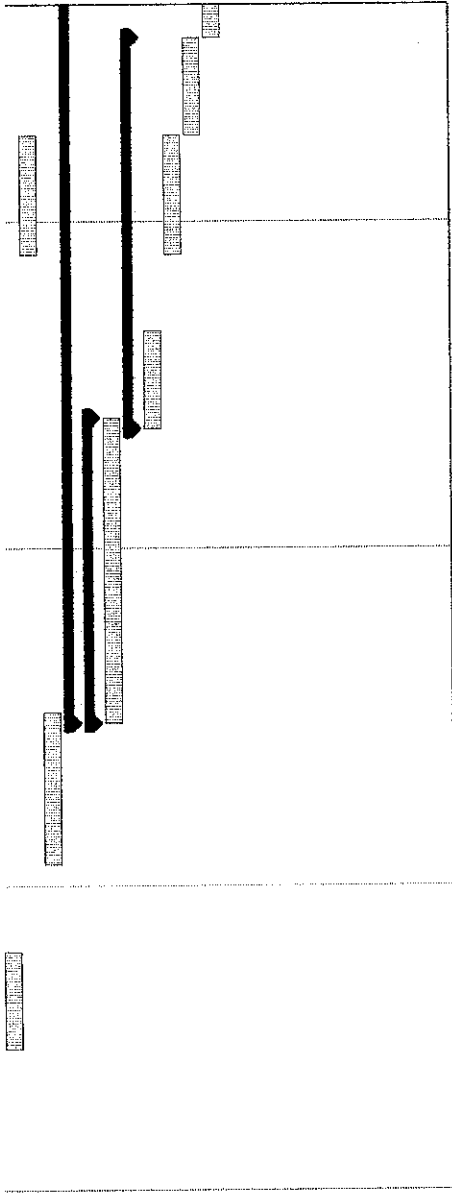
Alan Dennis, Barbara Haley Wixom, David Tegarden, 2002, *Systems Analysis and Design An Object-Oriented Approach with UML*, US, John Wiley & Sons

