TEXT SUMMARIZATION USING ANT COLONY OPTIMIZATION ALGORITHM

تلخيص النصوص باستخدام خوارزمية مستعمرة النمل المحسنة

A thesis submitted In Partial Fulfillment of The Requirements

For the Degree of Master in Computer Science

BY:

OMER FYSAL HASSAN

Supervisor:

Dr. Albaraa Abuobieda Mohammed

February 2015

ACKNOWLEDGEMENT

First and foremost, all praises be to the Almighty Allah.

I would also like to thank my both parents, my brothers and sisters for their love, support and supplications.

Finally, I would like to thank all my friends and colleagues for their support
And assistance.

Abstract

Automatic text summarization plays increasingly an important role with the exponential growth of documents on the Web. Numerous approaches have been proposed to identify important contents for automatic text summarization. Sentence scoring approaches mark scores for input sentences rank all of them decadently. Only higher ranked sentences are selected to be part of the summary. Find the informative sentences is an important issue for the researchers in an extractive based automatic text summarization. This research aim to use extraction based automatic single document text summarization method using evolutionary algorithm called Ant Colony Optimization algorithm ACO to produce good summaries. We use ACO algorithm to find out the best sub set feature weight score. To the best of our knowledge has never been used for solving text summarization problem before. To evaluate the proposed method standard dataset from Document Understanding Conference (DUC) 2002 in used and the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) as the standard evaluation measurement toolkit is used .Set of evolutionary algorithms are used in this research in term of evolutionary the experimental results showed our proposed method has performed well compared with algorithms (Particle Swarm Optimization methods and Genetic Algorithm). Although our targeted ACO algorithm is select to improve the performance of text summarization has not out performance the latest proposed method (Differential Evolution) but performance satisfactory.

المستخلص

التلخيص الالي للنصوص يلعب دورا متزايد الأهمية مع النمو الهائل للوثائق على شبكة الانترنت. وتم اقتراح عددة مناهج لتحديد المحتوى المهم من النصوص.منهجية حساب الدرجات للجمل تقوم علي اعطاء كل الجمل درجات و بعد ذلك يتم ترتيبها تصاعدياً ويتم اختيار الجمل ذات الدرجات الاحسن ليتم وضعها في الملخص.البحث عن الجمل المهمة في الوثائق يعتبر من اهم التحديات للباحثين في مجال التلخيص الالي.هذا البحث يهدف استخدام الية التلخيص بواسطة خوارزمية تطورية تعرف مستعمرات النمل .تم استخدام الخوارزمية للبحث عن احسن درجات من خلال افضل مجموعة جزيئة. إلى حد علمنا لم يتم استخدام الخوارزمية من أجل حل مشكلة تلخيص النصوص قبل. لتقييم الطريقة المقترحة، واستخدمت مجموعة البيانات القياسي تعرف ب(DUC)عام 2002و ايضاًاستخدمت ادوات (ROUGE) كمعيار لعملية التقييم. النتائج التجريبية تظهر أداء جيدا للنموذج المقترح مقارنة مع خوارزميات (طرق تحسين سرب الجسيمات، الخوارزميات الجينية). على الرغم من أن لدينا خوارزمية ACO المستهدفة هي مختارة لتحسين أداء تلخيص النص.تم مقارنتها مع احدث الطرق خوارزمية التفاضلية التطورية لم يكن الاداء مرضي.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGRES

LIST OF APPENDICES

1 Introduction

With the growth of structured information available on the Web and the amount of such information became extremely large. Looking for interesting information from the amount of information is a very hard task. In some cases, the navigation through millions of relevance documents frustrates the seeker of that information, making him or her think that interesting information is not available. Besides this, the discovery of a large number of documents one after the other is time consuming. Since Luhn's work (1958), automatic text summarization (ATS) researchers are trying to solve or at least relieve that problem by proposing techniques for generating summaries. These summaries serve as a quick guide to interesting information by providing a shorter form of each document in the document set.

In the same year (1958), a work related to Luhn's work was done by Baxendale (1958), where a summarizer based on the position of the sentence was introduced. Edmondson (1969) continued in the same way as Luhn and Baxendale by reusing two of the features which they have used in their studies (word frequency used by Luhn (1958) and positional importance used by Baxendale (1958)) in addition to another two features, which were pragmatic words (cue words, i.e., words would have positive or negative effect on the respective sentence weight like a significant, key idea, or hardly) plus title and heading words.

All above works were considered as surface-level approaches. They depend mainly on those shallow text features. The priority of a text unit to be included in the summary depends on the score of that unit. The text features are the cornerstone in the generation process of the text summary. The summary quality is sensitive to those features in terms of how the sentences are scored based on the chosen features. Therefore, the determination of the effectiveness of each feature

could lead to a mechanism that differentiates between the features having high importance and those with less importance. To this end, the effect of the feature structure on the feature selection is needed to be investigated, which will lead to finding and optimizing the feature weights.

The ant colony optimization algorithm (ACO) Marco origo (1992) has the ability to perform such a role and to learn the most optimized features weights.to the best of our knowledge, The ant colony optimization (ACO) has not been used for text summarization previously. however, it has been employed in related fields such as document clustering Łukasz Machnik(2005), Image Feature Selection(Ling Chen and Bolun Chen and Yixin Chen) and Spam Host Detection (Arnon Rungsawang, Apichat Taweesiriwate , Bundit Manaskasemsak, 2012).The ant colony optimization (ACO) as a machine learning algorithm will raise the problem of supervised machine learning approaches. The following are the reasons of why the ACO was chosen to solve the problem of automatic text summarization. A recent work published reported that the ACO algorithm has become quite popular in the machine intelligence and cybernetics communities. It has successfully been applied to different domains of science and engineering, such as mechanical engineering design signal processing and machine intelligence.

Naturally, the proposed methodologies are exposed to advantages and disadvantages. Although the optimization techniques are used to overcome some limitations of other proposed methods, they suffer many defects. Jun et al.,(2011) surveyed some evolutionary computing algorithms (ECAs). The survey discussed how the ECAs search performance could be optimized using machine learning techniques. This trend of research direction treated the term “Machine Learning for Evolutionary Computing (MLEC)” for the discussed purpose. The ECAs agreed in a general structure which includes the following stages: population initialization, fitness evaluation and selection, population reproduction and variation, algorithm

adaptation, and local search. The survey viewed the algorithm defects and the successful solutions.

