

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

THE APPLICATION OF MACHINE LEARNING TECHNIQUE (NAÏVE
BAYES) FOR AUTOMATIC TEXT SUMMARIZATION
[THE CASE OF AMHARIC NEWS TEXTS]

By

TEFERI ANDARGIE ACHENEF

*A thesis submitted to
the School of Graduate Studies of Addis Ababa University
in partial fulfillment of the requirements for the Degree of Master of Science in
Information Science*

June 2005

ADDIS ABABA UNIVERSITY
LIBRARIES
PC BOX 1176
ADDIS ABABA - ETHIOPIA

አዲስ አበባ ዩኒቨርሲቲ
ADDIS ABABA UNIVERSITY
FACULTY OF INFORMATICS
PHOTOGRAPHIC LAB

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
Faculty of Informatics
Department of Information Science

THE APPLICATION OF MACHINE LEARNING (NAÏVE BAYES) APPROACH FOR
AUTOMATIC SUMMARIZATION OF AMHARIC NEWS TEXT

BY
TEFERI ANDARGIE

Name and Signature of Members of the Examining Board

Prof. B. R. Krishina Rao, Chairman, Examining Board

Ato Tesfaye Biru, Advisor

Dr. Nega Alemayehu, Examiner

Chairman, Faculty

Signature

Date

Chairman, Graduate Council

Signature

Date

ADDIS ABABA UNIVERSITY
LIBRARIES
P.O. BOX 1176
ADDIS ABABA, ETHIOPIA

DEDICATION

To
My Lord and Savior Jesus Christ

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my Advisor, Ato Tesfaye Biru for his constructive comments and guidance throughout this study. I am also deeply grateful to Dr. Lars Asker for his invaluable comments during this study.

I offer my special thanks to my family and friends who have encouraging me during my study. I want to mention my great thanks to Eoyel Brihane for his support during the implementation of the prototype.

I am also grateful to my classmates for their unreserved help and my special thanks go to Yalemsew Mitesinot, Yoseph Shiferaw, and Seifu Tatek for their comments and support during this study. I wish also to express my warm and sincere thanks to W/ro Ethiopia Tadesse, Head of the Department, for her help and encouragement during this study.

Finally, I want to offer my thanks to all the department members who have contributed in one way or another for the completion of this thesis.

Teferi Andargie

ABSTRACT

This study presents an approach to automatic summarization of Amharic news texts by extracting sentences in a given document. The objective of this study is to investigate the application of machine learning technique (naïve Bayes method) to automatic summarization of Amharic news items. The focus is on how to use the naïve Bayes classifier for automatic Amharic news text summarization to extract sentences, i.e. on how to train the naïve Bayes to classify sentences from Amharic news texts.

First each sentence is represented by a set of predefined features (attributes) (i.e. location of a sentence in a document, title words occurring in the sentence, and cue words occurring in the sentence) that Edmundson (1969) found as a good indicator in giving an optimum summary for scientific papers. In addition, the thematic words occurring in the sentence. Then the naïve Bayes algorithm is used to train to classify sentences as "a summary" and "not - a summary" based on the feature vectors.

For the purpose of this study 480 Amharic news articles is used. Evaluation of the results of the experiments is done using 10-fold cross validation. Result of the experiment shows that the location feature gives the best result in the classification of sentences when using individual features. The results of different combinations of feature sets in which location feature is included shows better results than when location is not included.

Based on the feature values estimated on the training program for the combination of all the features a prototype summarizer is developed which extracts sentences to a desired compression level.

TABLE OF CONTENTS

DEDICATION.....	I
ACKNOWLEDGEMENT.....	II
ABSTRACT.....	III
TABLE OF CONTENTS.....	V
LIST OF TABLES, FIGURE, AND APPENDICES.....	VII
 CHAPTER ONE.....	 1
INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.1.1. What is Automatic Text summarization?.....	1
1.2. STATEMENT OF THE PROBLEM AND JUSTIFICATION OF THE STUDY.....	4
1.3. OBJECTIVES OF THE STUDY.....	7
1.3.1. General Objective.....	7
1.3.2. Specific objectives.....	7
1.4. METHODOLOGY.....	8
1.4.1. Related literature review.....	8
1.4.2. Dataset used for the experiment (The Corpus).....	8
1.4.3. Data Preprocessing.....	9
1.4.4. Design Tools.....	10
1.4.5. Test Plan.....	10
1.4.6. Experimentation.....	11
1.4.7. Evaluation of models developed.....	12
1.4.8. Prototype Development.....	13
1.5. SIGNIFICANCE OF THE STUDY.....	14
1.6. SCOPE AND LIMITATION OF THE STUDY.....	15
1.7. ORGANIZATION OF THE THESIS.....	16

CHAPTER TWO.....	17
AUTOMATIC TEXT SUMMARIZATION	17
2.1. BASIC CONCEPT	17
2.2. THE AUTOMATIC TEXT SUMMARIZATION PROCESS	19
2.3. TYPES OF SUMMARIES.....	21
2.3.1. Extract versus Abstract Summaries	21
2.3.2. Single versus Multi Document Summaries	21
2.3.3. Indicative Summaries versus Informative Summaries	22
2.3.4. Query oriented versus Generic Summaries.....	22
2.4 APPROACHES TO TEXT SUMMARIZATION	23
2.5. REVIEWS ON RELATED AUTOMATIC TEXT SUMMARIZATION STUDIES	25
2.5.1. The Automatic Creation of Literature Abstracts	25
2.5.2. New methods in automatic extraction.....	26
2.5.3. A trainable document summarizer	27
2.5.4. The Rhetorical parsing, Summarization, and Generation of natural language texts	28
2.5.5. A robust practical text summarizer	29
2.5.6 News Story Gisting at University College Dublin	29
CHAPTER THREE.....	31
MACHINE LEARNING APPROACH TO AUTOMATIC TEXT SUMMARIZATION	31
3.1. MACHINE LEARNING	31
3.2. MACHINE LEARNING BASED TEXT SUMMARIZATION	34
3.3. WHY MACHINE LEARNING TO AUTOMATIC TEXT SUMMARIZATION?	35
3.4. FEATURE SETS AND REPRESENTATIONS.....	36
3.5. SENTENCE SELECTIONS AS A FUNCTION OF CLASSIFICATION	39
3.6. TRAINING VERSUS TEST SETS	40
3.7. TYPES OF CLASSIFIERS	41
3.7.1. Naive Bayes.....	41

3.7.2. The Naïve Bayes classifier	44
3.8. EVALUATION METRIC	47
CHAPTER FOUR	48
EXPERIMENTATION	48
4.1. THE CORPUS	48
4.2. PREPROCESSING	50
4.2.1. Word boundary	50
4.2.2. Stop word and cue word list	51
4.2.3. Word Stemming	52
4.2.4. Amharic Fonts	52
4.3. FEATURE SETS	54
4.3.1. Feature values collection	54
4.4. THE NAÏVE BAYES CLASSIFIER	59
4.5. RESULT AND DISCUSSION OF THE EXPERIMENTS	60
4.6. DESIGN AND DEVELOPMENT OF PROTOTYPE SUMMARIZER	67
CHAPETR FIVE	71
CONCLUSIONS AND RECOMMENDATIONS	71
5.1. Conclusions	71
5.2. Recommendations	73
REFERENCES	74
DECLARATION	98

List of tables

Table 1. Confusion Matrix	61
Table 2. Evaluation performance of all the features	64
Table 3. Model prediction for unlabelled documents	66

List of figure

Figure 1. Design of the summarizer	68
--	----

List of Appendices

Appendix I: Database of news articles and summaries taken for the experiment.....	82
Appendix II: List of cue words used.....	86
Appendix III: List showing suffixes used for stemming	87
Appendix IV : List showing the symbols used in the Visual Ge'ez font for the Amharic fidel.....	88
Appendix V: Evaluation performance of individual features	91
Appendix VI: Evaluation performance of combined features.....	95

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

1.1.1 What is Automatic Text Summarization?

With the coming of information revolution, more and more information is available and accessible. As a result this brought problems to users in finding relevant information to their interests and needs in as short an amount of time as possible. Since it has become virtually impossible for any user to browse or read a large number of full documents without seeing their importance, it was considered to explore methods that allow users to locate and browse information within collection of documents.

Summarization of contents of documents is one of the key methods found for this problem (Constantin, et al. n.d; Harabagiu, et al., 2002). Summarization deals with a large amount of information by comprising only the essential part. Text summarization often occurs in everyday communication for example, in search engines to present short description of search results, news paper head lines, journal abstracts, and book reviews (Yona, 2003).

Text summary also has many possible interesting uses including aiding in information retrieval like building indexes, storing only a summarized version of a document instead of a full-text. Summaries may also be useful for mobile (small screen) devices, user-interest summaries for newswire content, i.e. summarizing all the news on a topic of interest to the user (Hassel, 2004).

In the past, text summaries were produced manually i.e. by human experts. For example, an abstract for scientific journal articles is usually done by the author. Automatic text summarization is one attempt to replace and save human effort required in performing manual summarization (Moens 2000).

Automatic text summarization is the task to produce the most important content from a given text document to the user in a condensed form and in a manner sensitive to the user's or application's need (Mani, et al., 2001; Chinchor et al., 1998).

The idea of producing summaries by automatic means is not new. When discussing automatic summarization most documents start with the work of H.P.Luhn (1958). His research was based on the idea of counting the frequency of words on the intuition that the most frequent words represent the most important concept of the text (this quantify as a good candidate to represent the text), disregarding the very high frequency common words. Thus, he proposed to assign a value to each word and this value, with in the context of a sentence, created significance value for that sentence. When it comes to summarization it represents the document.

There has been an increase in the research and development budgets devoted to automatic text summarization. For example the United States DARPA¹ has identified text summarization as a critical research area, and is investing in huge sum of money in the field (www1). To further advance and enables researchers to participate in large scale experiments; NIST² is performing an evaluation series in text summarization field, called the Document Understanding Conference³ (DUC).

Currently, it has become possible to commercialize this technology. There are some summarizers already commercially available in the market. These include Copernic (www2), Sinope (www3), and AutoSummarize, embedded as part of Microsoft Word. Several summarizers are available for different languages. For example, Dalianis and

¹ Defense Advanced Research Projects Agency

² National Institute of Standards and Technology

³ <http://duc.nist.gov/>

Hassel (2000) developed a summarizer for Scandinavian texts, Pachantouris (2004) for Greek text, and Mazdak (2004) for Persian language.

On the applicability of machine learning algorithms to develop automatic summarization a variety of them have been proposed including the decision tree algorithm, the naïve Bayesian classifier (Bayesian) and neural network learning algorithms (Kupiec et al., 1993; Teufel et al., 1999; Chuang and Yang, 2000). Kupiec, et al. consider the problem of part of a document extraction for summarization using a machine learning approach as a classification task and developed trainable summarization technique that is grounded in a Bayesian classifier rule.

Chuang and Yang (2000) also used supervised learning algorithms to train important sentence segment extractor which is experimented on U.S patent. In this experiment based on the feature vector, they found results that are comparable with other approaches (including Microsoft word summarizer) in terms of precision, recall, and classification accuracy.

1.2 STATEMENT OF THE PROBLEM AND JUSTIFICATION OF THE STUDY

The boom of electronic communication has resulted in the availability of electronic resources through the Internet in a rate that is becoming very difficult to keep up with. This growth in content of information is mainly attributable to the number of information service providers as well as the globalized nature of information demand. News is one such information that is highly characterized as an international information requirement and similarly service providers in this case is numerous. This being the case, access to relevant information would be difficult given the rate of production and sheer volume.

The local context here in Ethiopia is also following the same trend with the level of electronic media and publishing online increasing by the day. To mention a few, news service providers like Ethioindex, Ethiomedia, and Addiszena are some of the service providers updating their websites daily with news of all kinds in an average of 15-20 articles.

This coupled with local newspapers that are following the same trend in publishing their content, like the Ethiopian Reporter, make grounds important for a mechanism by which a potential user would be able to browse through contents of these sites without actually having to go through the entire content.

With the lack of filtering and summarization ^{problem} services that can greatly reduce the time and other resources spent while browsing, it can be fairly said users spend more time than they should browsing over content that they are not interested in. While filtering would enable to screen the desired content for a user in line with his/her interests, summarization would enable greatly enhanced and highlighted information to reach the user. Though clear lines can be drawn between the two, we can see they both aim to ^{rich}

present the user with relevant information without having to spend too much time looking for it.

Summarization of content, more specifically, enables users to decide if an article is interesting enough, to update their general awareness and even revise what has been learned. In light of these, summaries have been produced and are being produced for various types of information content, be it in business or academia, to enhance information access to users.

Its implementation so far however is mainly limited to manual approaches in almost all establishments locally as I have not been able to find automatic approaches being employed. Producing summaries manually may sound a feasible approach where the content of information is limited and where there exist trained and experienced personnel to develop it. However, as the amount of information and publication grows it becomes impossible for these people to keep producing at the rate and quality of output they are used to with limited content. In light of this,[¶] research towards automatic summarization of Amharic documents would be of great impetus in creating systems capable of providing information in desirable manner.

[¶] Automatic Amharic text summarization, for an area like news, could then be justified in light of the amount of news being generated. Research in the area has commenced and is currently underway in the Faculty of Informatics at Addis Ababa University,^{*} though no final outputs have yet been recorded.^{*} The methodology adopted in this study is different from the one that is still being carried out and mentioned above in that the method of assigning weights to the parts of documents extracted to form a summary is subjective in that study which is opposed to the machine learning approach which extracts weights from the training data.

This research mainly attempts to explore the applications of Machine Learning towards automatic text summarization. It aims to investigate how far trials generated by this

approach can actually help to generate summaries that can be able to satisfy user requirements.