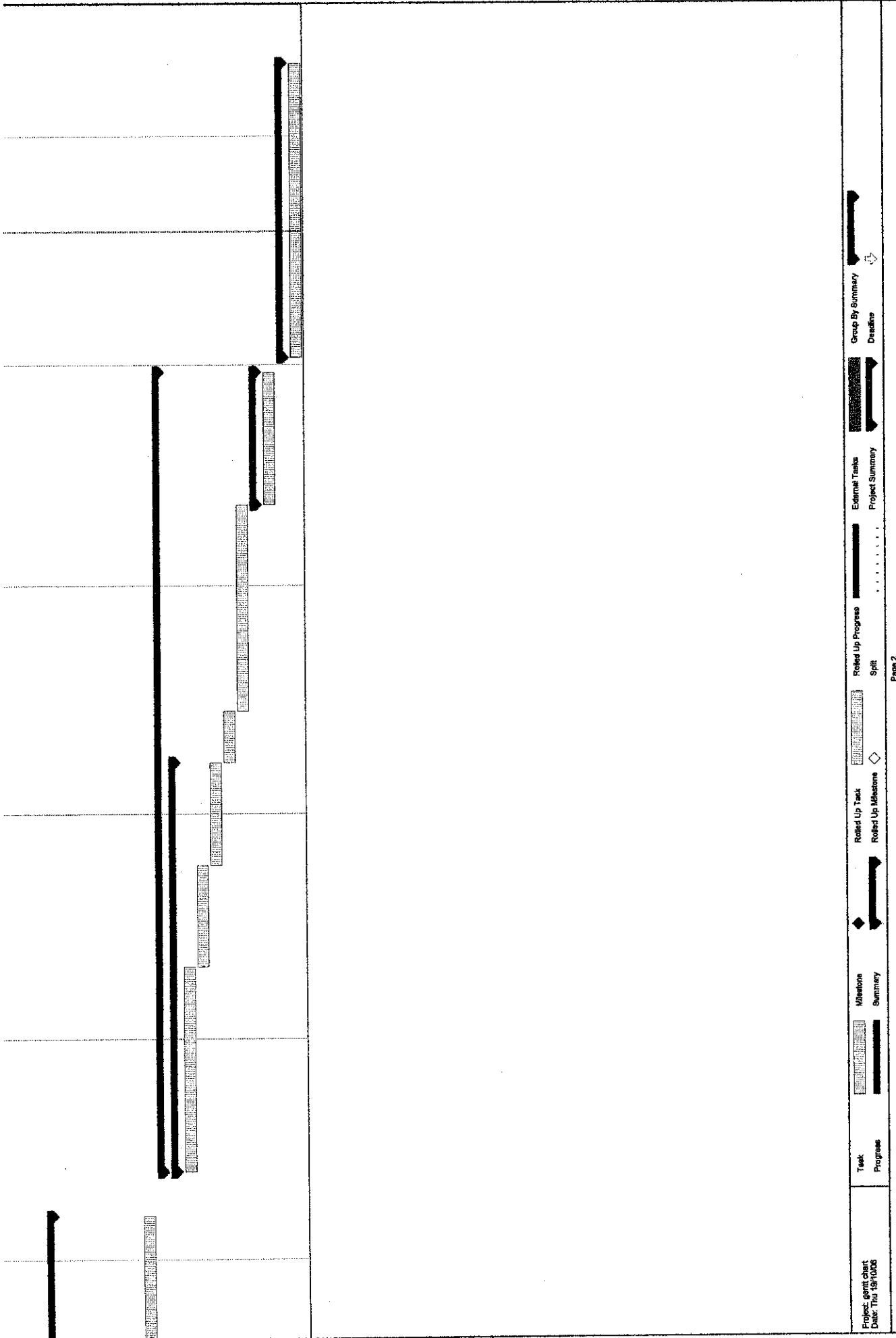
## **APPENDICES**

# **APPENDIX 1**

Gantt chart

|    |   |              |         |              |
|----|---|--------------|---------|--------------|
| 5  | Undersand the Project Topic                       | Tue 7/4/2006 | 7 days  | Wed 22/2/206 |
| 6  | Research on Project Topic                         | Fri 26/3/06  | 7 days  | Mon 18/7/06  |
| 7  | Work on Preliminary Report                        | Fri 19/3/06  | 10 days | Thu 13/7/06  |
| 8  | Analysis  | Tue 14/3/06  | 89 days | Tue 8/8/06   |
| 9  | Develop Analytic Plan                             | Thu 16/3/06  | 20 days | Wed 12/4/06  |
| 10 | Requirements Gathering                            | Thu 16/3/06  | 20 days | Wed 12/4/06  |
| 11 | User-Case Modeling                                | Wed 12/4/06  | 28 days | Wed 17/5/06  |
| 12 | Develop User Cases                                | Wed 12/4/06  | 7 days  | Thu 20/4/06  |
| 13 | Develop Flowchart                                 | Fri 26/4/06  | 7 days  | Mon 18/5/06  |
| 14 | Develop Entity Relationship Diagram               | Tue 06/5/06  | 7 days  | Wed 17/5/06  |
| 15 | Interim Report                                    | Thu 18/5/06  | 14 days | Tue 06/6/06  |
| 16 | Design  | Tue 13/6/06  | 79 days | Fri 20/8/06  |
| 17 | System Architecture Design                        | Tue 13/6/06  | 40 days | Mon 07/8/06  |
| 18 | Work on read file from database                   | Tue 13/6/06  | 20 days | Mon 05/7/06  |
| 19 | Work on read the content of file                  | Tue 11/7/06  | 10 days | Mon 24/7/06  |
| 20 | Develop Pseudo code                               | Tue 25/7/06  | 10 days | Mon 07/8/06  |
| 21 | User Interface Structure Design                   | Tue 08/8/06  | 5 days  | Mon 14/8/06  |
| 22 | Manage Programming                                | Tue 15/8/06  | 20 days | Mon 11/9/06  |
| 23 | Object Persistence Design                         | Tue 12/8/06  | 14 days | Fri 20/8/06  |
| 24 | Connecting from logical design to physical design | Tue 12/8/06  | 14 days | Fri 20/8/06  |
| 25 | Implementation                                    | Mon 02/10/06 | 30 days | Fri 10/11/06 |
| 26 | Construction                                      | Mon 02/10/06 | 30 days | Fri 10/11/06 |