This study will also propose a method that generates summaries by selecting sentences based on sentences score using important features and adjusting the weighting for each of them to improve the total results. The key point of this research is based on automatic summary extraction using ACO to extract features of the text as its summary for the original text.

The heart of the text summarization is evaluation is very important task. Manual evaluation is done by human or automatic by special tools. Two categories of methods used in text summarization are extrinsic and intrinsic (Jing et al., 1998; Mani and Maybury, 1999; Afantenos et al., 2005).The Recall-Oriented Understanding for Gisting Evaluation (ROUGE) (Lin, 2004), for example, is an automatic intrinsic evaluator of summary systems for the Document Understanding Conference (DUC). ROUGE is said to correlate highly with the results of human judgments of content and quality (Lin, 2004).

1.1Problem background

With the growth of information available on the Web and size of information became extremely large, this proved the need for generating high quality of automatic summary Automatic Text Summarization (ATS)main duty helps users to compress the information and present it in a brief way, in order to make it easier to process the vast amount of documents related to the same topic that exist these days.

1.2Problem statement

As we stated in the beginning of this chapter, designing a good mechanism for ATS is very sensitive task. The literature review showed that building a designing these mechanisms Evolution Computing Algorithms (ECA) s are very effective and successful. ACO is a promised algorithms and tented widely in similar research area as stated. This research aim to investigate the performance of ACO as feature scoring mechanism for ATS purpose. To evaluate its performance, ACO will be compared with similar ECAs designed for same purpose.

1.3Research quotations

The main questions which must be answered in dealing with such a problem are as follows:

1- Can selection of Features based a good document summary?
2- What are the important features that can be used to identify sentences that are important for including in a text summary?
3- What are the techniques that can be used for selection of features to produce a good summary?
4- Can feature selection and feature weight adjustment based on Ant Colony Optimization algorithm (ACO) produce a good summary?

1.4 Research objectives

The main goal of this study is to introduce a model of Ant Colony and examining its ability for effective text summarization. Based on that model, the hypothesis of the study can be stated as:

"Ant Colony algorithm could identify appropriate features to be build an excellent text summarization that can represent large document"

The specific objectives of the study are as follows:

1- To look for important features that can be used.
2- To propose a new model of ACO to produce a good text summarizer by considering feedback from summarization evaluation.

1.5 Scope

We will use the Experimental methodology in the study. The Ant Colony Optimization is trained using DUC 2002 dataset to learn the weight of each feature.

1.6 Research significance

This thesis contributes a framework for automatic text summarization we can use as web service on different application such as web search engine and summarize the news, articles, etc.

The usage of Ant Colony Optimization (ACO) in Feature Subset Selection (FSS) processes, the proposed ACO Based-FSS method is among the few feature subset selection methods that handle high dimensional data sets.

1.7 Thesis structure

This thesis is organized into four chapters, of which you are currently reading the first. The following chapters will form the foundation on which this thesis rests.

These chapters will be providing the problem of the thesis as a research field as well as detailing the methods and concepts relevant to the research carried out during the work with this thesis.

The concept of automatic text summarization, which is a term commonly used to denote summarization carried out by means of a computer program, is introduced. Followed by an overview of a selection of representative systems and

approaches, we here also find a brief look of summarization in Chapter2. Presents design, implementation of ACO algorithm, Feature Subset Selection concepts, steps, approaches in Chapter 3. Experimental results are discussed in Chapter4. We render a close look at the limitations of the new ACO-Based FSS, which is a main contribution of the thesis. Finally, we conclude with a peek into the future with some suggested directions.

CHAPTER 2
BACKGROUND AND RELATED WORK

2-1 Background

2-1-1 Introduction

This chapter introduces some of the existing approaches in automatic text summarization, background for this research. and the literature reviews . Furthermore, The most important evaluation measures of automatic text summarization are also presented.

2-1-2 Text Summarization

It is very difficult for human beings to manually summarize large documents of text. Summarization is a tool for assisting and interpreting text information in today's fast-growing information age.

Early experimentation in the late 1950s suggested that text summarization by computer was feasible though not straightforward. After some decades, progress in language processing, coupled with the growing presence of online text in corpora and especially on the web renewed interest in automated text summarization. So the huge amount of available electronic documents in Internet has motivated the development of very good information retrieval systems. However, the information introduce by such systems, like searched engine, only show bit of text where the words of the request query appears. Therefore, the user has to decide if a document is interesting only with the extracted piece of a text. Moreover, this part does not have any information if the retrieved document is interesting for the user, so it is necessary download and read each retrieved document until the user finds satisfactory information. It was not needed and time-consuming routine. A solution for such problems is to achieve a text summarization of the document extracting

the essential sentences of the document. There are various definitions on text summary in the literature and Spark Jones (1999), Mani (2001), Hovy (2005), and Fatah and Ren (2008) are among the researchers that define the word.

A summary as "a reductive transformation of source text into summary text through content condensation by selection and/or generalization on what is important in the source." Jones, S. (1999).

The other defines for summary "The aim of automatic text summarization is to condense the source text by extracting its most important content that meets a user's or application's needs." Hovy (2005).

Fattah and Ren (2008) said, "text summarization is the process of automatically creating a compressed version of a given text that provides useful information for the user."

2-1-3 Approaches to Automatic Text Summarization

Summarization approaches are often, divided into two main groups, text extraction and text abstraction. Extraction approach dependent on the selection of particular pieces of text from a document where the sentences and/or phrases with the highest score are considered as salient sentences and are chosen to form the summary.

Text abstraction, on the other hand, being the more challenging than, extraction. It necessarily to find out the main topic of the text and paraphrase it in a compressed form, interpret the text semantically into a formal representation, find new more concise concepts to describe the text and then generate a new shorter text, an

abstract, with the same basic information content. The difficulty of abstraction makes extraction more widely used in text summarization.