1.3 OBJECTIVES OF THE STUDY

1.3.1 General Objective

The general objective of this research is to study the current works in the area of summarization and to investigate the application of Machine Learning techniques (specifically, the naïve Bayes method) to automatic summarization of Amharic News texts.

1.3.2 Specific Objectives

To achieve the aforementioned general objective, the following specific objectives are formulated.

1. Review literatures on automatic text summarization,
2. Review text summarization techniques with specific emphasis on machine learning,
3. Make a review of the characteristics of Amharic language to identify properties that are relevant for automatic summarization,
4. Collect Amharic news texts available in electronic format for making experiments on their summaries,
5. Build and train models using the naïve Bayes method,
6. Test the performance of the models,
7. Development of prototype Amharic news summarizer based on the result of the models.

1.4 METHODOLOGY

The following methods were employed in order to achieve the above stated objectives.

1.4.1 Literature Review

Review of relevant literatures was made to get further understanding of automatic summarization methods with specific emphasis to machine learning techniques that are used in English and other languages. Also the problems of Amharic language writing were studied in relation to summarization processes and techniques.

1.4.2 Dataset used for the experiment (The Corpus⁴)

The dataset used for this research contained about 1500 Amharic news articles written in Ge'ez font. 480 of them were used for the experiment which a manual summary could be developed due to the available resources. The same dataset was used by Saba (2001) and Theodros (2003) for retrieval model testing and Samuel (2004) for retrieval model testing using artificial neural network approach. The dataset was taken from Walta information center⁵ news archive. Even though the news articles in the available dataset did not have extracts that were used for the experimentation, in light of the amount of data available from this dataset and its accessibility it was decided that it would be suitable for the experimentation.

To make the dataset suitable for the experimentation (training and testing on the dataset) there was a need of human extracts for the articles so that a manual summary was made for 480 of the articles with a 30% sentence compression. Therefore, the 480 document / extract (summary) pairs were used for training and testing of models developed.

⁴In this report the words corpus and dataset are used interchangeably

⁵Walta Information Center is a private news and information service establishment based in Addis Ababa, Ethiopia

1.4.3 Dataset Preprocessing

The articles and their manual extract (summaries) were stored in a database. The text was lexically analyzed, stop words (non-content bearing words) removed, and the words were stemmed and indexed. A stop word list was required to avoid unnecessary words from being used in the analysis of document sentences.

The preprocessing stage split the text into word groups. This word level feature value works for the three features used in this study (namely the thematic words, cue words, and title words) except for the location feature which requires knowledge about the position of each sentence in the documents. Each sentence is then represented as a vector with the four features.

A stemmer was used to bring words with different inflected forms into one in order to increase the efficiency of the experimentation and the prototype developed. For instance, it is not the same to look for those sentences containing a title word in its exact form and to look for those that have the word in an inflected form but with the same topic they are discussing when collecting feature value for title words and thematic words in sentences.

An indexer was also used to collect and analyze terms that are frequently occurred in documents for the identification of thematic words in documents. Zelalem (2001) has developed an indexer, and stemmer for the Amharic text but these tools were not readily accessible as they are embedded with the system he has developed. Due to this, the stemmer and indexer for this study were implemented from scratch by adapting principles from him.

1.4.4 Design Tools

To make accessing of the dataset easy, a database was created using Microsoft Access 2000 so that each document /extract pair was identified by unique identification number. Microsoft Access was used for building the database because it is easy to make use of it.

For training of the classifier WEKA (A machine learning tool) was used. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes and presents a collection of algorithms for solving real-world data mining problems (WWW4). WEKA is open source software issued under the General Public License (Kirkby and Frank, 2004) and is freely available at <http://www.cs.waikato.ac.nz/ml/weka/>.

Visual Basic 6.0 programming language was used for collection of feature values on the sentences which was used as an input for training the classifier and also used for the development of the prototype summarizer which was developed based on the values estimated from the model developed. Though there were different scripting languages suitable for dealing with such problems, Visual Basic 6.0 was used due to its simplicity and familiarity to the researcher.

1.4.5 Test Plan

Before the evaluation of models on different combination of feature sets, the dataset (the document/extract pairs) was divided into two. 100 of them for initial experiment and the remaining 380 for the main experiment. The initial experiment was used for adjustment of some parameters in the classifier (machine learning tool) and to ensure the values collected were correctly allocated. The final experiment was used for the development of models on different features combinations and estimation of feature values used for the development of the prototype summarizer.

1.4.6 Experimentation

Given the sentences in the dataset with their classes ("summary " or "non- summary" identified manually for each document), the naïve Bayes probabilistic classifier (in the WEKA⁶ package) was trained to classify these sentence with the relevant values collected for features (attributes) whether they should or should not appear in the summary.

There are a number of reasons for choosing the naïve Bayes method for this experiment. It is appropriate to represent sentences in terms of features (since in machine learning every instance is represented as a vector, in this case an instance is a sentence, the term feature in this case is an attribute in data mining terminology) that are then selectively combined together to yield a significant value (a value which is used to "weight" sentences) which is used to extract sentences. Also studies have shown it is a reliable method in comparison with other machine learning methods like Support Vector Mechanics (SVM) and Artificial Neural Networks (ANN) for classification task (Mitchell, 1997).

It is also based on Bayes probability concepts, which is generally easy to understand. This is in contrast to other classifier techniques, such as Support Vector Mechanics (SVM) and Artificial Neural Networks (ANN). In addition, the assumption that the classifier makes on the features, as conditionally independent, was important because there were no obvious dependencies between the features under consideration.

For training of classifiers, usually two methods are used:

- I. To divide the samples into two parts. To ensure as large an area of analysis as possible from the experiment, the samples will be divided normally with a greater

⁶ WEKA (which stands for Waikato Environment for Knowledge Analysis) is a collection of machine learning algorithms for solving data mining problems, which is developed in the University of Waikato, New Zealand.

bulk of the samples being in the training section and the remainder, being used for testing.

- II. To divide the samples into a fixed number of folds, or partitions then to use each partition in a rotation method once for testing and the remaining for training. This method is suitable when there is a smaller amount of data is available (the method also called the cross validation method) (Witten and Frank, 2000).

In this experiment, the cross validation method was used for evaluation of the classifier, since it is suitable for the limited amount of data used (Witten and Frank, 2000). In particular, a 10-fold cross validation approach was used. In this case the data was split into ten approximately equal partitions, and each in turn was used for testing while the remainder was used for training. This was, to use nine-tenths of the sample for training and one-tenth for the testing, and repeat the procedure ten times so that in the end, every instance was used exactly once for testing.

The features used in this study were based on Edmundson's (1969) research. In particular, the optimum three feature title word, cue word and location features were used for the purpose of extracting sentences for a summary. In addition, the word frequency feature was used so that the presence or absence of these four features was identified for each sentence in the training corpus and an optimal combination of these features to select sentence extracts was identified.

1.4.7 Evaluation of models developed

After training of the models developed for different combination of the feature sets, their performance of classifying summary sentences was measured based on classification success rate, precision and recall by comparing sentence match predicted by the classifier used and the manual summaries used for testing performance of the classifier.

1.4.8 Prototype Development

A prototype summarizer was developed based on the combination of all the feature sets used. The summarizer extracts sentences based on the combination of all the features and their "weights" that was estimated in the training phase of the experimentation.

1.5 SIGNIFICANCE OF THE STUDY

This research is important in many ways other than for the fulfillment of academic exercise. It is believed to produce results that can indicate the possible application of machine learning technique (in this case the naïve Bayes method) for automatic creation of news summaries for Amharic language. The result of the research can also be used as an input to the development of full-fledged news summarizer for online Amharic news service providers that will help their users in accessing the news.

1.6 SCOPE AND LIMITATION OF THE STUDY

This study focuses on exploring the application of machine learning (naïve Bayes) for automatic text summarization of Amharic news text. It tries to present how far applicable the trained outputs of machine learning are towards generating summaries. In doing so, it makes use of the title words, cue words, thematic words, and location of sentences to be trained by the system.

This research does not aim to present machine learning approach as a principal application to develop summarizers nor is it regarded as the only solution. At the same time, the findings of this research are not totally ends or solutions by themselves. They represent indicative results that can be used as inputs for summarization projects. Therefore the attempt is to study the possible application of Machine Learning (naïve Bayes) in automatic text summarization of Amharic news articles.

The study has been constrained to some extent due to the unavailability of experts who work on manual summarization which was used for the experimentation. It is my belief that the availability of the experts would have enabled better development of the corpus.

1.7 ORGANIZATION OF THE THESIS

This report is organized into five chapters. The first chapter consists of the background to the problem and justification for conducting the research. The objectives behind this study are also presented. The methods used in conducting the research are presented and the scope and limitation of the study is also defined.

The second chapter of this report presents the basic concepts behind automatic text summarization. The process of automatic text summarization and types of text summarization are also discussed in this chapter. This chapter also comprises literature review on some of the related automatic summarization studies.

Chapter three presents machine learning concepts and review on automatic text summarization using a machine learning approach. A naïve Bayes machine learning algorithm is also reviewed.

In chapter four the experiments which are the preprocessing and the output of the research are discussed and reported. The development of prototype is also discussed in this chapter. In relation to the preprocessing, Amharic writing problems are also discussed. Chapter five contains conclusions drawn as a result of the experiment and recommendations made for further research.

CHAPTER TWO

PREVIEW

This chapter briefly discusses the basic concept on automatic text summarization, types and processes to automatic text summarization and review of related research works on automatic text summarization.

AUTOMATIC TEXT SUMMARIZATION

2.1 BASIC CONCEPT

Text summaries are used to reduce the complexity and length of texts, while retaining the most important information (Luhn, 1958; Kupiec et al., 1993). Its purpose is to give the reader an accurate and complete idea of the contents of the source (Farzindar and Lapalme, 2004). In this way "It addresses the problem of selecting the most important portions of the text and the problem of generating coherent summaries" (Brunn et al., 2001).

Determining what information in the source to be included in a summary is a key problem (Mani and Bloedorn, 1998). The determination of important information from the source needs several factors. Mani and Bloedorn (1998) mentioned the following as the determining factors for generating a summary: nature and genre of the source text, the desired compression (summary length as a percentage of source length), and the user's information needs.

As the manual and automatic methods to summarization uses different tools to generate summaries the challenges to the creation of summaries may be different. Specifically, two challenges are frequently mentioned for automatic summarization (Chuang and Yang, 2000). The first is the extent to which a chosen text is "understood", and the second challenge relates to the features or characters used to generate a summary.

Understanding a document without digesting the whole text is difficult (Chuang and Yang, 2000). In automatic text summarization, a computer program produces automatically summary from an original source text using different techniques (Lie, D.H. et al., 2001). Lie, D.H. et al. (2001) mentioned two techniques to summarization: text extraction and text abstraction. Text abstraction is the process of parsing the original text in a linguistic way, interpreting the text and finding new concepts to describe the text and then generating a new shorter text with the same information content. In text extraction, pieces of the original text is extracted on a statistical basis or based on surface domain-independent linguistic analysis and put together into a new shorter text with the same information content (Karakaya et al., n.d). In text extraction the method is basically to give scores to each sentence depending on the importance of each sentence (Visser et al., n.d). The scores can be based on high-frequent words, bold or numerical text, proper nouns, citations, position in text etc.

2.2 THE AUTOMATIC TEXT SUMMARIZATION PROCESS

According to Moens (2000) there are three different steps in performing text Summarization:

- I. Analysis of the source text to identify the essential content to build an internal representation,
- I. Transformation of the internal representation into a summary representation,
- II. Synthesis of output text which involves drafting and generation of the summary text based upon the summary representation. This last step is especially concerned with the organization of the content and essential for abstract type summary (Moens, 2000). Since the purpose of this study is to develop an extract type summary using a sentence as a basic component, the focus is on the methods related to the first two steps which are used in extract type summary production.

Source analysis is a process of determining the features that will drive the summarization process. Depending on the summarization technique, the analysis may take a variety of forms ranging from statistical methods that only search for specific key words to complex techniques that employ natural language understanding (Moens, 2000). In statistical approach the major concern is the identification of important topic terms and the extraction of contextual sentences that contain them. The method severely reduces the domain and linguistic need for analysis of the source text and it is more independent from the subject domain and text genre, and can be broadly applied (Moens, 2000). The idea of using this technique is shown in Luhn (1958). The other approach for source analysis needs the complete understanding of the source text, whereby each sentence is processed into its propositions representing the meaning of the sentence (Moens, 2000).

The transformation step pruned and condensed the representation of a content resulted in the source analysis stage in order to form a summary representation (Moens, 2000). This stage requires additional knowledge about the task and audience of the summary to guide the selection of the information and about the subject domain to conduct an accurate generalization of the information (Moens, 2000). Finally, the summary is generated using the content chosen in the transformation phase. The generation phase also provides some form of processing that improve the readability of the final summary output (Moens, 2000).

2.5 REVIEWS ON RELATED AUTOMATIC TEXT SUMMARIZATION STUDIES

Since the introduction of the study of automatic text summarization by Luhn (1958), many approaches have been proposed to resolve the problem in the field. Some of the related works of automatic text summarization are reviewed. In the next section, the works of Luhn (1958) and Edmundson (1969) which I base my study to select the features, and the works of Kupiec et al. (1995), Marcu (1997), Strzalkowski, et al. (1999), and Doran et al. (2004) which use a machine learning technique for summarization are reviewed.