Project: split chart  
Date: Thu, 19/10/06

Task  
Progress

Milestone  
Summary

Rolled Up Task  
Rolled Up Milestone

Rolled Up Progress  
Split

External Tasks  
Project Summary

Group By Summary  
Deadline



## **APPENDIX 2**

Pseudo code's for Porter Stemming Algorithm

## Pseudo code of Porter Stemming Algorithm

% Phase 1: Plurals and past participles.

```
select rule with longest suffix {
  sses -> ss;
  ies -> i;
  ss -> ss;
  s -> NIL;
}
select rule with longest suffix {
  if ( (C)*((V)+(C)+)(V)*eed) then eed -> ee;
  if (*V*ed or *V*ing) then {
    select rule with longest suffix {
      ed -> NIL;
      ing -> NIL; }
    select rule with longest suffix {
      at -> ate;
      bl -> ble;
      iz -> ize;
      if ((C1C2) and (C1 = C2) and (C1 not in {l,s,z})) then C1C2 -> C1;
      if (((C)*((V)+(C)+)C1V1C2) and (C2 not in {w,x,y})) then C1V1C2 -> C1V1C2e; }
  }
}
```

if (\*V\*y) then y -> i;

% Phase 2

if ( (C)\*((V)+(C)+)(V)\* ) then

select rule with longest suffix {

```
  ational -> ate;
  tional -> tion;
  enci -> ence;
  anci -> ance;
  izer -> ize;
  abli -> able;
  alli -> al;
  entli -> ent;
  eli -> e;
```

ousli -> ous;

```
  ization -> ize;
  ation -> ate;
  ator -> ate;
  alism -> al;
  iveness -> ive;
  fulness -> ful;
  ousness -> ous;
  aliti -> al;
  iviti -> ive;
  biliti -> ble; }
```

% Phase 3

if ( (C)\*((V)+(C)+)(V)\* ) then

select rule with longest suffix {

```
  icate -> ic;
  ative -> NIL;
  alize -> al;
  iciti -> ic;
  ical -> ic;
  ful -> NIL;
  ness -> NIL; }
```

% Phase 4

if ( (C)\*((V)+(C)+)((V)+(C)+)(V)\* ) then

select rule with longest suffix {

```
  al -> NIL;
  ance -> NIL;
  ence -> NIL;
  er -> NIL;
  ic -> NIL;
  able -> NIL;
  ible -> NIL;
```

```
ant -> NIL;
ement -> NIL;
ment -> NIL;
ent -> NIL;
ou -> NIL;
ism -> NIL;
ate -> NIL;
iti -> NIL;
ous -> NIL;
ive -> NIL;
ize -> NIL;
if (*s or *t) then ion -> NIL; }
% Phase 5
select rule with longest suffix {
  if ( (C)*((V)+(C))*((V)+(C))+*(V)*) then e -> NIL;
  if (((C)*((V)+(C)+(V)*) and not (( *C1V1C2)
    and (C2 not in {w,x,y}))) then e -> nil; }
if ( (C)*((V)+(C))*((V)+(C))+V*ll) then ll -> l;
```

## **APPENDIX 3**

### Steps in Porter Stemming Algorithm

## Steps in Porter Stemming Algorithm