2-1-4 Summary Types

Summaries of text can be divided into different categories: indicative summary, informative summary, critical summary and extract. The most important summaries are indicative summary and informative summary. Informative this type of summary expresses the important factual content of the text. Indicative these summaries are meant to give the reader an idea as to whether it would be worthwhile reading the entire document. The topic and scope of the text should be expressed but not necessarily all of the factual content. The generated summaries can contain information from a single document (single document summaries) or from a collection of documents (multi-document summaries). Summarization process emphasizes mainly on two goals, high compression ratio and redundancy reduction, especially in multi-document summarization. These objects are accomplish by keeping the important ideas in each document, reducing the size of each document and comparing ideas across documents. This poses many challenges, as follows (Mani and Maybury, 1999):

- Identification of scaling algorithms.
- Redundancy removal.
- Intelligent ways are employed to exploit ordering among documents.
- The relationships are represented by effective presentation and visualizations strategies.

2-1-5 Automatic Text Summarization System

Automatic summarization is the creation of a briefer representation of a body

of information by a computer program. The product of this procedure should Still contain the most central facts of the original information. Automatic text Summarization, thus analogously, is the shortening of texts by computer, while still retaining the most important points of the original text.

Automatic text summarization is a multi-faceted endeavor indeed. If we first broadly define three aspects of a summarizing system as i) source, representing the multitude of input formats and possible origins of the information being summarized, ii) purpose being the intended use for the generated summary, and iii) composition, denoting the output format of the summary and the information contained therein, we can Then, according to(Spärck-Jones 1999, Lin and Hovy 2000, Baldwin et al. 2000)Among others, roughly make the following, by necessity inconclusive, division:

Source (Input):

- Source: single-document vs. multi-document.
- Language: mono-lingual vs. multi-lingual vs. cross-lingual.
- Genre: news vs. technical report vs. scientific paper etc.
- Specificity: domain-specific vs. general.
- Length: short (1–2 pages) vs. long (> 50 pages).
- Media: text, graphics, audio, video, multi-media etc.

Purpose:

- Use: generic vs. query-oriented (aimed to a specific information need).
- Purpose: what the summary is used for (e.g. alert, preview, inform, digest, provide biographical information).
- Audience: untargeted vs. targeted (aimed at a specific audience).

Composition (Output):

- Derivation: extract vs. abstract.
- Format: running text, tables, geographical displays, time lines, charts, illustrations etc.
- Partiality: neutral vs. evaluative (adding sentiment/values)

The automatic text summarization process consists of three stages (Mani, 2001):

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

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

- Selection of the most important and the diverse content.
- Aggregation of the selected content (put all together).
- Generalization (replacing specific content with more general content).

One of the main concepts in the ATS approach is that the reader should be aware how the system computes summary length. Equation (2.1) demonstrates this.

$$Summary - Length = \frac{(UserDefinedRatio \times DocumentLength)}{100} \quad (2.1)$$

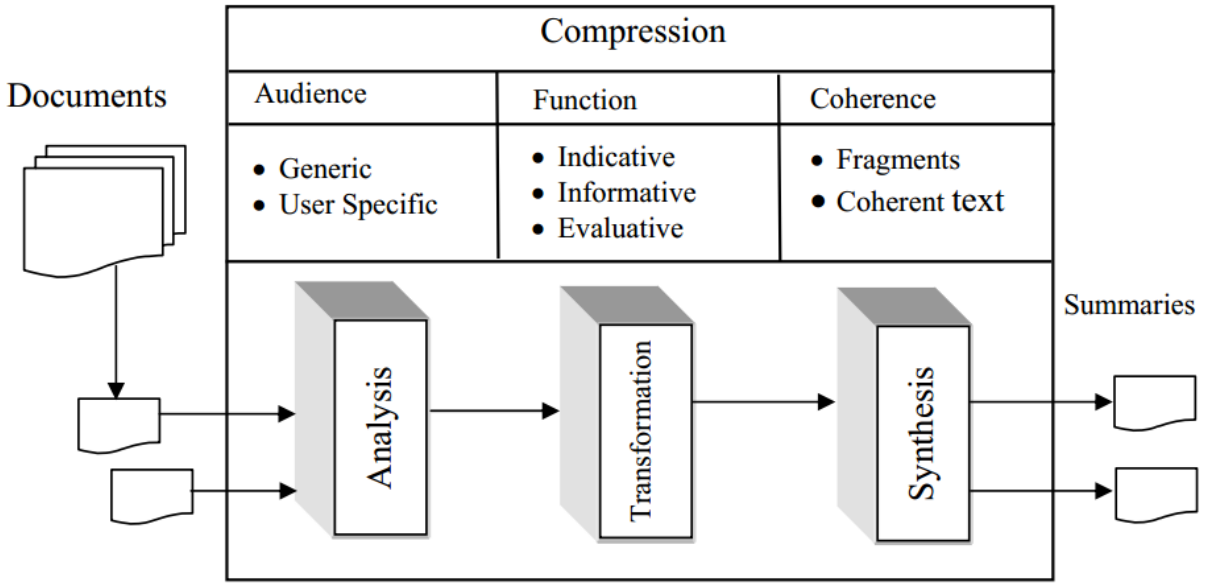

Figure 2 A typical automatic text summarization system (Mani and Maybury, 1999)

2-1-6 Summarization Applications

The application areas for text summarization are extensive. Information is published simultaneously on many media channels in different versions, for instance, a paper newspaper, web newspaper, sms message, mobile radio newscast, and a spoken newspaper for the visually impaired. Also, documents can be made accessible in other languages by first summarizing them before translation, which in many cases would be sufficient to establish the relevance of a foreign language document, and hence save human translators work since they need not translate every document manually. In particular, automatic text summarization can be used to prepare information for use in small mobile devices, such as a PDA, which may need considerable reduction of content.

2-1-7 Text Summarization Techniques

Summarization techniques can be classified into different kind of approaches. we investigate into two main categories : single document summarization techniques and multi-document summarization techniques.

2-1-7-1 Single Document Summarization

Usually, document is not uniform, which means that some parts are more important than others. The major challenge in summarization lies in distinguishing the more informative parts of a document from the less ones. This section, we describe some eminent extractive techniques. First, we look at early work from the 1950s research on summarization. Second, we concentrate on approaches involving machine learning techniques published in the 1990s. Finally, we briefly describe some techniques that we are use.