2.5.1 The Automatic Creation of Literature Abstracts

Luhn (1958) proposed a single statistical approach, term frequency, for automatic text summarization of scientific documents. He was aimed at the idea of counting the frequency of words based on the intuition that the most frequent words represent the most important concept of the text, disregarding the very high frequency common words. He initially assigned a value to each word and this value, within the context of a sentence, created a significance value for that sentence. He further proposed that the relative position of frequent terms within a sentence also provides a useful measurement for determining the significance of sentences to be included in a summary. The significance factor of a sentence will therefore be based on a combination of these two factors. Words that occurred most frequently and least frequently were considered not significant. Also lists of common words were excluded from the list before the calculations started. The sentences with the highest significance value made up the summary.

His conclusions reported that since he aimed his summarization technique at papers within the science and technology fields, he believed that his "auto-abstracts" were satisfactory indicative abstracts for this subject area even though the results were

neither particularly good condensations nor very coherent texts (Hutchins, 1987). He also felt that they were a reliable and consistent form of summary.

2.5.2 New methods in automatic extraction

Edmundson (1969) proposed cue phrases, location and title features in the creation of automatic summary. He used a corpus of 200 scientific papers all from the area of chemistry. In total he ran 17 sets of experiments, with different combinations of one or more or all of keywords, cue words, location and title features, to find the optimum performing combination (i.e. the combination which would produce a summary with the most overlap with a human-generated summary). It turned out that cue-title-location produced the "best" summary, and that the keyword feature was not significantly important.

The cue words were based on a subset of the whole corpus, and consisted of single words only (not phrases). The title words were derived from the title, sub-title and headings. Keywords were any words within the document and were put in ascending order of frequency; there was a cut-off point for frequency. All non-cue words above the cut off point were classed as keywords. The location involved the "hand-weighting" of sentences; the first and last paragraphs were positive, headings such as Introduction and Conclusions were positive, and first and last sentences in the paragraph were positive.

To evaluate the summaries, Edmundson studied two different methods. The first was to get human evaluators to subjectively evaluate the similarity of the auto-generated summaries against those created manually. The second involved statistical error analysis, based on giving each sentence a value for the following properties: worthy to be in an extract of unrestricted length, in the target extract, in the automatic extract, and the negations of these properties. Edmundson's final equation overlaps with those criteria used by professional abstractors and is a faster form of evaluation as keyword analysis is a very time consuming feature.

2.5.3 A trainable document summarizer

This system applies a machine learning technique into automatic text summarization. The system uses the naïve Bayes as a trainable algorithm. To do this, Kupiec et al. (1995) took a corpus of 188 texts, consisting of document–summary pairs sampled from 21 publications in the scientific and technical domain, which had been written by a professional abstractor. They proposed summarization as a classification problem, where the goal is to discover a classification function that accepts a sentence as input and then tested each sentence for certain features which produced a total probability that a sentence belongs in a summary given a vector of features. Then the system chooses the sentences with the highest probability as summary sentences. The features they measured for each sentence were:

- Sentence length cut-off - indicates whether or not the sentence length is below a specified threshold. Sentences longer than a certain threshold have a value of true, and the others are false.
- Fixed-phrase – sentences containing a number of key phrases tend to be included in summaries. Kupiec et al. use 26 indicator phrases. Sentences containing those phrases, or ones following section heads with key words, have a value of true. All other sentences have a value of false.
- Paragraph – sentences are categorized as being the initial sentence, the middle sentence, or the final sentence in a paragraph.
- Thematic word feature – the most frequent content words were defined as thematic words. Sentences are ranked based on the frequency of theme words.
- Uppercase words –proper names and acronyms considered to be important terms. Similar to thematic word, sentences are ranked according to the frequency of upper case words that appear in the document.

For each sentence s , the probability (P) of a sentence appearing in the summary S , given the features $\{F_j\} j=1... K$ using a Bayesian Classifier is as follows:

$$P(s \in S \mid F_1, F_2 \dots F_k) = \frac{P(F_1, F_2, \dots, F_k \mid s \in S) * P(s \in S)}{P(F_1, F_2 \dots F_k)}$$

Where s – Sentence, S – Summary, F – Feature, P – Probability and k – the maximum number of features.

The evaluation was analyzed by considering the overlap between the auto-generated summaries' content and the professional abstractors' content. Even though they were trying to create extracts (entirely made of text spans extracted from the original document) the evaluation measurements were carried out with abstracts (at least some materials of which do not exist in the original document). The best results (43% correctness in correspondence with manually extracted sentences by professional abstractors) are obtained by a combination of location, cue phrases, and sentences lengths features.

2.5.4 The Rhetorical parsing, Summarization, and Generation of natural language texts.

Marcu's (1997) research attempts to exploit the structural aspect of text, which is different from the aforementioned non-structured features such as term frequency, title words, and sentence location to extract sentences for a summary. It is based on the assumption that cue-phrases and discourse markers are direct hints of a text underlying discourse structure. He used a corpus to identify discourse markers. He created a tree known as Rhetorical-Structure Trees (R-S Trees) for each segment in a document. After the tree is created segments are laid out in a manner in which more important segments occupy the upper level of the tree nodes, where as less important segments reside deeper in the tree. The segments are scored, with the top of the tree being given the highest score. The summary is constructed from sentences which contain the most important segments. The evaluation of the summaries was carried out by comparing the

auto-generated summaries with human-generated ones (using the same algorithm) and Microsoft Office97's in-built summarizer. The auto-generated summarization program out-performed Microsoft's summaries in terms of recall and precision of the units.

2.5.5 A robust practical text summarizer

Strzalkowski, et al. (1999) have proposed a system for summarizing news which combined features (such as key word, cue phrases and title words) with a Discourse Marco Structure (DMS). The system worked by segmenting paragraphs. The topic of the paragraphs are selected through several criteria, including the presence of words and phrases frequent in the text, title words, words occurring in the initial sentence of paragraphs, paragraphs close to the beginning of the document, occurrence of names of people, places, and organizations, certain phrases that indicate the main point of the text, and the presence of connectives of discourse. This system is particularly developed to news articles but the authors agreed that it is possible to handle other types of document as long as the documents had a well defined paragraph structure.

2.5.6 News Story Gisting at University College Dublin

This study presents a machine learning technique into generation of a very short news story summary which combines linguistic, statistical, and positional information. For this study Doran et al. (2004) took DUC (Document Understanding Conference) 2003 collection as a training data. The system uses a decision tree classifier to establish which phrases in a text should be included in the summary. To do this, the classifier attempts to predict the summarization potential of a word based on a set of features. For each occurrence of a term in a document they calculate the following features:

- The term frequency words in the document;
- The inverse document frequency terms in the corpus;

- The relative position of a word with respect to the start of the document in terms of word distance;
- A lexical cohesion score that measures the lexical cohesive strength of the relationships between a word and the document in which it occurs. Their hypothesis is that if a word has a high cohesion score then it is a useful summary word because it is strongly associated with other important terms in the document;
- A noun, a verb, or an adjective, or a word occurs in a noun or proper noun Phrase considered to be an important terms and are assigned a binary score.

The evaluation was analyzed by considering the overlap between the summary sentences predicted by the classifier and the human summaries. The DUC 2003 evaluation provides four human summaries for each document (Doran et al., 2004). The appropriateness of a word as a summary term is determined based on existence of the word in the four human generated summaries. Therefore, if the classifier predicts that a word will occur in all four of the human-generated summaries, then it is considered to be a more appropriate summary word than a word predicted to occur in only three of the model summaries. Evaluation of the results of the classifier on 90%/ 10% training/testing data split showed a precision of 63% and recall of 20%. Recall, in this context, is the total number of true positives returned by the classifier divided by the total number of true positives and false negatives, while precision is the total number of true positives returned by the classifier divided by the total number of true positives and false positives.

CHAPTER THREE

PREVIEW

This chapter briefly discusses basic concept of machine learning, the need for machine learning, the benefit of machine learning and machine learning based text summarization and related concepts such as, feature sets and representations, sentence selection and training and testing of sentence classifier are discussed. In addition, for the purpose of this research the naïve Bayes machine learning algorithm is made reviewed.

MACHINE LEARNING APPROACH TO AUTOMATIC TEXT SUMMARIZATION

3.1 MACHINE LEARNING

Machine learning is the ability of a machine to recognize patterns that have occurred repeatedly and improve its performance based on past experience (Techdictionary, 2005). Machine learning develops computational methods that would implement various forms of learning, in particular mechanisms capable of inducing knowledge from examples or data (Kubat et al., 1996). Machine Learning is the field of scientific study that concentrates on induction algorithms and on other algorithms that can be said to "learn." (Kohavi & Provost, 1998). Russel & Norvig (1995) defined machine learning as a subfield of artificial intelligence concerned with programs that learn from experience. Mitchell (1997) defined machine learning more specifically as "A computer program is said to learn from Experience E with respect to some class of Tasks T and Performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

As such, the field of machine learning is more particularly concerned with the question of how to construct computer programs that automatically improve with experience. Machine learning theory attempts to answer questions such as "How does learning performance vary with the number of training examples presented?"; "Which learning algorithms are most appropriate for various types of learning tasks?" and "How can the learner automatically alter its representation to improve its ability to represent and learn the target function?" (Mitchell, 1997).

One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner. In particular, many of the machine learning algorithms searches a hypothesis space defined by some underlying representation (e.g., linear functions, logical descriptions, decision trees, artificial neural networks).

Machine learning usually distinguishes supervised, unsupervised, and reinforcement learning. Supervised learning involves learning a function from examples of inputs and outputs. Unsupervised learning involves learning patterns in the input when no specific output values are supplied. Reinforcement learning is learning through trial and error via the reception of a numerical reward. With this learning method the learner attempts to map state and action combinations to their utility, with the aim of being able to maximize future reward. Reward is usually received after a number of actions have been taken by the learner (Russel and Norvig, 2003).

In this study a supervised method which involves learning from the patterns of the input and output examples is used. In the remainder of this section concepts related to the learning method is presented.

Russel and Norvig (1995) define induction as learning a function from examples of its inputs and outputs. They also put it formally as:

"an example is a pair $(x, f(x))$, where x is the input and $f(x)$ is the output of the function applied to x . the task of pure inductive inference (or induction) is this: Given a collection of examples of f , return a function h that approximates f ."

The function h is called a hypothesis. "The Inductive learning hypothesis is any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples"(Mitchell, 1997).

Although the learning task is to determine a hypothesis h identical to the target concept c over the entire set of instances X , the only information available about c is its value over the training examples. Therefore, inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data. The fundamental assumption of inductive learning is that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data. Inductive learning methods require a certain number of training examples to achieve a given level of generalization accuracy. Decision trees, k-nearest neighbors, neural networks and Bayesian learning algorithms are examples of inductive learning algorithms that require large data (Mitchell, 1997).

3.2 MACHINE LEARNING BASED TEXT SUMMARIZATION

The idea of using machine learning techniques for automatic text summarization is not new. For example, Kupiec et al. (1995) as shown in section 2.4.3 proposed a machine learning technique for automatic text summarization. Teufel et al. (1999) developed a technique for summarizing texts by selecting a subset of sentences that preserves some information about the general rhetorical structure of the text. To train the summarizer they use seven features. Methodologically, they build on the experiments done by Kupiec et al. (1995).

They use 201 articles with author-written abstracts collected from the computation and language archive. The results for the classification (varying between 16.2% – 56.3% for the individual features) also show that the thematic word feature produces lowest individual score, while the Indicator rhetoric feature produces highest individual score. The best result achieved when combining every feature except Indicator rhetoric feature, the cumulative score reaches 66%. Chuang and Yang (2000) also used supervised learning algorithms to generate a summary by extracting sentence segments. In their approach first sentences were broken into segments by special cue markers. Then each segment was represented by a set of predefined features. The total number of features they collected were 23, and they divide these features into three groups for analysis. One of the groups contains non-structural features. These are features associated with the location of the segment, number of significant (bonus) words (out of the pre-defined set of 46 words), and the average term frequency of the words in the segment. The remaining two groups contain structural features of the text and they classified as distinct rhetorical relations and collective descriptions of the rhetorical relations. The decision tree algorithm (C 4.5), the naïve Bayesian classifier, and the inter pattern distance-based constructive neural network learning algorithms were used in their experiment on U.S patent data. The performance of the summarizer was evaluated by a 9-fold cross validation and they found results that are comparable with other approaches (including Microsoft word summarizer) in terms of precision and recall.

3.4 FEATURE SETS AND REPRESENTATIONS

A large variety of features are found in the text summarization literatures which are typically used in sentence extraction approaches to automatic summarization. These features can be classified as non-structured features (example, title words, location, cue words, etc.) and structured features, that attempt to exploit the structural aspect of text (example, rhetorical relations) (Chuang and Yang, 2000). The following are the features examined in some of the related automatic summarization literatures.

Key words feature: words that occur frequently (after stop wording) are considered to be thematic (key) words (Luhn, 1958). This feature indicates whether or not the sentence captures the main concepts of the document (Neto et al., 2000). A writer normally repeats certain words as he or she advances or varies the arguments or as he or she elaborates on an aspect of the subject. This means of emphasis is taken as indicator of significance. A content term that occurs frequently in a text is more important in the text than an infrequent term. The frequency of occurrence of a content word is used to indicate term importance for content representation (Luhn, 1958). This feature assigns a value to a sentence based on the frequency of each uncommon word counted in a sentence. For example, Edmundson (1969) uses this feature to assign a score to each word equal to its frequency. The score of the sentences was calculated by summing the scores of its words. Luhn (1958) also used this feature, which did not use the exact frequency of each word, to assign a score for sentences. But he also considers the relative position of significant words in to account.

TF-IDF (term frequency-inverse document frequency) feature: In information retrieval TF-IDF is one of the very important measures for content representation (Salton et al. as in Spark Jones & Peter Willet, 1997). The method assumes that the term frequency method make no distinction between terms that occur in every document of a collection and those that occur in only a few documents. In this regard the method favors terms with high frequency in particular documents but a low frequency over all in the collection Salton et al. (1983). This feature is used in text summarization to capture the extent to

3.4 FEATURE SETS AND REPRESENTATIONS

A large variety of features are found in the text summarization literatures which are typically used in sentence extraction approaches to automatic summarization. These features can be classified as non-structured features (example, title words, location, cue words, etc.) and structured features, that attempt to exploit the structural aspect of text (example, rhetorical relations) (Chuang and Yang, 2000). The following are the features examined in some of the related automatic summarization literatures.

Key words feature: words that occur frequently (after stop wording) are considered to be thematic (key) words (Luhn, 1958). This feature indicates whether or not the sentence captures the main concepts of the document (Neto et al., 2000). A writer normally repeats certain words as he or she advances or varies the arguments or as he or she elaborates on an aspect of the subject. This means of emphasis is taken as indicator of significance. A content term that occurs frequently in a text is more important in the text than an infrequent term. The frequency of occurrence of a content word is used to indicate term importance for content representation (Luhn, 1958). This feature assigns a value to a sentence based on the frequency of each uncommon word counted in a sentence. For example, Edmundson (1969) uses this feature to assign a score to each word equal to its frequency. The score of the sentences was calculated by summing the scores of its words. Luhn (1958) also used this feature, which did not use the exact frequency of each word, to assign a score for sentences. But he also considers the relative position of significant words in to account.

TF-IDF (term frequency-inverse document frequency) feature: In information retrieval TF-IDF is one of the very important measures for content representation (Salton et al. as in Spark Jones & Peter Willet, 1997). The method assumes that the term frequency method make no distinction between terms that occur in every document of a collection and those that occur in only a few documents. In this regard the method favors terms with high frequency in particular documents but a low frequency over all in the collection Salton et al. (1983). This feature is used in text summarization to capture the extent to

which a sentence contains terms which are significant in a document (Neto et al., 2000). In this case we have a single document, and we have to select a set of relevant sentences to be included in the extractive summary out of all sentences in the document.

Hence the notation of collection of documents in information retrieval can be replaced by the notation of a single document in text summarization. Therefore, the notation of document which is an element of a collection of documents in information retrieval corresponds to the notation of sentence which is an element of a document in summarization. Neto et al. (2000) denoted this measure as term frequency times inverse sentence frequency.

Sentence length: This feature describes the length of sentences as short or long depending on whether they are less than or more than certain words in length. Kupiec, et al. (1995) employed this feature to exclude sentences that are too short. In their method section headings were generally counted as short sentences. To identify the length of sentences they establish a threshold so that sentences longer than the threshold have a value of true, and the others are false.

Title feature: This feature determines the presence of title word in a sentence so that the sentence gets a score based on occurrence of the title words. The central idea behind this feature is that authors conceive the title as circumscribing the subject matter of the document (Visser and Wieling, 2004).

Location feature: This feature can involve several items, such as the location of a sentence in a document as a whole, its location in a section, in a paragraph, etc, and has presented good results in several researches (for example, Edmundson, 1969, and Kupiec et al. 1995).

Upper case word feature: proper names and acronyms tend to be important to give significance value for a sentence (Kupiec, 1995). As it is discussed in their study, this

feature was not useful for scientific journal articles. However, this feature was included for the robustness that they considered one of the several important issues in addition to the training of features and performance evaluation.

Cue features: Cue words and cue phrases have been used as a criterion to create summaries. Each of these words or phrases is often used to signal important content to the reader, thus they can be used by automatic summarization to select sentences (Salton et al., 1983). Cue words and phrases can be classified into three different categories, each representing the significance of the word or phrase to the summarization. Bonus words are those words indicating the positive relevance of a sentence to the summary; stigma words, words that indicate a negative relevance for a summary, can be used to filter out unimportant sentences; and, null words, that are irrelevant for selecting summary content (Edmundson, 1969).

3.5 SENTENCE SELECTIONS AS A FUNCTION OF CLASSIFICATION

"Sentences are used to assert, query, command, or bring about some partial description of the world and sentences organized in such a way as to minimize the communicative effort of the user of the text" (Moens, 2000). Several methods are used to assign a score to sentences and then to select the highest scoring subset of sentences that summarize a text document (Edmundson, 1969; Kupiec, 1995). In machine learning approach to automatic text summarization, which needs a collection of documents and their corresponding manually extracted summaries, the sentences of each document are modeled as vectors of features extracted from the text (Chuang and Yang, 2000).

The summarization process can be taken as a classification problem, if a collection of documents with their corresponding manual summary is provided (Shen et al., 2004; Mani et al., 1998; Kupiec et al., 1995; Neto et al., 2002). Training a text classifier involves a classification procedure from a set of example texts for which the true classes are known (Moens, 2000). She puts the general approach to a classification task as follows. An expert or supervisor assigns subject or classification codes to the example texts of the training set. It is assumed that the assignment is correct. Then, a classifier is constructed based on the training set. The aim is to detect general, but high accuracy classification patterns or rules in the training set, which are highly predictive to correctly classify new, previously unseen texts. The set of new texts is called the test set.

3.6 TRAINING VERSUS TEST SETS

The machine learning approach relies on the availability of an initial corpus of documents (Kupiec et al., 1995). If sufficient amount of data is available, prior to the construction of classifier the initial corpus is split in two usually with a greater part of it in the training section, to ensure as large an area of analysis as possible, with the remainder of it, the test data, being used for the creation of summaries and evaluation (Witten & Frank, 2000). Most researchers use 20% (McCallum & Nigam, 1998), 30% (Koller & Sahami, 1997) or 33 % (Joachims, n.d) of data for test set and the remaining for training set. If the data available is not sufficient it is a better way to use cross validation method (for example, Kupiec et al., 1995). In this method the data is split in to partitions of approximately equal size so that each partition in turn is used for testing while the remainder is used for training. In this way every instance will be used for training and testing (Witten & Frank, 2000).

The training set is inductively built by observing the characteristics of the initial corpus. Then the test set is used for testing the effectiveness of the classifier. Each document from the test set is fed to the classifier, and the classifier decisions are compared with the expert decisions. The documents in test set cannot participate in any way in the inductive construction of the classifiers; otherwise, the experimental results obtained would likely be unrealistic, and the evaluation is considered not scientific (Sebastiani, 2002).

3.7 TYPES OF CLASSIFIERS

Mitchell (1997) describes four main types of machine learning: Concept learning; decision tree learning; artificial neural networks and Bayesian classification. Concept learning is based on the idea that we can define objects or events over a larger set. Decision Tree Learning attempts to come to the correct conclusion by traversing a tree of options. Essentially, the algorithm classifies an instance by applying a series of hierarchical tests. While the above methods are similar in their approach in that they are based on adapting heuristics to fit a given set of data, classifiers based on artificial simulations of the animal brain (neural networks) work in a very different way. Here a system is composed of neurons, each of which allows a number of weighted inputs and one output (Although many neurons may be connected to that one output). Weights on the inputs are adapted based on examples, that are usually given by a trainer but which can also be learned from nature. Bayesian learning allows examining classification problems in terms of the probability that a particular instance satisfies a hypothesis (Mitchell, 1997).

The focus of this study is on one of the Bayesian probabilistic classifier (i.e. naïve Bayes). The next sections discuss the general Bayes probabilistic theory which underlines the naïve Bayes classifier and the classification principle of the naïve Bayes classifier.

3.7.1 Naive Bayes

The naïve Bayes classifier is one of the highly practical Bayesian learning methods, which are based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data (Mitchell, 1997). The naïve Bayes algorithm employs a simplified version of Bayes formula to decide which class a novel instance belongs to. The posterior probability of each class is calculated, given the feature values present in the instance; the instance is assigned the class with the

highest probability. The basis for all Bayesian learning algorithms is the Bayesian theorem. The following discussion on Bayes theorem is based on Mitchell (1997).

Bayes theorem is a simple mathematical formula used for calculating conditional probabilities. Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability which may reflect any background knowledge about the chance that h is a correct hypothesis before we have observed the training data, $P(h)$; the probability of observing various data given the hypothesis, $P(D|h)$; and the observed data itself given no knowledge about which hypothesis holds, $P(D)$. If we have no prior knowledge, then we might simply assign the same prior probability to each candidate hypothesis, $P(h)$.

The formula for Bayes Theorem is given:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (3.1)$$

$P(h|D)$ increases with $P(h)$ and with $P(D|h)$. $P(h|D)$ decreases as $P(D)$ increases, because the more probable it is that D will be observed independent of h , the less evidence D provides in support of h .

Machine learning problems are interested in the probability $p(h|D)$ that h holds given the observed training data D , which is called the posterior probability of h , because it reflects the confidence that h holds after seeing the training data D , in contrast to the prior probability $p(h)$, which is independent of D .

In many learning scenarios, the learner considers some set of candidate hypotheses H and is interested in finding the most probable hypothesis $h \in H$ given the observed data D (or at least one of the maximum probable if there are several). Any such maximum probable hypothesis is called a maximum a posteriori (MAP) hypothesis. It is possible to determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis, i.e., h_{MAP} is a MAP hypothesis provided

$$\begin{aligned}
 h_{MAP} &\equiv \arg \max_{h \in H} P(h \setminus D) \\
 &\equiv \arg \max_{h \in H} \frac{P(D \setminus h)P(h)}{P(D)} \\
 &\equiv P(D \setminus h)P(h)
 \end{aligned} \tag{3.2}$$

The denominator, $P(D)$, is generally removed as it is constant independent of h and it thus has no bearing on which has the highest probability. In some cases like if we have no any background knowledge about the hypothesis sets; it is assumed that every hypothesis in H is equally probable a priori ($P(h_i)$ and h_j in H). Therefore, we need only to consider the term $P(D \setminus h)$ to find the most probable hypothesis. $P(D \setminus h)$ is often called the likelihood of the data D given h , and any hypothesis that maximizes $P(D \setminus h)$ is called a maximum likelihood (ML) hypothesis,

$$h_{ML} \equiv \arg \max_{h \in H} P(D \setminus h) \tag{3.3}$$

In machine learning problems the data D is considered as training examples of some target function H as the space of candidate target functions. Since Bayes theorem provides a principled way to calculate the posterior probability of each hypothesis given the training data, it can be used as the basis for a straightforward learning algorithm that calculates the probability for each possible hypothesis, and then outputs the most probable.

3.7.2 The Naïve Bayes classifier

The naïve Bayes classifier (NBC) is applicable to learning tasks where each instance is described by a conjunction of attribute values and a target function which takes a value from a finite set V . A set of training examples for the target function is provided, a new instance described by the attribute values $(a_1, a_2, \dots, a_n)$ is then presented, and the learner is asked to predict the target value or classification of new instance. The Bayesian approach to classify the new instance is to assign the most probable that is the Maximum A Posteriori (MAP) hypothesis, given the attribute values that describe the instance (Mitchell, 1997).

$$V_{\text{MAP}} = \arg \max_{v_j \in V} P(V_j | a_1, a_2, \dots, a_n) \quad (3.4)$$

Where, v_{MAP} is the most probable target value. Using Bayes theorem equation (3.4) can be written as follows:

$$\begin{aligned} V_{\text{MAP}} &= \arg \max_{v_j \in V} \frac{P(a_1, \dots, a_n | v_j) P(V_j)}{P(a_1, \dots, a_n)} \\ &= \arg \max_{v_j \in V} P(a_1, \dots, a_n | v_j) P(V_j) \end{aligned} \quad (3.5)$$

In this way for the training part of the program, the inputs will be the examples of the target functions, and the values of the attributes will be filled by these examples. Using training data the two terms in equation (3.5) must be calculated. It is very easy to estimate each $P(V_j)$ by counting the frequency of occurrence of each target value in the training data (i.e. $P(V_j) = s_j/S$ where s_j is the number of training samples of class V_j and S is the total number of training samples. However, estimating the different $P(a_1, \dots, a_n | v_j)$ terms in this way is not possible unless a huge set of training data is available. The problem is that the number of these terms is equal to the number of possible instances times the number of possible target values. In order to make the classifier much more practical and computationally efficient, the simplifying assumption

that the attribute values are conditionally independent given the target value is used which helps to see every instance in the instance space many times in order to obtain reliable estimate. In other words, the assumption is that given the target values of the instance, the probability of observing the conjunction $a_1, a_2, \dots, a_n$ is the product of the probabilities for the individual attributes. This implies that:

$$P(a_1, \dots, a_n | v_j) = \prod_i P(a_i | v_j) \quad (3.6)$$

Substituting equation (3.6) in to equation (3.5) results in the approach used by naïve Bayes classifier given by equation (3.7):

$$V_{NB} = \arg \max_{v_j \in V'} P(v_j) \prod_i P(a_i | v_j) \quad (3.7)$$

Where, V_{NB} denotes the target value output given by the naïve Bayes classifier.

It is possible to estimate $P(a_i | v_j)$ based on the frequencies over the training data. Therefore $P(v_j)$ and $P(a_i | v_j)$ estimates corresponds to the learned hypothesis. This hypothesis is then used to classify each new instance by applying the rule in equation (3.7).

Despite the fact that the independence assumption is often violated in practice, naïve Bayes classifier has shown itself competitor with other classifiers. It is shown to be very effective in many practical domains such as text categorization (Mitchell, 1997; Surafel, 2003).

The naïve Bayes classifier has several distinctive features which make it suitable for classification task. It has exhibited high accuracy and speed when applied to large databases. It provides a very simple yet surprisingly accurate technique and is an efficient and effective inductive learning algorithm for machine learning and data mining (Mitchell, 1997).