2-1-7-1-1 Early Work

Most early work on single-document summarization focused on technical documents. The Important paper on summarization is that of (Luhn, 1958), that describes research done at IBM in the 1950s. In his work, Luhn proposed that the frequency of a particular word in an article provides a useful measure of its significance. There are several key ideas put forward in this paper that have assumed importance in later work on summarization. As a begin step, words were stemmed to their root forms, and stop words were deleted. Luhn then compiled a list of content words sorted by decreasing frequency, the index providing a significance measure of the word. Then a sentence level, an importance factor was derived that indicate the number of occurrences of import words within a sentence, and the linear distance between them due to the intervention of non- import words. All sentences are ranked in order of their importance factor, and the top ranking sentences are finally selected.

2-1-7-1-2 Machine Learning based approaches

In the 1990s, with the advent of machine learning techniques in NLP, a series of seminal publications appeared that employed statistical techniques to produce document extracts. These approaches are categorized as either supervised-based learning methods or unsupervised-based learning methods. While initially most systems assumed feature independence and relied on naive-Bayes methods, others have focused on the choice of appropriate features and on learning algorithms that make no independence assumptions. We next describe some of these approaches in more detail.

2-1-7-1-3 Naive-Bayes Methods

(Kupiec, 1995) Explains a method Inherit from Edmundson (1969) that is able to learn from data. The classification function categorizes each sentence as worthy of extraction or not, using a naive-Bayes classifier. Let s be a particular sentence, S the set of sentences that make up the summary, and$F_1 \dots F_k$ the features. Assuming independence of the features:

$$P(s \in S|F_1, F_2, .. F_k) = \frac{\prod_{i=1}^{k} P(F_i|s \in S).P(s \in S)}{\prod_{i=1}^{k} P(F_i)} \qquad (2.2)$$

The features were obedient to (Edmundson, 1969), but additionally included the Sentence length feature is a sentence of higher length may not considered a suitable selection for inclusion in the summary, the presence of uppercase words feature, Sentence-position that a score sentences based on their position in a document, thematic-word feature to Identifying the most frequent words may assist in discovering optimal summary sentences as they hold most of the document's topics and fixed-phrase feature a sentence that is placed after a section header is considered important . Each sentence was given a Score according to the upper equation, and only the Excellent marks sentences will be extracted

The Bayesian classifier is based on these five features after which a binary probability for each sentence plays an important role in deciding whether or not to include the sentence in the summary.

2-1-7-1-4 Rich Features and Decision Trees

At the end of the nineties decade Lin and Hovy (1997) studied the importance of a single feature, sentence position. Sentence position. Just weighing a sentence by its position in text arises from the idea that texts generally follow a predictable discourse structure, and that the sentences of greater topic centrality tend to occur in certain specifiable locations (e.g. title, abstracts, etc). However, since the discourse structure significantly varies over domains, the position method cannot be defined as naively as in (Baxendale, 1958). Then ranked the sentence positions by their average yield to offer the Optimal Position Policy (OPP) for topic positions.

Lin (1999) broke away from the hypothesis that features are independent of each other and tried to model the problem of sentence extraction using decision trees, instead of a naive-Bayes classifier. He examined a lot of features and their effect on sentence extraction.

Some new features were the query signature (normalized score given to sentences depending on number of query words that they contain), IR signature (the m most salient words in the corpus, similar to the signature words of (Aone et al.,1999)), numerical data (binary value 1 given to sentences that contained a number in them), title proper name (binary value 1 given to sentences that contained a proper name in them), pronoun (binary value 1 given to sentences that contained a pronoun or adjective in them), weekday or month (like as previous feature) and

quotation (like as previous feature). Noting that some features like the query signature are question-oriented.

From previous experience prefer using only the positional feature, or using a simple combination of all features by adding their values. When evaluated by matching machine extracted and human extracted sentences, the decision tree classifier was clearly the better for the whole dataset, but for three topics, a naive combination of features beat it. Lin conjectured that this happened because some of the features were independent of each other. Feature analysis suggested that the IR signature was a valuable feature, corroborating the early findings of(Luhn ,1958).

With the millennium (Conroy and O'leary, 2001) give a method to produce the generic extracts using a hidden Markov model that decide the likelihood of the including sentence in/excluding the sentence from the summary.

(Neto ,2002) used two kind of machine learning approaches: Naive Bayes and C4.5 which is a decision-tree algorithm to produce a trainable text summarizer. The set of the extracted features extracted from the original text is used to classify the sentences into summary sentences and un-summary sentences. The results showed that Naive Bayes classifier-based method outperforms the C4.5 classifier-based method.

Shen et al. (2007) use a machine learning method for single summarization based on a Conditional Random Fields (CRF) (Lafferty et al., 2001). The method assigns a sentence with 1 if it is worth to be a summary sentence or 0 if it is not worth to be a summary sentence. In this method the labeling proceeds in sequent manner and it is not fix depending on the relationship of the sentence with other sentences.

Fattah and Ren (2008), as section of their work, trained GA for producing weights of the features, where the average precision was used as fitness function, the method was proposed for single document summarization.

Liangda Li , Ke Zhou , Gui-Rong Xue, Hongyuan Zha, and Yong Yu (2009) do formulate the extract-based summarization problem as learning a mapping from a set of sentences of a given document to a subset of the sentences that satisfies the three requirements(diversity, sufficient coverage and balance) .The mapping is learned by incorporating several constraints in a structure learning framework, and we explore the graph structure of the output variables and employ structural SVM for solving the resulted optimization problem.

Vishal Gupta (2010) the importance of sentences is decided based on statistical and linguistic features of sentences.

Hui Lin, Jeff Bilmes(2011) design a class of submodular functions meant for document summarization tasks. argue that monotone nondecreasing submodular functions F are an ideal class of functions to investigate for document summarization. in fact,to submodular function optimization, adding further evidence that submodular functions are a natural fit for document summarization.

Róbert Móro, Mária Bieliková (2012) propose a method of personalized text summarization which improves the conventional automatic text summarization methods by taking into account the differences in readers' characteristics. they use annotations added by readers as one of the sources of personalization.

Ha Nguyen, Thi Thu (2013)present a Vietnamese text summarization method based on sentence extraction approach using neural network for learning combine reducing dimensional features to overcome the cost when building term sets and reduce the computational complexity.

2-1-7-2 Multi-Document Summarization

This is where one summary needs to be composed from many documents. extract a summary for multi-documents was good field for Researchers in mid 1990s.

Multi-document summarization differs from single in that the issues of compression, speed, redundancy and passage selection are critical in the formation of useful summaries.