The computational efficiency of this classifier has made it usable in a number of research efforts; its learning time is short and linear in the number of training example making it suitable for real-time learning (Witten & Frank, 2000; Mitchell, 1997; Zheng,

1998; Zhang & et al., n.d; Keswani, 2002). From equation (3.7) it is clear that learning is simply done through counting the frequency of various data combinations with in the training examples.

3.8 EVALUATION METRIC

How to evaluate computer-produced summaries has been recognized as a problem in the field of automatic summarization. According to Mani and Maybury (1999) as in the Fukushima, T. and Okumura, M. (2003) there are two classification of evaluation: extrinsic and intrinsic. An extrinsic method of evaluation is where the quality of the summary is judged on how well it helps a person perform another task such as information retrieval. An intrinsic evaluation is where humans judge the quality of summarization directly on an analysis of the auto-generated summary. In this way intrinsic methods measure a system's quality; extrinsic methods measure a system's performance in a particular task.

The summarization systems reviewed in this study uses the evaluation systems. Most evaluations of these systems is based on intrinsic one (for example, Edmundson, 1969; Kupiec, Pederson, and Chen 1995; Marcu 1997; and Doran et al. 2004). Their approach is to create an "ideal" summary, either by professional abstractors or by merging summaries provided by multiple human subjects. The output of the systems is then compared with the "ideal" summary. Precision (the proportion of actual positive class members returned by the system among all predicted positive class members returned by the system) and recall (the proportion of predicted positive members among all actual positive class members in the data) were used to measure the quality of the summaries.

CHAPTER FOUR

PREVIEW

This chapter discusses about the corpus (dataset) used, preprocessing of the dataset in preparation for the training and testing of the classifier used, the results of the classifier, and the development of summarizer and the evaluation process are explained. The preprocessing and training and testing of the classifier are performed based on the concepts discussed in the previous chapters. In addition problems in Amharic writing system in relation to the preprocessing of the data are also discussed.

EXPERIMENTATION

4.1 THE CORPUS

The full corpus used for the purpose of this experiment consists of about 1500 articles written in different Amharic fonts. The articles in the corpus were collected for four months (from November 2001 to February 2002). The dataset was taken from Walta Information Center. The dataset was organized on the date the articles were published.

From these large amounts of articles, 480 of them were used for the experiment which development of a manual summary was possible in relation to the available resources. These articles were selected randomly from the whole dataset. Three 4th year Amharic language students were participating in the extraction of manual summaries. Initially, for each article three manual summaries were produced by each of them using a 30% sentence compression.

The reason for using a 30% sentence compression to the manual extracts used in this experiment was that using a smaller compression (for example, 10%) would be unlikely to produce effective results for some of the documents, as these documents contained smaller number of sentences 10% of it could not produce a single sentence. A higher compression rate has not been found suitable for the corpus, as the aim of summarization is to produce smaller number sentences that fulfill the intended purpose. There fore 30% sentence compression is found a compromise between these two factors.

There were variations of sentences selected by these people to summarize a single article. Then sentences common to some or all of them at the same compression rate (30% sentence compression) were used.

The 480 document/extract pairs consisted of 3236 instances (sentences) which may be smaller in comparison with the number of instances used in other data mining applications to extract knowledge from a given data. However, this is considered reasonably enough for the type of application under consideration (see Kupiec et al., 1995). Nevertheless, to compensate, the cross-validation method (discussed at sec. 3.6) is used which divides the dataset on a "rotation" method and provides a system to use all the data for training and testing.

After the manual extracts were made for the 480 documents, the document/extract pairs were stored in MS-Access database so that each pair is given a unique identification number that made it easy to access the articles as well as the extracts (the snapshot of the document/extract pairs is attached on appendix I).

4.2.2 Stop word and cue word list

Non-content bearing (stop words) is a list of terms to be excluded from consideration when, for example, weighting terms or sentences (Frakes, 1992). Excluding certain words is, in fact, an important part of any summarization research, because it avoids unnecessary bias towards words which brings little benefit to an analysis of a technique. (For example, the keywords frequencies count (Luhn1958)). Excluding stop words has been a part of most summarization techniques, Luhn (1958), Edmundson (1969), and Kupiec, et al. (1995). In this study the content (thematic) words and title words features presence or absence was analyzed for the sentences after stop words were removed from the documents. The non-content bearing words (stop words) were identified in the following way: The frequency of words in each document in the collection was counted automatically and words which were high frequent in most of the document were identified as a non content bearing (stop word). In this case there was no cutoff values considered but to the whole collection the top high frequent words were used. In this way a total of 318 words were identified in the corpus. Some of the identified stop words were: ናቸው, ብለዋል, ገልፀዋል, ሺ, አመት, ሚሊየን, ላይ, ሃላፊ.

The cue word list that was used also based on a subset of the whole corpus, and consisted of single words only. Most of the cue words identified (by the people who work on the manual summary) were in the stop word list since these words also used in most of the documents frequently. List of cue words used is attached on appendix II.

4.2.3 Word Stemming

A stem is the portion of a word which is left after the removal of its affixes (Baeza-Yates and Ribeeiro-Neto, 1999). Stemming reduce variants of the same root word into a common concept. As such, it helps to improve retrieval performance and also helps to reduce the size of indexing structure by reducing the number of distinct index terms (Baeza-Yates and Ribeeiro-Neto, 1999). Various attempts have been made to develop stemming algorithms for the Amharic language (Alemayehu and Willet, 2002); Zelalem (2001) also developed a simple affix and prefix stripping technique.

The following stemming example (taken from Alemayehu and Willet, 2002) is for all words related to the verb **ሠራ** " sera" " work" (**እንደሚሰራ**, **እንደሚሰሩ**, **እንደሚሠሩ**, **እንደሠሩ**, **ሠሩ**, **እየሰሩ**...) in to their root **ሰራ** "ser". As a statistical method is used in this study to collect feature values, stemming helps by reducing the variants of different forms of words across given document. For this purpose Zelalem's (2001) stemming principle was adapted. List of prefixes used in his work were **በ**, **ከ**, **ለ**, **ስለ**, **እንደ**, **የ** and list of suffixes used are attached on appendix III.

4.2.4 Amharic Fonts

Amharic alphabets do not have a representation in the ASCII (American Standard Code for Information Interchange) code table. As a result, font developers have tried to develop their own keyboard driver programs that make use of the existing English Keyboard (ASCII codes) for writing Amharic. The English keyboard buttons are used in various combinations to produce Amharic characters (Zelalem, 2001).

Different Amharic fonts have been produced over the years (e.g. Alpas, Brana I, Brana II, Power Ge'ez, Geez, Agafari, Alxethiopian, Visual Ge'ez, etc.) but they all use the existing symbol sets differently. The Amharic text used for this research was written in the Visual Ge'ez font. The symbol representation of the visual Ge'ez Amharic alphabets is attached in Appendix IV. In the Visual Ge'ez font, as in most Amharic fonts, the base

character is one byte and characters (those with diacritic markings) extend to more than one byte in their internal representation. E.g. **ñ**, which is a composition of **n** and **@**, which is internally represented as the two symbols 'b@'.

4.4 THE NAÏVE BAYES CLASSIFIER

The naïve Bayes classifier equation in sec. (3.7) is summarized as:

$$V_{NB} = \arg \max_{v_j \in I'} P(V_j) \prod_i P(a_i | v_j)$$

Where, V_{NB} denotes the target value output by the classifier. In this study it denotes the probability of sentences being in a summary or not in a summary. There fore for each sentence s , the classifier determines the probability that it will be included in a summary S or not in a summary S , given features $F_j, j=1...4$. There fore the above equation can be re-written as:

$$P(s \in S | f1, f2, f3, f4) = \prod_{j=1}^4 P(f_j | s \in S) p(s \in S)$$

And

$$P(s \notin S | f1, f2, f3, f4) = \prod_{j=1}^4 P(f_j | s \notin S) p(s \notin S) \quad (4.1)$$

$P(s \in S)$, stands for the compression ratio of the extraction task, $P(s \notin S)$ is the probability of a sentence not in a summary, $P(f_j | s \in S)$ is the conditional probability of each feature j given the sentence is in a summary, and $P(f_j | s \notin S)$ is the conditional probability of each feature j given the sentence is not in a summary, which are estimated from the training data. This means that the training program will be checking every sentence in a document for whether it contains title words, cue words, content words, and what position is in the document and whether that sentence included in the human generated extract. Therefore, equation (4.1) is the weights that are computed for the sentences which are used to extract them that are computed in the training program.

4.5 RESULT AND DISCUSSION OF THE EXPERIMENTS

To start the experimentation, the 480 document/extract pairs were divided into two, for initial experiment with a randomly selected 100 document/extract pairs and a main experiment with the remaining 380 pairs. The first one was used for some preliminary experiments such as for ensuring that the feature values were accurately allocated and for setting some system parameters. To see the results of the classifier on the initial data a number of experiments run for the different combination of the features. In this case, the values collected for the title words, cue words, and content words features didn't not show the relevance of the features to the classes ("summary" or "not summary"), since these features take an infinite number of values which is the ratio of the number of these features to the total number of words in a sentence.

This method was used to treat long and short sentences equally because longer sentences will otherwise have a higher probability to contain these features. That is, unless such a method is used, longer sentences will have a larger value. For the case of location feature, since it has a smaller number of values to be assigned to the sentences, their relative position in a document as 1, 2, and 3, it was possible to see which parts were related to the classes. To assign a smaller number of values for the three features, the feature values collected for each sentences were converted into 0's and 1's (Boolean values). Generally, values different from zero were converted in to 1 because of the difficulty to set some cutoff values for conversion (because the number of words in the sentences were larger in comparison with the number of features found in the sentences which resulted a very small value). Then the main experiment was performed based on the Boolean values collected. In both the preliminary and the final experiments, all the results were obtained using a 10-fold cross validation (recommended for smaller dataset).

Evaluation of the results of the classifier is measured using predictive accuracy, precision, and recall which are set as a standard measures (Van Rijsbergen, 1979). The classification accuracy for a test can be shown using confusion matrix (Witten & Frank,

2000). Confusion matrix contains rows and columns where the row is the actual class and the column is predicted output to the corresponding classes.

Actual Classes	Predicted Classes	
	Summary sentences	Non-summary sentences
Summary sentences	a	b
Non-Summary sentences	c	d

Table 1. Confusion Matrix

In the context of this experiment “a” and “b” denote the number of summary sentences that are used during the evaluation of the classifier (manually selected summary sentences). “a” is the number of correctly predicted summary sentences by the classifier and “b” is the number of summary sentences predicted as a non-summary sentences by the classifier. On the other hand “c” and “d” denote the number of non-summary sentences that are used during the evaluation of the classifier (judged as non-summary sentences by manual extractors). “c” is the number of non-summary sentences predicted as a summary sentences by the classifier and “d” is the number of non-summary sentences predicted correctly.

Accuracy concerns the percentage indicating how the classifier correctly classified the sentences and is given by:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$

$$= \frac{\text{Number of testing sentences correctly predicted}}{\text{Total number of testing sentences}}$$

$$\text{Recall} = \frac{a}{a+b}$$

$$= \frac{\text{Number of summary sentences classified correctly}}{\text{Number of known summary sentences, during testing}}$$

$$\text{Precision} = \frac{a}{a+c}$$

$$= \frac{\text{Number of summary sentences classified correctly}}{\text{Number of examples classified positive, during testing}}$$

WEKA provides the results of the classifier using confusion matrix. The evaluation of the final set of experiments in predicting the classes was conducted for different combination of the features.

The results of the analysis collected by the training program for the final experiment were as follows. When considering the individual features in predicting the classes they tended to predict the non-summary class except the location feature which results a precision of 75.00%, recall 74.90%, and classification accuracy 86.03% in predicting the summary sentences. The results of the individual features are attached on appendix IV.

Such results may be due to the skewed-class distribution in which only 28% of the instances were in the class summary. The problem of imbalanced data is familiar in machine learning and results in the classifier performing best for classifying majority class but failing in the minority class (Japkowicz and Stephen, 2002). And also the title, cue words, and thematic words feature found in 40-50% of the non- summary sentences which might result in a poor performance in predicting the class summary.

When looking at the different combination of the features, adding the location feature has been observed to improve the classifier performance in predicting the summary sentences. The results of some of the combined features is attached on appendix V.

Results on the different combination of feature sets which includes the location feature shows a better performance in all measures. Especially the combination of cue words, location, and title words; cue words, thematic words, and location; and thematic words, location, and title words shows better results.

The combination of all the features used also shows a precision of 75.60%, recall 72.30% and, a classification accuracy of 85.76% which is comparable with the above three combinations. Table below is the result of all features combination.

As it is noted, the results obtained shows the location feature performs better in comparison with the other individual features in predicting summary sentences. In a combined set of features also the inclusion of this feature gives a better result in prediction of summary sentences.

A model was developed using the combination of all the features even though, other combinations also give good results. But using all the features helped to consider sentences that have the different features to be part of the summary. To evaluate the performance of the model with documents that were not used in the training program, 20 documents were used. Manual summaries were made for these documents with a 30% sentence compression. Then the model was fed with the documents' sentences. The match between sentences predicted by the classifier and manually tagged sentences were compared. The table below shows the model prediction on the input documents' sentences and its match with the manual tagged sentences.