Inderjeet Mani, Eric Bloedorn(1997) A particular challenge for text summarization is to be able to summarize the similarities and differences in information content among these documents in a way that is sensitive to the needs of the user.

Maniand Maybury(1999) Key factors for make summaries are: putting the important concept in each document, reducing the size of each document and comparing ideas across documents.

2-1-8 Ant Colony Optimization Algorithm

Ant colonies, and more generally social insect societies, are distributed systems that, although of the simplicity of their individuals, give a highly structured social community. As a result of this community, ant colonies can traverse complicated tasks that in some cases far exceed the individual capabilities of a single ant. The field of ‘‘ant algorithms’’ studies models derived from the observation of real Ants’ behavior, and uses these models as a source of inspiration for the design of Brilliant algorithms for the solution of optimization and distributed control problems. The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems. Several different aspects of the behavior of ant colonies have inspired different kinds of ant algorithms. This type of algorithms is known as Ant Colony Optimization, whose

first member, called the Ant System, was initially proposed by Marco Dorigo (Dorigo, Maniezzo & Colorni, 1991).

The ACO algorithm, which emulate the foraging behavior of real life ants, is a mutual population-based search algorithm. While traveling, Ants put an amount of pheromone (a chemical substance). When other ants find pheromone way, they decide to follow the way with more pheromone, and while following a specific way, their own pheromone reinforces the way followed. Therefore, the continuous put of pheromone on a way maximizes the probability of selecting that way by next ants. Moreover, ants use short paths to the food source, return to the nest sooner and therefore, quickly mark their paths twice, before other ants return. As more ants complete shorter paths, pheromone accumulates faster on shorter paths; on the other hand, longer paths are less reinforced. Pheromone evaporation is a process of decreasing the intensities of pheromone on the way over time. This process is used to avoid local convergence (old pheromone strong influence is avoided to prevent premature solution stagnation), to explore more search space and to decrease the probability of using longer paths.

2-1-8-1 Pseudocode for the ACO algorithm
is shown below:

```
Initialize the pheromone trails and other parameters, and
calculate heuristic information;
While termination condition not met do
    ConstructAntSolutions
    ApplyLocalSearch (optional)
    UpdatePheromones
endwhile.
```

After initialization, the ACO algorithm iterates over three main stage: at each repetition, a number of solutions are constructed by the ants; these solutions are then change for the better over a local search (this step is optional), and finally the pheromone is updated.

Construct Ant Solutions: A group of ants build solutions from elements of a finite set of available solution components. A solution construction phase starts from an empty partial solution. At each construction step, each ant extends its partial solution by adding a feasible solution component from a set of neighbor components that can be added to its current partial solution. The choice of a solution component is guided by a stochastic mechanism, which is biased by the pheromone associated with each of the elements of the set of components that can be added to the current partial solution.

The ant select next **i** city by calculating probabilities:

$$P\left(c_{ij}\,\middle|\,s^P\right) = \frac{\tau_{ij}^{\alpha} * \eta_{ij}^{\beta}}{\sum_{c_{ij} \in N(s^P)} \tau_{ij}^{\alpha} * \eta_{ij}^{\beta}} \quad , \forall c_{ij} \in N(s^P) \tag{2.3}$$

Here

- $\mathbf{s^P}$ - partial solution #p.
- **N** -set of all paths from the location **i** to all adjacent locations still not visited by the ant.
- $\boldsymbol{c_{ij}}$ - path from the location **i** to the location **j.**
- **P** - probability.
- $\boldsymbol{\tau_{ij}}$ -amount of pheromone on the path $\boldsymbol{c_{ij}}$**.**
- $\boldsymbol{\eta_{ij}}$ -some heuristic factor.

ApplyLocalSearch: After constructing solutions, and before updating the pheromone trails, it is common to improve the solutions obtained by the ants through a local search. The ApplyLocalSearchphase, which is highly problem-specific, is optional.

UpdatePheromones: The Update Pheromones stage aims to (i) increase the pheromone values associated with promising solutions,and (ii) to decrease those that are associated with bad solutions. Usually, Update Pheromones stage is achieved (i) by decreasing all the pheromone values associated with all solutions through pheromone evaporation, and(ii) by increasing the pheromone values associated with a chosen set of promising solutions.

2-1-8-2 Type of ACO Algorithms

The ACO algorithm has many type as listed below (Dorigo, Maniezzo & Colorni, 1996; Dorigo, & Stützle, 2004):

- The Ant System (AS) is the first ACO algorithm , it has been applied to solve travelling salesman problem (TSP).

  Three different type of AS have been reported:

  - Ant-density (Dorigo, Maniezzo & Colorni, 1996).
  - Ant-quantity (Dorigo, Maniezzo & Colorni, 1996).
  - Ant-cycle (Dorigo, Maniezzo & Colorni, 1996)

  In ant-density and ant-quantity, ants only deposit pheromone directly after crossing An arc. In ant-quantity, the amount of pheromone is inversely proportional to the length of the arc crossed, whereas in ant-density a constant amount of pheromone per unit distance is deposited. In ant-cycle the ant are only allowed to deposit pheromone when they completed a Round.

- Elitist strategy for ant system (ASe) (Dorigo, Maniezzo &Colorni, 1996), the primary idea of this algorithm is to increase the importance of the ant that found the best solution.
- Ant Colony System (ACS) (Gambardella & Dorigo, 1996) and Ant-Q (Gambardella & Dorigo, 1995): the main difference between them is the definition of the pheromone trail formula.
- The Rank-Based Version of the Ant System (Bullnheimer, Hartl & Strauss, 1999): in this algorithm, each ant deposits an amount of pheromone that decreases with its rank.
- The Max-Min Ant System (Stützle & Hoos, 1996; Stützle & Hoos, 1998): this algorithm goal to exploit more strongly the best solutions found during the search process and to direct the ants' search towards very high quality solutions. On the other hand, Max-Min Ant System objective to avoid premature stagnation of the ants' search.