Document id	Model prediction		Match with the manual tagged sentences	
	Summary sentence	Non-summary sentence	Summary sentences	Non-summary sentences
1	2	5	2	5
2	1	6	1	5
3	2	7	2	6
4	1	7	1	6
5	2	6	2	5
6	2	4	1	3
7	4	10	2	8
8	2	6	2	6
9	2	6	1	5
10	1	6	1	6
11	1	5	0	3
12	3	8	2	7
13	2	5	2	5
14	3	6	2	5
15	3	7	2	6
16	4	9	2	7
17	2	6	1	5
18	3	6	1	4
19	3	8	2	7
20	3	7	2	6

Table 3. Model's prediction for unlabelled documents

Evaluation of the model based on these articles gives an average precision of 80%, average recall 83.3%, and an average classification accuracy of 81.17% on predicting summary sentences. Finally, a prototype summarizer was developed based on the feature weights estimated from the model developed.

4.6 DESIGN AND DEVELOPMENT OF THE PROTOTYPE SUMMARIZER

The prototype summarizer is developed based on the feature weights estimated from the model developed on the final set of data using all the feature sets. The model classifies sentences as being a "summary" and "not a summary" by comparing their probability of being in one of the two classes. Therefore, when a new document (which is not trained) is supplied to the model it resulted in classifying the sentence in one of the two classes. But all the classified summary sentences may not be used to represent the documents. If only some of the summary sentences are required to represent the full document there is no way to select them (as the purpose of summarization is to use a smaller representative part of document).

To handle this problem a system is developed that gives a score to each sentence, which is the probability that the sentence is a summary sentence given the feature values instead of comparing the values to be in one of the classes (a "summary" or "not a summary"). The system developed reads sentences in a document and checks the four features and then assigns their values accordingly (i.e. the probability of the features given a sentence is in a summary, which is taken from the model developed).

Then the system chooses sentences with the highest score as a summary sentence until the desired compression length is reached. The overall design of feature value collection and probability estimation in the training program and the prototype is shown below.

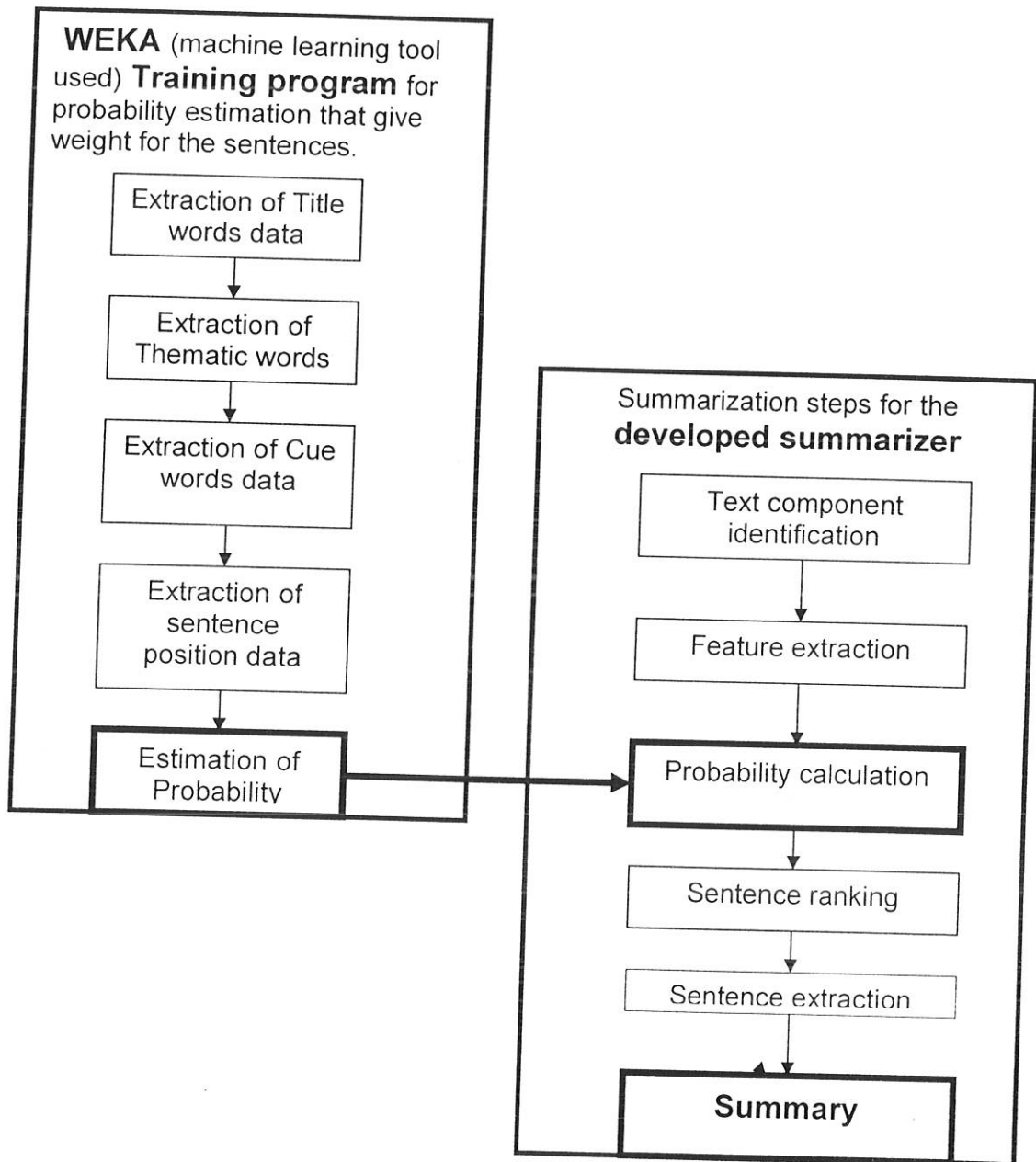


Figure 1. Design of the summarizer

The code for the implementation of the summarizer is developed using VB (Visual Basic 6.0). The code introduces the feature values which were collected in the training program (model development) of the final experiment of all the features. These have been set as variables in the program code, with each value being the appropriate probability. The algorithm of the prototype is as follows:

1. Initialize stop word list.
2. Initialize a document.
3. Read a single Amharic character.
4. Check Amharic word delimiter.
5. If it is not a delimiter,
 - Store it by concatenating on a temporary variable.
 - 5.1. Go to step 3.
6. Else
 - Check the word if it is already read.
 - If the word is found,
 - 6.1. Increment the respective frequency by 1.
 - else
 - 6.2. Assign the contents of the temporary variable to words array.
 - 6.3 assign 1 to the frequency
 - 6.4 Increment the word counter by 1.
7. Repeat step 3 until the end of the text.
8. Sort the words array by descending frequency.
9. for each word in the words array,
 - 9.1. Compare it with stop word list
 - 9.1.1. If found,
 - Remove the word from the array
 - 9.2. Repeat step 9.
10. Store the top three words.
11. Read the document title words using steps 3 to 7.

12. Compare the title words with the list of stop words using step 9.

13. Store the title words.

14. Split the document in to sentences.

14.1. for each sentence,

14.1.1. Check the location information

14.1.2. Assign their location value estimated from the training program.

Next

14.2. for each sentence,

14.2.1. for each title word,

14.2.1.1. Check the title word in the sentence.

14.2.1.2. Assign title word value estimated from the training program.

Next

14.2.2. for each cue word,

14.2.2.1. Check the cue word in the sentence.

14.2.2.2. Assign cue word value estimated from the training program.

Next

14.2.3. for each selected high frequent word,

14.2.3.1. Check the high frequent word in the sentence.

14.2.3.2. Assign high frequent word value estimated from the training program.

Next

Calculate sentence weight by multiplying location value, title value, cue word value, high frequent word value, and the probability of a sentence being in a summary.

Sort sentences in descending order of their values

Output sentences to the desired compression rate.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

Since the introduction of automatic method for text summarization by Luhn in 1958, with an analysis of a document's key word terms to produce a summary, a large number of researches have been done over the years using different techniques including a machine learning approach. Automatic text summarization is of great importance in the current world in which the society is facing information over load.

This research has presented a machine learning technique (naïve Bayes) to automatic text summarization of Amharic news articles. Using the approach a sentence based summarization system has developed, which combines ideas from different automatic summarization techniques.

This research consists of a training program, and development of a summarizer based on the results of the training program. The benefit of using a training program is acknowledged by Kupiec et al. (1995). The training program allows an analysis of the data and collection of values, which is used for the implementation of a summarizer.

For the purpose of this research 480 Amharic news articles were used. These articles were partitioned into two sets. The first set consisted of 100 of the articles for initial experiment and the remaining 380 articles for final experiment. The initial experiment was used for some preliminary investigation such as for ensuring feature values were accurately allocated and for setting some system parameters. The final experiment was performed with the 380 articles for development of the models and estimation of feature

weights. All of the results of the preliminary and the final experiments were obtained using a 10-fold cross validation (this is preferable to smaller dataset).

Overall, it was able to see that taking single features results in very poor results in predicting sentences for a summary. In this regard, however, the location feature is the only feature that showed marked difference as compared to the rest individual features. Therefore, the results were able to show the use of multiple features as a combination yielding better results in predicting summary sentences. Accordingly, this made the selected method a good option to be considered as an approach to be implemented in text summarization areas. Recommendations here forth are made to enable even greater performance outputs.

5.2 RECOMMENDATIONS

In this study an attempt was made to investigate the applicability of a machine learning (naïve Bayes) approach for automatic summarization of Amharic news texts. Experiments have been conducted for selection of feature sets and estimation of feature values for development of prototype summarizer. Some of the results of the experiment were not satisfactory so that there is a lot to be done in the future by researchers.

The following recommendations are some of the options for future research.

1. Availability of standard Amharic corpus for the purpose of text summarization would possibly facilitate researches in the areas of automatic summarization. It should be noted that developing a manual summary is a difficult task. The Rath (1961) study as in Kupiec et al. (1995) notes that the overlap in manually tagged documents is very low, and even the same person changes tagging of sentences over time.
2. Analysis of single features - this is because some of the features did not help in the prediction of sentences for the summary. The analysis on the cue words feature showed only 19.93% of the sentences in the extract contained one or more of the words. Therefore, with a greater analysis of the corpus, the cue words list could possibly be re-developed.
3. Testing for other feature – using other features such as tf.idf, and sentences length in addition to the feature sets used might give good result in predicting summary sentences.
4. Finally, the availability of stop-word list would possibly facilitate researches in the areas of automatic summarization. Nevertheless, there is no available stop word list for use in the Amharic language in general and specifically for news articles. Therefore; stop-word list should be developed.

REFERENCES

- Amini, M. and Gallinari, P. 2002 The Use of Unlabeled Data to Improve Supervised Learning for Text summarization.
URL= [www.connex.lip6.fr/~amini/ Publis/UseUnlTxtSum_sigir02.pdf](http://www.connex.lip6.fr/~amini/Publis/UseUnlTxtSum_sigir02.pdf).
- Beaza-Yates, R. And Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison Wesley. ACM Press, A division of the association for computing machinery.
- Beletu Reda. 1982. A Graphemic Analysis of the Writing System of Amharic. Paper for the Requirement of the Degree of bachelor of Art in Linguistics. Addis Ababa University. Addis Ababa.
- Bender, M. L. et al. 1976. *Language of Ethiopia*. London: Oxford University Press.
- Brunn, M. et al. 2001. Text summarization Using Lexical Chains.
URL= <http://www-nlpir.nist.gov/projects/duc/pubs.html>
- Chinchor et al. 1998. The hub-4 ie-ne task definition version 4.8.
URL= http://www.nist.gov/speech/tests/bnr/hub4_98/hub4_98.htm.
- Chuang, W. T., and Yang, J. 2000. Extracting Sentence Segments for Text Summarization: A Machine Learning Approach.
URL= <http://widit.slis.indiana.edu/irpub/SIGIR/2000/pdf21.pdf>
- Constantin et al. n.d. a computer-aided summarization tool. Research group in Computational linguistics. University of wolverhpton.
URL= clg.wlv.ac.uk/projects/CAST/cast-final-big.pdf
- Dalianis, H. 2000. SweSum - A Text Summarizer for Swedish.
<http://www.dsv.su.se/~hercules/papers/Textsumsummary.html>
- Edmundson, H. P. 1969. New methods in automatic extraction. *Journal of the ACM*. 16(2) pp 264-285.
- Farzindar, A. and Lapalme, G. 2004. Using Background Information for Multi-document Summarization and Summaries in Response to a Question.
URL= <http://rali.iro.umontreal.ca/Publications/urls/0Farzindar-Lapalme-Duco3>

- McCallum, A. and Nigam, K. 1998. "A Comparison of Event Models for Naive Bayes Text Classification." In AAAI / ICML-98 Workshop on Learning for Text Categorization. pp. 41-48.
 URL= <http://citeseer.nj.nec.com/mccallum98comparison.html>
- Mitchell, Tom M. 1997. *Machine Learning*. New York: The McGraw-Hill Companies, Inc.
- Moens, M. 2000. *Automatic Indexing and Abstracting of Document Texts*. Belgium: Kluwer Academic Publishers.
- Nega Alemayehu & Willet, P. 2002. Stemming of Amharic Words for Information Retrieval. *Literary and Linguistic Computing*, Vol-17, No. 1.
- Neto, J. L and Santos, A, D. 2000. A trainable algorithm for summarizing news stories.
 URL= chirouble.univ-lyon2.fr/~pkdd2000/Download/WS4_04.pdf
- Neto, L. et al. 2002. In G Bittencourt and GL Ramalho, editors, proc. 16 th Brazilian symp. On Artificial Intelligence (SBIA-2002). Lecture notes in artificial intelligence 2507, pages 205-215. springer-verlag.
- Neto, L. et al. n.d. Automatic Text Summarization using a Machine Learning Approach.
 URL= www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/SBIA-2002-Joel.pdf.
- Nomoto, T. and Matsumoto, Y. 2003. The diversity- based approach to open-domain text summarization. *Information processing and management*. 39:363-389.
- Radev, D.R. 2002. Introduction to the Special Issue on Summarization. *Association for Computational Linguistics*.
- Russel, S., Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.
- Russel, S., Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.
- Saba Amsalu. 2001. The Application of Information Retrieval Techniques to Amharic Documents on the Web. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.
- Saggion, H. Et al. n.d. Robust Generic and Query-based Summarization.
 URL= www.dcs.shef.ac.uk/~saggion/papers/189.pdf

- School of Information Studies for Africa. Addis Ababa University. Addis Ababa.
- Van Rijsbergen, C.J. 1979. *information retrieval*. Butterworth, London.
- Visser, W.T. and Weiling M.B. n.d. sentence-based summarization of scientific documents. The design and implementation of an online available automatic summarizer. URL= <http://www.martijnwieling.nl>
- Witten, Ian H. and Frank, E. 2000. *Data mining: Practical Machine Learning Tools and Techniques with Java implementations*. San Francisco: Morgan Kaufmann publishers.
- Yona, S. 2003. A Presentation introducing automatic summarization and sections. URL= yeda.cs.technion.ac.il/~yona/talks/ir_presentation.pdf
- Zahu, L. and Hovy, E. 2003. A web-trained extraction summarization system. *In proceedings of HLT-NAACL*, pages 205-211, Edmonton.
- Zhang et al. n.d. The Learnability of Naïve Bayes. *Lecture Notes in Computer Science*. <http://citeseer.nj.nec.com/434974.html>.
- Zelalem Sintayehu. 2001. Automatic classification of Amharic news items: The case of Ethiopian news agency (Master thesis). Addis Ababa University. Addis Ababa.
- Zeng, Z.1998. Naïve Bayesian Classifier committees. *In proceedings of ECML*, Berlin: Springer Verlag, 196-207.

Web sites

(WWW1) DARPA – Defense Advanced Research Projects Agency,
Unites States Department of Defense.

<http://www.darpa.mil>

(WWW2) <http://www.copernic.com/index.html>

(WWW3) <http://www.sinope.nl/en/sinope/index.html>

(WWW4) Weka Machine Learning Project.

<http://www.cs.waikato.ac.nz/~ml/>.

APPENDICES

Appendix I: sample of source news articles and summaries taken for the experiment

docids	docname	docText	summary
16	U18	በቀጣይ አምስት ዓመት ለደስ በሽ ቁጥጥር ትኩረት ይሰጣል አዲስ አበባ ሕዳር 1/1993/ዋኢማ/ በእንስሳት ገበያ በሰፊው ለመሳተፍ በሚቀጥሉት አምስት ዓመት የደስ በሽ ቁጥጥር ላይ ከፍተኛ ትኩረት እንደሚሰጥ በግብርና ሚኒስቴር የመላው አፍካ ደስ በሽ ማስወገድ ብሔራዊ አስተባባ አስወቁ። አስተባባው ዶክተር ስለሺ ዘውዴ ዛሬ ለዋል ኢንፎርሜሽን ማዕከል እንዳስወቁት የደስ በሽን ከዓለም በ2 ሺ 10 ነፃ ለማድረግ የቀደ ቢሆንም ኢትዮጵያ በእንስሳት ገበያ ያለምንም ችግር እንድትሳተፍ የተቀናጀ አሠራርን ተግባራዊ ለማድረግ ትዟል። ለዚህም ከተያዘው ዓመት ጀምሮ የድንበር ዘለል የእንስሳት በሽ ቁጥጥር አዋጅና በሽው ቢከሰት በአፋጣኝ እርምጃ ለመወሰድ የሚያስችል የአደጋ ዝግጁነት ሥልት እንደሚነደፍ ዶክተር ስለሺ አስውቀዋል። እንዲሁም በሁሉም ክልሎች የደስ በሽ ክትትል፣ ከዳጋ ከብቶች የደምናሙና በመውሰድ በላቦራቶ የመመርመር ሥራ እንዲሁም ከግብርና ባለሙያዎችና ከአርሶ አደሩ መረጃዎችን በመሰብሰብ ዕርምጃ እንደሚወሰድ ገልፀዋል። ባለፉት አምስት ዓመት በተደረገው ጥረት በአገራችን ምንም አይነት የደስ በሽ ምልክት እንዳልየና ክትባትም እንዳልተሰጠ የጠቆሙት አስተባባው፣ ሆኖም በሽው በአጎራባች ደቡብ ሱዳንና ሶማሊያ አካባቢ ሊከሰት ይችላል የሚል ግምት በመኖሩ በአካባቢዎቹ በሚገኙ 46 ወረዳዎች ልዩ ትኩረት እንደሚሰጥ ተናግረዋል። በሽውን ለመቆጣጠር በሚደረገው አሠራር የአንድ አገር ጥረት ብቻ ወሳኝ ባለመሆኑ አጎራባች አገሮች የበኩላቸውን ድርሻ እንዲወጡ ጥያቄ እንደሚቀርብላቸው አስውቀዋል። በተመሳሳይ በሚቀጥሉት አምስት ዓመት የቀንድ ከብቶች የሳንባና ሌሎች ተዛማጅ በሽዎችን ለመቆጣጠር የተቀናጀ	በቀጣይ አምስት ዓመት ለደስ በሽ ቁጥጥር ትኩረት ይሰጣል አዲስ አበባ ሕዳር1/1993/ዋኢማ/ በእንስሳት ገበያ በሰፊው ለመሳተፍ በሚቀጥሉት አምስት ዓመት የደስ በሽ ቁጥጥር ላይ ከፍተኛ ትኩረት እንደሚሰጥ በግብርና ሚኒስቴር የመላው አፍካ ደስ በሽ ማስወገድ ብሔራዊ አስተባባ አስወቁ። አስተባባው ዶክተር ስለሺ ዘውዴ ዛሬ ለዋል ኢንፎርሜሽን ማዕከል እንዳስወቁት የደስ በሽን ከዓለም በ2 ሺ 10 ነፃ ለማድረግ የቀደ ቢሆንም ኢትዮጵያ በእንስሳት ገበያ ያለምንም ችግር እንድትሳተፍ የተቀናጀ አሠራርን ተግባራዊ ለማድረግ ትዟል። ለዚህም ከተያዘው ዓመት ጀምሮ የድንበር ዘለል የእንስሳት በሽ ቁጥጥር አዋጅና በሽው ቢከሰት በአፋጣኝ እርምጃ ለመወሰድ የሚያስችል የአደጋ ዝግጁነት ሥልት እንደሚነደፍ ዶክተር ስለሺ አስውቀዋል።

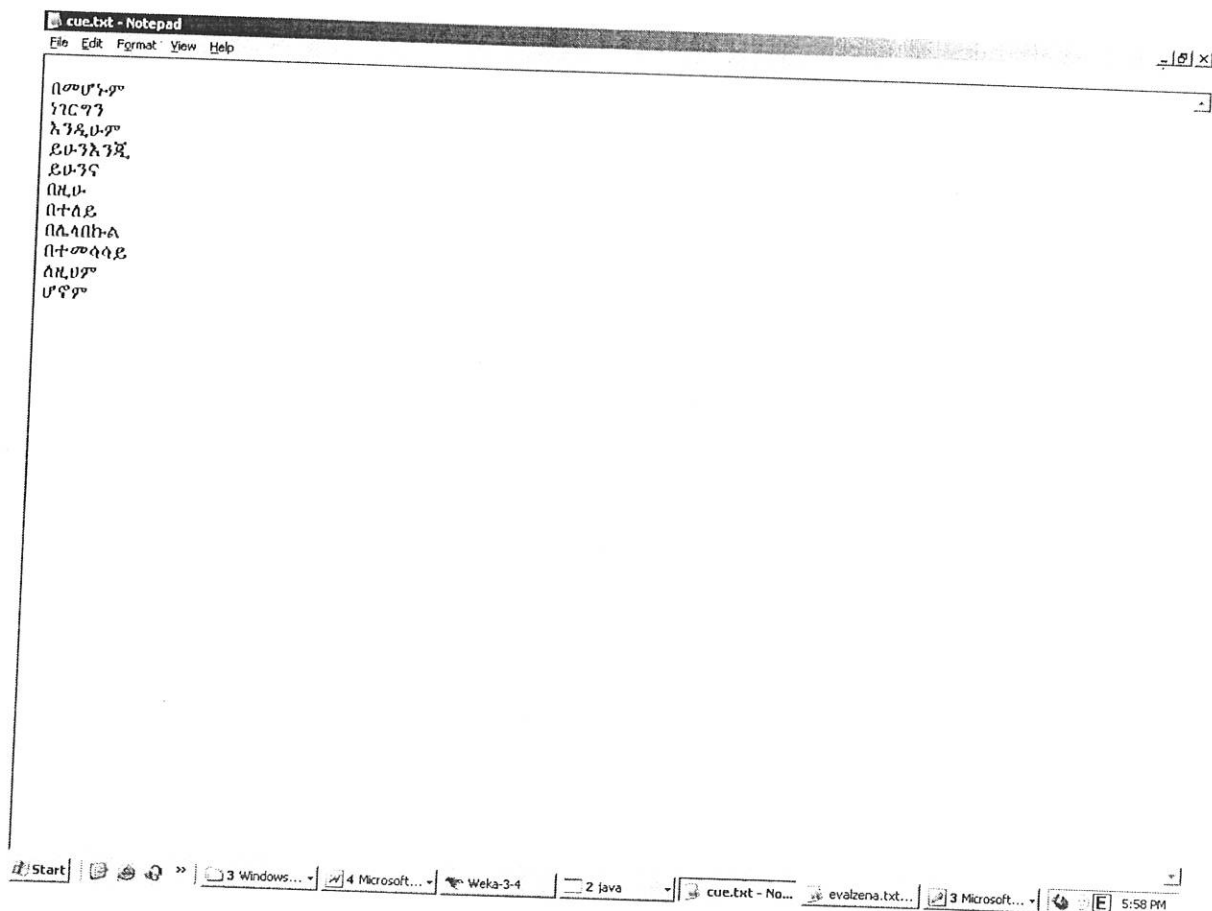
		ሥልት እንደሚቀየረው በሽዎችም በክትባትና በመድኃኒቶች ለማጥፋት ጥረት እንደሚደረግ አስተባባው አስረድተዋል። በአሁኑ ወቅት በአገራችን ከሚገኙ ወረዳዎች 30 በመቶ ያህል ብቻ የእንስሳት በሽ መረጃ እንደሚያስተላልፉ የጠቆሙት ዶክተር ስለሺ በቀጣይ ወደ 80 በመቶ ለማሳደግ መቀዱንና ለዚህም የመስተዳድር አካላት ትብብር እንዲያደርጉ ጠይቀዋል። በመሆኑም አርሶ አደሩ በቀበሌው የተለየ የእንስሳት በሽ በሚከሰትበት ወቅት በአቅራቢያው ለሚገኙ የግብርና ባለሙያዎች እንዲያሳውቅ አሳስበዋል። ለመርሐ ግብሩ ተግባራዊነትም የአውሮጳ ሕብረት 34 ነጥብ 3 ሚሊየን ብር እና የኢትዮጵያ መንግሥት 20 ሚሊየን ብር መመደባቸውን አስተባባውን ጠቅሶ ዋል ኢንፎርሜሽን ማዕከል ዘግቧል።	
17	ሀ19	ኬሚካሎችን ማስወገጃ 37 ሚሊየን ብር የሚጠጋ የገንዘብ ድጋፍ ተገኝ አዲስ አበባ ሕዳር 1/1993/ዋኢማ/ በኢትዮጵያ ጊዜ ያለፈባቸውና ከጥቅምት ውጪ የሆኑ የፀረ - ሰብል ኬሚካሎችንና ቁሳቁሶች ክምችት ለማስወገድ የ36 ነጥብ 9 ሚሊየን ብር ድጋፍ መገኘቱን የግብርና ሚኒስትሩ አስወቀ። የኬሚካሎቹን አወጋገድ አስመልክቶ ዛሬ በግብርና ሚኒስቴር አዳራሽ በተከፈተ አውደ ጥናት ላይ ሚኒስትሩ ዶክተር መንግሥቱ ሁሉቃ እንደተናገሩት በየክልሎች በ402 ቦታዎች ተከማችቶ የሚገኘውን 1 ሺ 500 ቶን ፀረ - ተባይ ኬሚካልን ለማስወገድ ባለፈው ዓመት በቀረበ የድጋፍ ጥያቄ መሠረት እስከ አሁን የገንዘብ ድጋፍ ተገኝቷል። የኢትዮጵያ መንግሥትና የዓለም የምግብና እርሻ ድርጅት የተለያዩ መንግሥትና ዕርዳ ሰጪ ድርጅቶችን በጠየቁት እገዛ መሠረት ከኔዘርላንድስ መንግሥት 18 ነጥብ 6 ሚሊየን ብር፣ ከሲዊድን ዓለም አቀፍ የዕርዳ ድርጅት 10 ነጥብ 3 ሚሊየን	ኬሚካሎችን ማስወገጃ 37 ሚሊየን ብር የሚጠጋ የገንዘብ ድጋፍ ተገኝ አዲስ አበባ ሕዳር 1/1993/ዋኢማ/ በኢትዮጵያ ጊዜ ያለፈባቸውና ከጥቅምት ውጪ የሆኑ የፀረ - ሰብል ኬሚካሎችንና ቁሳቁሶች ክምችት ለማስወገድ የ36 ነጥብ 9 ሚሊየን ብር ድጋፍ መገኘቱን የግብርና ሚኒስትሩ አስወቀ። የኬሚካሎቹን አወጋገድ አስመልክቶ ዛሬ በግብርና ሚኒስቴር አዳራሽ በተከፈተ አውደ ጥናት ላይ ሚኒስትሩ ዶክተር መንግሥቱ ሁሉቃ እንደተናገሩት በየክልሎች በ402 ቦታዎች ተከማችቶ የሚገኘውን 1 ሺ 500 ቶን ፀረ - ተባይ ኬሚካልን ለማስወገድ ባለፈው ዓመት በቀረበ የድጋፍ ጥያቄ መሠረት እስከ አሁን የገንዘብ