2-1-8-1 Applications of Ant Colony Optimization

Applications of ACO algorithms listed according to problem types and chronologically

Table 2-1 Applications of Ant Colony Optimization

Problem type	Problem name	Main references
Routing	Traveling salesman	Dorigo, Maniezzo, & Colorni (1991a,b, 1996) Stu¨tzle & Hoos (1997, 2000) Cordo´n, de Viana, Herrera, & Morena (2000)

	Vehicle routing	Bullnheimer, Hartl, & Strauss (1999a,b) Gambardella, Taillard, & Agazzi (1999)
	Sequential ordering	Gambardella & Dorigo (1997, 2000)
Assignment	Quadratic assignment	Maniezzo, Colorni, & Dorigo (1994) Stu¨tzle (1997b) Maniezzo (1999) Stu¨tzle & Hoos (2000)
	Graph coloring	Costa & Hertz (1997)
	Generalized assignment	Lourenc¸o&Serra (1998, 2002)
	Timetabling	Socha, Sampels, & Manfrin (2003)
Subset	Multiple knapsack	Leguizamo´n&Michalewicz (1999)
	Max independent set	Leguizamo´n&Michalewicz (2000)
	Weight constrained graph tree partition	Cordone & Maffioli (2001)
	Arc-weightedl-cardinality Tree	Blum & Blesa (2003)
Machine learning	Classification rules	Parpinelli, Lopes, & Freitas (2002b)
	Bayesian networks	de Campos, Ga´mez, & Puerta (2002b)
	Fuzzy systems	Casillas, Cordo´n, & Herrera (2000)
Network routing Network routing	Connection-oriented network routing	Schoonderwoerd, Holland, Bruten, & Rothkrantz (1996)

2-1-9 Evaluation Measure

The most import challenge for automatic text summarization systems. The Evaluation makes a summary useful. There are at least two categories of the summary that must be measured when evaluating summaries and summarization systems (Jing et al., 1998; Mani and Maybury, 1999; Afantenos et al., 2005):

2-1-9-1 Intrinsic Evaluation

Intrinsic evaluation measures the system in of itself. This is mostly done by similarity between gold standard, which can be made by a reference summarization system or, more often than not, is man-made using informants. Intrinsic evaluation has mainly concentrate on the coherence and informativeness of summaries.

- Summary Coherence: Summaries generated through extraction-based methods (cut-and-paste operations on phrase, sentence or paragraph level) sometimes suffer from parts of the summary being extracted out of context, resulting in coherence problem example(gaps in the rhetorical structure of the summary).
- Summary Informativeness: to compare the generated summary with a reference summary, measuring how much information in the reference summary is present in the generated summary.
- Sentence Precision and Recall: Often text summarization systems based on extraction approaches, the most significant sentences in the source text are selected together into a summary without changing any of their original text. In such settings, the commonly used information retrieval metrics of precision and recall can be used. Recall is the fraction of sentences selected by a human that were also correctly obtained by the system while precision is the fraction of system sentences that were

correct. F-measure or balanced F-score combines Precision, and Recall as the weighted harmonic mean of precision and recall.

$$\boldsymbol{Recall} = \frac{|system\ summaries| \cap |human\ summaries|}{|human\ summaries|} \quad (2.4)$$

$$\boldsymbol{Precision} = \frac{|system\ summaries| \cap |human\ summaries|}{|system\ summaries|} \quad (2.5)$$

$$\boldsymbol{F} = \frac{2(Recall * Precision)}{Precisiom\ +\ Recall} \quad (2.6)$$

Demand for precision and recall as evaluation measure is apparent, it can be Frequently used to evaluate, where subsequent to the gold-standard sentence selection defined by human.

2-1-9 -2 Extrinsic Evaluation

Measures the efficiency and acceptability of the produced summaries in some job, for example importance assessment or reading comprehension. Other possible measurable tasks are information gathering in a large document collection, the effort and time required to post-edit the machine generated summary for some specific purpose, or the summarization system's impact on a system of which it is part of, for example relevance feedback (query expansion) in a search engine or a question answering system.

2-1-9-3 Evaluation Measure Tools:

(ROUGE) Recall-Oriented Understudy for Gisting Evaluation:it provided by Lin (2004) is the most convenient evaluation tool used in the Document

Understanding Conference (DUC) text summarization evaluations. ROUGE includes measures to compute and determine the characteristic of a summary by comparing system generated summaries (system summary) with humans generated summaries (human summary). The measures count the number of overlapping units such as n-gram, word sequences between the system generated summaries to be evaluated and compared with summaries generated by humans. ROUGE package contains various automatic evaluation methods such as the following.

- ROUGE-N: N-gram Co-Occurrence Statistics: an n-gram recall (shared n-grams) between system summary and a set of summaries generated by human. ROUGE-N is computed as follows:

$$\frac{\sum_{S\in\{referencrsummaries\}}\sum_{gram_n\in S}^{\text{Count }_{match}(gram_n)}}{\sum_{S\in\{referencrsummaries\}}\sum_{gram_n\in S}^{\text{Count }_{match}(gram_n)}} \quad (2.7)$$

Where n is the length of the n-gram ($gram_n$) $\text{Count }_{match}(gram_n)$ is the most possible number of n-grams shared between a systems generated summary and a set of reference summaries.

2-2 Related Work

the ant colony optimization algorithm (ACO) (Marco Dorigo 1992) has the ability to perform such a role and to learn the most optimized features weights. To the best of our knowledge, a Ant Colony Optimization (ACO) has not been used for text summarization previously. however, it has been employed in related fields such as document clustering(Łukasz Machnik 2005), Image Feature Selection(Ling Chen and Bolun Chen and Yixin Chen) and Spam Host Detection (Arnon Rungsawang,Apichat Taweesiriwate, Bundit Manaskasemsak, 2012).The ant colony optimization (ACO) as a machine learning algorithm will raise the

problem of supervised machine learning approaches. The following are the reasons of why the ACO was chosen to solve the problem of automatic text summarization. A recent work published reported that the ACO algorithm has become quite popular in the machine intelligence and cybernetics communities. It has successfully been applied to different domains of science and engineering, such as mechanical engineering design signal processing and machine intelligence Naturally, the proposed methodologies are exposed to advantages and disadvantages. Although the optimization techniques are used to overcome some limitations of other proposed methods, they suffer many defects. (Jun et al., 2011) surveyed some evolutionary computing algorithms (ECAs). The survey discussed how the ECAs search performance could be optimized using machine learning techniques. This trend of research direction treated the term "Machine Learning for Evolutionary Computing (MLEC)" for the discussed purpose. The ECAs agreed in a general structure which includes the following stages: population initialization, fitness evaluation and selection, population reproduction and variation, algorithm adaptation, and local search. The survey viewed the algorithm defects and the successful solutions.