		ብር ዕርዳ መገኘቱን ገልፀዋል። በተጨማሪም ከዩናይትድ ስቴትስ ዓለም አቀፍ የዕርዳ ሰጪ ድርጅት የ8 ሚሊዮን ብር እገዛ እንደተገኘ የጠቆሙ ዶክተር መንግሥቱ፣ በሌሎች ዕርዳ ሰጪ ድርጅቶችም የመሳያዎች ድጋፍ እየተጠበቀ እንደሆነ ተናግረዋል። ፀረ - ተባይ ኬሚካሎች ያለጥቅም ሊከማቹ የቻሉት በተለያዩ ጊዜያት ወደ አገር ውስጥ ሲገቡ ከአስመጪ ድርጅቶች ተገቢውን ቁጥጥር ሳይደረግባቸው መሆኑን ሚኒስትሩ አስውቀዋል። በ1987 ዓ.ም በተደረገው ጥናት መሠረት ከ1 ሺ 150 ሜትር ቶን ፀረ - ተባይ ኬሚካሎችና 100 ቶን የተመረዙ ኬሚካሎችና በርሜሎች ተከማችተው እንደሚገኙ ከዶክተር መንግሥቱ ገለፃ ለመረዳት ተችሏል።	ድጋፍ ተገኝቷል።
18	ሀ20	አንዲት ወይዘሮ ሶስት ሕፃናትን ተገባገላ ጎማ ሕዳር 1/1993/ዋኢማ/ በጅማ ዞን ጎማወረዳ አንዲት እናት ያለ ሃኪም ዕርዳ ሶስት ሴት ሕፃናትን መገላገላቸውን የዞኑ ፖሊስ አስወቀ። በወረዳው የዴዶ ኢሬቻ ቀበሌ ነዋ የሆኑት የ35 ዓመቷ ወይዘሮ ከትላንት በስቲያ ሕፃናቱን በቤቸው ውስጥ የተገባገሉት በ3፣ በ2ና በአንድ ሰዓት ልዩነት ውስጥ መሆኑን በመምያው የወንጀል መከላከል ክፍል ሹም አስር አለቃ ደረጃ ከበደ ገልፀዋል። ወይዘሮዋ ቀደም ብለው የተገባገሏቸው ሁለቱ ሕፃናት ሙሉ አካል ሲኖራቸው መጨረሻ የተገባገሏት አንገትና ራስ የሌላትና ከአንገቷ በች ግን ሙሉ ሰውነት ያላት መሆኑን ሹሙ ተናግረዋል። ከተወለዱትም ሁለቱ ወዲያውኑ ሲሞቱ ጤናማ አካል ያላት አንዷ ሕፃን ግን ከወላጅ እናቷ ጋር በአጋሮ ከተማ በሕክምና ላይ እንደምትገኝ ገልፀዋል። ወይዘሮዋ ስለ ሁኔው ተጠይቀው ቀደም ሲል አንድ ልጅ በደህና ሁኔ መውለዳቸውንና የዘንድሮው ዕንስ ግን እየከበዳቸውና የተለየ ስሜት እያሳደረባቸው እንደ ነበር ገልፀው፣ በእርግዝና ጊዜ የሚካሄዱትን ምርመራዎችንም	አንዲት ወይዘሮ ሶስት ሕፃናትን ተገባገላ ጎማ ሕዳር 1/1993/ዋኢማ/ በጅማ ዞን ጎማ ወረዳ አንዲት እናት ያለ ሃኪም ዕርዳ ሶስት ሴት ሕፃናትን መገላገላቸውን የዞኑ ፖሊስ አስወቀ። በወረዳው የዴዶ ኢሬቻ ቀበሌ ነዋ የሆኑት የ35 ዓመቷ ወይዘሮ ከትላንት በስቲያ ሕፃናቱን በቤቸው ውስጥ የተገባገሉት በ3፣ በ2ና በአንድ ሰዓት ልዩነት ውስጥ መሆኑን በመምያው የወንጀል መከላከል ክፍል ሹም አስር አለቃ ደረጃ ከበደ ገልፀዋል።

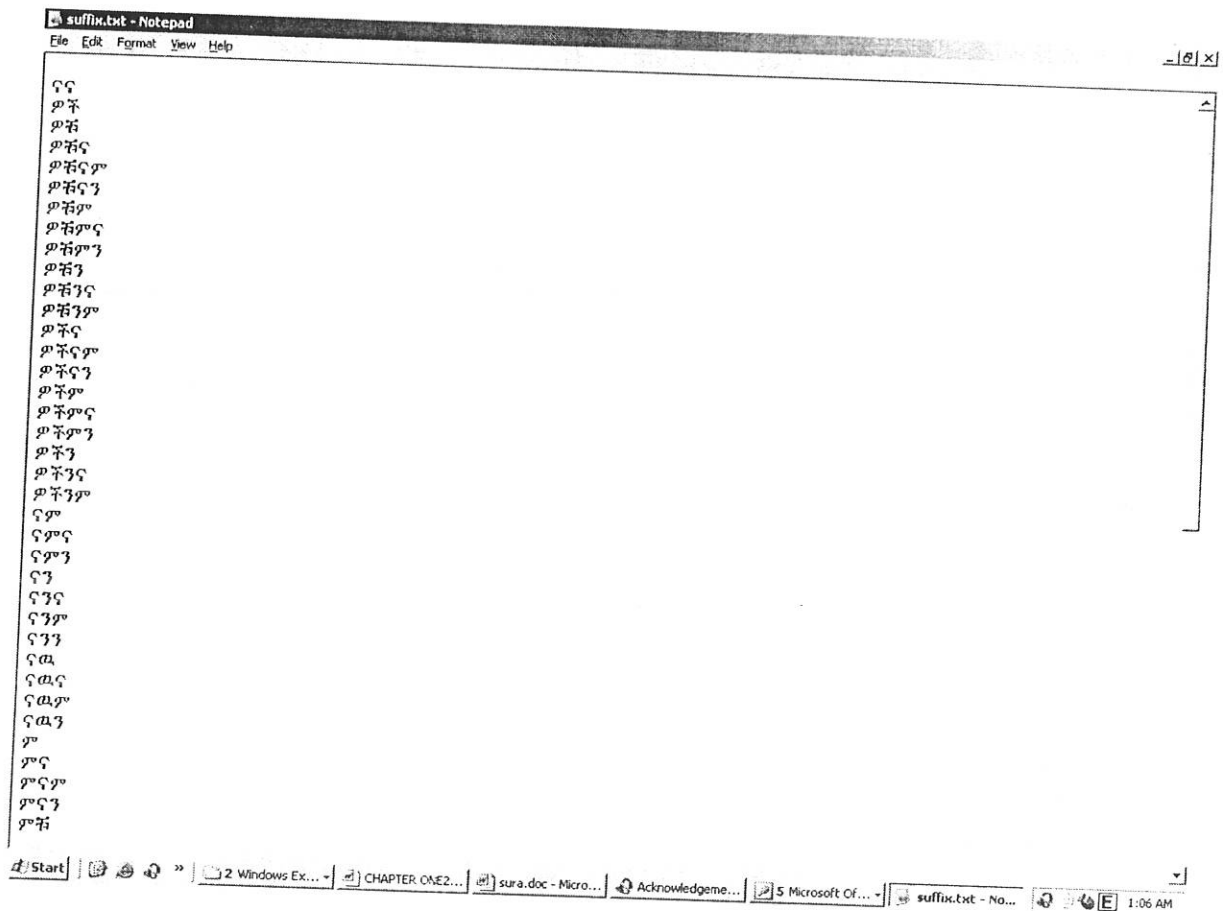


	አለማድረጋቸውን ተናግረዋል። በቀበሌው ጤና ጣቢያ የእናቶችና ሕፃናት ጤና ክትትል ኃላፊ ሲስተር አብነት በቀለ በበኩላቸው መን ሕፃናት ሊወለዱ የሚችሉት በሴት ማሕፀን ሁለትና ከዚያ በላይ የተዘጋጀው እንቁላል ከወንድ የዘር ፍሬ ሲገናኝ እንደሆነ ገልፀዋል። ሙሉ አካል የሌለው ሕፃን ሊወለድ የሚችለው ግን ወላድ ሴቶች ፅንሱን ለማስወጣት መድኃኒት ሲጠቀሙና አንዳንዴም የፅንሱ አቀማመጥ ያልተስተካከለ ሲሆን ነው ማለቸውን ዋል ኢንፎርሜሽን ማዕከል ዘግቧል።	
--	--	--

Appendix II: List of cue words used



Appendix III: some of the suffixes used for suffix removal



]	]#	É!	É	É@	}	ò
ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ
[	[	[!	Ú	[@	A	Ů
ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ
f	û	ö	í	Ø	F	æ
T	T	T	T	T	T	T
p	p\$	pE	-	p&	P	±
ñ	ñ	ñ	ñ	ñ	ñ	ñ
v	v#	v!	Š	v@	V	ˋ
ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ
%	Đ	—	—	,	"	a
ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	
i	•	...	þ	à	O	
ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ
þ	a	ÿ	ì	ù	ˋ	Æ*

Appendix V. Evaluation performance of individual features

== Run information ==					
Scheme: weka.classifiers.bayes.NaiveBayes					
Attributes: 2					
Title words					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1615			72.0982 %	
Incorrectly Classified Instances	625			27.9018 %	
Kappa statistic	0				
Mean absolute error	0.3794				
Root mean squared error	0.4355				
Relative absolute error	94.2889 %				
Root relative squared error	97.0976 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	summary
1	1	0.721	1	0.838	non-summary
=== Confusion Matrix ===					
a	b	<-- classified as			
0	625		a = summary		
0	1615		b = non-summary		

=== Run information ===					
Scheme: weka.classifiers.bayes.NaiveBayes					
Attributes: 2					
Cue words					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1615			72.0982 %	
Incorrectly Classified Instances	625			27.9018 %	
Kappa statistic	0				
Mean absolute error	0.4018				
Root mean squared error	0.4482				
Relative absolute error	99.8514 %				
Root relative squared error	99.9376 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	summary
1	1	0.721	1	0.838	non-summary
=== Confusion Matrix ===					
a b <-- classified as					
0	625	a = summary			
0	1615	b = non-summary			

== Run information ==					
Scheme: weka.classifiers.bayes.NaiveBayes					
Instances: 2240					
Attributes: 2					
Content words					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1615			72.0982 %	
Incorrectly Classified Instances	625			27.9018 %	
Kappa statistic	0				
Mean absolute error	0.3615				
Root mean squared error	0.4251				
Relative absolute error	89.8342 %				
Root relative squared error	94.7827 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	summary
1	1	0.721	1	0.838	non-summary
=== Confusion Matrix ===					
a	b	<-- classified as			
0	625		a = summary		
0	1615		b = non-summary		

=== Run information ===					
Scheme: weka.classifiers.bayes.NaiveBayes					
Attributes: 2					
Location					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1927			86.0268 %	
Incorrectly Classified Instances	313			13.9732 %	
Kappa statistic	0.6525				
Mean absolute error	0.2273				
Root mean squared error	0.3368				
Relative absolute error	56.4885 %				
Root relative squared error	75.1015 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.749	0.097	0.75	0.749	0.749	summary
0.903	0.251	0.903	0.903	0.903	non-summary
=== Confusion Matrix ===					
a	b	<-- classified as			
468	157	a = summary			
156	1459	b = non-summary			

Appendix VI. Evaluation performance of combined features

= Run information ===					
Scheme: weka.classifiers.bayes.NaiveBayes					
Attributes: 3					
Title words					
Cue words					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1615	72.0982 %			
Incorrectly Classified Instances	625	27.9018 %			
Kappa statistic	0				
Mean absolute error	0.3787				
Root mean squared error	0.435				
Relative absolute error	94.1096 %				
Root relative squared error	96.9893 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	summary
1	1	0.721	1	0.838	non-summary
=== Confusion Matrix ===					
a	b	<-- classified as			
0	625	a = summary			
0	1615	b = non-summary			

= Run information ===					
Scheme: weka.classifiers.bayes.NaiveBayes					
Attributes: 4					
Title words					
Content words					
Location					
Class					
Test mode: 10-fold cross-validation					
=== Stratified cross-validation ===					
=== Summary ===					
Correctly Classified Instances	1921	85.7589 %			
Incorrectly Classified Instances	319	14.2411 %			
Kappa statistic	0.6413				
Mean absolute error	0.2093				
Root mean squared error	0.3466				
Relative absolute error	51.9993 %				
Root relative squared error	77.2879 %				
Total Number of Instances	2240				
=== Detailed Accuracy By Class ===					
TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.723	0.09	0.756	0.723	0.739	summary
0.91	0.277	0.895	0.91	0.902	non-summary
=== Confusion Matrix ===					
a b <-- classified as					
452	173		a = summary		
146	1469		b = non-summary		

DECLARATION

This thesis is my original work, has not been presented for a degree in any other university and all sources of material used for the thesis have been duly acknowledged.

Teferi Andargie Achenef

June, 2005

The thesis has been submitted for examination with my approval as university advisor

Tesfaye Biru

June, 2005