This study will also propose a method that generates summaries by selecting sentences based on sentences score using important features and adjusting the weighting for each of them to improve the total results. The key point of this researches based on automatic summary extraction using ACO to extract key sentences of the original text as its summary by estimating the relevance of sentences through capturing the main content.

2-2-1 Research Group

This group consists of four members; Dr. Albaraa Abuobieda[1] is the leader of the group. (Asem Abdulla, Abdelrahman Yousif and Omer Fisal)[2] are members of the group. In previous work has been the comparison between GA (Suanmali *et al*., 2011), (Albaraa Abuobieda *et al*., 2013) and (Binwahlan et al., 2009), this comparison was unfair because the use different features. (Asem Abdullah and Albaraa Abuobieda, 2014) are applied GA to extract features weights. (Omer Fisal and Albaraa Abuobieda, 2014) used binary ACO algorithm as new method to extractive features weights. This research applied ACO algorithm to extract features weights. These works are compared with DE algorithm (Albaraa Abuobieda *et al*., 2013). These group is used the same features.

[1] Dr.Albaraa Abuobieda Mohammed Ali he received his PhD from Univeristi Teknologi Malaysia in the area of Text Summarization. He received his B.Sc in Computer Science from the International University of Africa, Sudan, in 2004. He earned M.Sc in Computer Science from Sudan University of Science and Technology in 2008. His current areas of research include text summarization, plagiarism detection, Ontology, network and network security. Currently he is a dean of the Faculty of Computer Studies - International University of Africa.

[2] Asem Abdulla, Abdelrahman Yousif and Omer Fisal are master students.

2-3 Summary

This chapter look at text summarization idea, approaches, type, automatic text summarization system, and deploy of summarization applications. This chapter reviewed main tow techniques: the first techniques single document include early work, machine learning based approaches, Nive-Ba6yes methods. The second techniques multi-document summarization. This chapter gave a brief about Ant colony optimization algorithm. This chapter reviewed evaluation measures that used in summarization. At least reviewed the related work.

CHAPTER 3

METHODOLOGY

3-1 Introduction

This chapter discusses the overall methodology proposed in this research to design and develop text summarization model.

3-2 Research Design

This study will try new techniques ant colony optimization in order to design a novel summarization model. This study has main experiment: ant colony optimization based text summarization method.

3-3Operational Framework

Operational framework provides milestone to lead the reader through the well-organized documentation of the methods used. The operational framework of this research depend of five phases, Phase 1: Basic Elements; Phase 2 Ant Colony Optimization Algorithm Feature Subset Selection; Phase 3 Training procedure; Phase 4: Testing procedure.

3.3.1 Phase 1: Basic Elements

This phase are cares with gathering the standard data set (test bed) for evaluating the methods. The Document Understanding Conference (DUC) turn into the essentially test bed used for evaluating automatic text summarization research. The DUC datasets are aggregations of well-known newswire articles written by professional language experts. Subsection 3.3.1.2 states the pre-processing steps that are wanted for preparation where the articles are processable. The feature scoring process is a foundation in processing a document for automatic text summarization. Subsection 3.3.1.3 describes how to compute a set of selected features over all preprocessed documents in the data set. All basic elements belong

to this phase are shared between the proposed methodologies and are support on all coming phases designed in this research.

3.3.1.1 Collecting the DUC2002 Data set

The National Institute of Standards and Technology of the U.S. (NIST) created the DUC 2002 evaluation data which contains of 60 datasets. The following ten documents D075b, D077b, D078, D082, D087, D089, D090, D092c, D095c, and D096c comprising of one hundred documents are used. See Appendix A for more details. The data was designed from the Text REtrieval Conference (TREC) disks used in the Question-Answering task of TREC-9. The DUC 2002 dataset came with two tasks single-document Extract/abstract and multi-document extract/abstract with different topics such as natural disaster and health issues. Each document in DUC2002 collection is supplied with a set of human generation summaries provided by two different experts (see appendix B, C and d). For evaluating the experimental parts of this research, one of the Human-made summaries is assigned as a reference summary and called (H1); the other named (H2) use as a benchmark method. The H2 method is compared to H1, named H2-H1. This comparison is established to easily estimate how close the performances of the proposed methods are against human performance.

3.3.1.2 Text Data Preprocessing

The text preprocessing is a necessary step since the quality of produced summaries on accurate text preprocessing and representation. In this stage, there are four main tasks performed: sentence segmentation, tokenization, stop words removal, and word stemming.

- Sentence Segmentation: Sentence segmentation as a natural-language processing task is not only used to locate each sentence in a separate line within the document, it also defines the challenge of detecting sentence

boundary. There are several notation marks that share the characteristic of sentence end point such as ".", "?", and "!".

- Tokenization: is separating the input document into individual words. There are many notation marks used to discover the boundary of tokens such as the tab, white space, period, colon, semicolon, and comma and so on.
- Stop Words Removal: Stop words are words that not share text relevance retrieval. A different number of stop words lists are available online that are found to increase the precision of information retrieval (IR) systems, see Appendix E for more stopwords examples. In this research, the 571stop-words list of Cornell University project was used (Salton and Buckley, 1988).
- Words Stemming: returning each word to its root form. In this research the Porter's stemmer algorithm (Porter, 1997) is used to remove all affixes (prefixes and suffixes) of the words. The Porters' stemmer is a famous algorithm used in IR research.

3.3.1.3 The Selected Features

Many type of text summarization's features have been offered to extract

Salient sentences from the text. In this study, five statistically rich features are selected to score each sentence in the document. Since the feature scoring process is a foundation of the summary sentence selection approach, all phases are suggested to run this process. The features are: Title-Feature "TF" and Sentence-Position "SP"(Edmundson, 1969), Sentence-Length "SL" (Nobata et al., 2001), Numerical-Data "ND" (Fattah and Ren, 2009), and Thematic-Word "TW" (Luhn, 1958, Edmundson,1969, Luo et al., 2010).

- Title Feature (TF): a sentence containing each of the "Title" words is considered an important and topic-related sentence. Title feature T F can

be calculated using Equation (3.1), where CountWord ( ) is a function used to count words of the input parameter such as the i^{th} sentence in the document $S_i$that are intersected with the Title words; CountLength () is a function computes the length of the title based on the number of words enclosed.

$$TF(S_i) = \frac{\text{CountWord } (S_i) \cap \text{CountWord (Title)}}{CountLength\ (Title)} \quad (3.1)$$

- Sentence Length (SL): In order to prevent selecting sentences that are either too short or too long, a normalized division may solve the issue. Equation (3.2) shows such a normalization, where S_i refers to the i^{th} sentence in the document consists of n words, S_j refers to the longest sentence in the document consists of m words, and CountLength () is a function computes the length of each input sentence based on the number of words found.

$$SL(S_i) = \frac{CountLength\ (S_{(i,w\in\{1\ldots n\})})}{CountLength\ (S_{(i,w\in\{1\ldots m\})})} \quad (3.2)$$

- Sentence Position (SP): The first sentence in the paragraph is a valuable sentence and a good candidate for add in the summary. Equation (3.3) is used to calculate the SP feature, where S_i refers to the i^{th} sentence in the document wanted to extract its position score, and CountTotal ( ) is a function that retrieves the total number of the sentences in the input parameter document d and CurrentPosition () is a function that retrieves the current order of sentence S_i in document d.

$$SP\ (S_i) = \frac{CountT\ otal\ (d) - CurrentP\ osition\ (S_i)}{CountTotal(d)} \quad (3.3)$$

- Numerical Data (ND): A sentence that contains numerical data often have important information such as a date of event, money transaction, damage percentage, and etc. Equation (3.4) shows how to compute this

feature where CountND () is a function computes the Numerical Data (ND) found in the i^{th} sentence S in the document, and CountLength ( ) is a function used to compute the sentence length of S_i.

$$ND(S_i) = \frac{CountND(S_i)}{CountLength\ (S_{(i,w\in\{1..n\})})} \quad (3.4)$$

- Thematic Words (TW): are a list of top n selected terms with the highest frequencies. To calculate the thematic words, first the frequencies of all terms in the document are computed. Then, a threshold is specified in order to assign terms that should be selected as thematic words. In this case, the top ten frequent-terms max(TW) would be assigned as a threshold. To compute the ratio of T W found in the i^{th} sentence S in the document Equation (3.5) is used where CountThematic() is a function used to compute the number of the thematic words found in Sentence S_i.

$$TW(S_i) = \frac{CountT\ hematic\ (S_i)}{\max(TW)} \quad (3.5)$$

3.3.1.4 Adjusting the ACO Algorithm Parameters

Marco Dorigo (inventor of Ant Colony Optimization) provides a very useful discussion of the problem parameters in his article "The Ant System: Optimization by a Colony of Cooperating Agents" (Dorigo et al, 1996).

- Alpha (α)/Beta (β): A number of α/β combinations were found to yield good solutions in a reasonable amount of time. These are found in Table 3.1.

Table 3.1 Alpha (α)/Beta (β)

Alpha (α)	Beta (β)
0.5	5.0
1.0	1.0
1.0	2.0
1.0	5.0

- The α parameter is associated with pheromone levels (from Equation 2.3), where the β parameter is associated with visibility (distance for the edge). Therefore, whichever value is higher indicates the importance of the parameter within the edge selection probability equation.
- Rho (ρ): Recall that while ρ represents the coefficient applied to new pheromone on a path, (1.0 - ρ) represents the coefficient of evaporation of existing pheromone on the trail. Tests were run with $\rho > 0.5$, all of which yielded interesting solutions. Setting ρ to a value less than 0.5 resulted in less than satisfactory results. This parameter primarily determines the concentration of pheromone that will remain on the edges over time.
- Number of Ants: The quantity of ants in the simulation had an effect on the quality of solutions that resulted. While more ants may sound like a reasonable idea, setting the number of ants in the simulation to the number of cities yields the best result.

3.3.2 Phase 2 Ant Colony Optimization Algorithm Feature Subset Selection

ACO algorithms have been used to several time for many problems, just like as shown by the table 2-1. Based on Those applications, it is easy to identify the basic issues that play important roles in the use of ACO in any combinational problem. These basic issues are the following (Dorigo & Stützle, 2004):

- Construction Graph: The application problem must be presented as a graph with a set of nodes and edges between nodes.
- Pheromone Trails Definition: A very important choice when applying ACO is the definition of the meaning of the pheromone trails update. It is important to definition of pheromone trails is crucial and a poor choice will

result in poor ACO performance. Typical methods involve selecting a number of best solutions (ants) and updating the edges they chose.

- Balancing Exploration and Exploitation: An effective metaheuristic algorithm must achieve an appropriate balance between exploitation of the search experience gathered so far and the exploration of the unvisited search space. One simple approach to exploiting the ant's search experience is to define the pheromone trail updating as a function of the solution (ant) quality. Another simple approach in the balance of exploration and exploitation is tuning α and β, where α determines the influence of the pheromone trail and β determines the effect of heuristic information.
- Heuristic Information: Using problem related knowledge as heuristic information to direct the ants' probabilistic solution construction is an important factor to achieve better quality solutions. The main types of heuristic information are static and dynamic.
- ACO and Local Search: When coupled with local search algorithms, ACO algorithms perform best in many applications to NP-hard optimization problems.
- Candidate Feature Lists: If ants have a large number of possible moves from which to choose, then the computational complexity increases.
- Number of Ants: Generally speaking, although a single ant may generate a solution, the number of ants should be greater than one, and most of the time, the number of ants is set experimentally (Dorigo, Maniezzo & Colorni, 1996).

3.3.2.1 ACO Based Feature Subset Selection Steps(FSS):

1. The FSS search space (features) is represented by a weighted graph (nodes with edges connecting them), where the nodes represent features and the edges denote the choice to select the next features. An optimal subset of

features can be searched by an ant that traversed through the graph where a predefined number of nodes (features) are visited (selected) that satisfy a traverse stopping criterion.

2. The pheromone trail updating is defined to lay an amount of pheromone proportional to the quality of the best solutions achieved.
3. The heuristic information is defined by $\eta_{ij} = \frac{1}{d_{ij}}$ that is, the heuristic desirability of going from vertex i directly to vertex j is inversely proportional to the distance between the two vertexes.
4. α and β are used to balance exploration and exploitation.

3.3.2.2 Proposed ACO Based Feature Subset Selection Steps

The FSS process may be reformatted into an ant colony optimization suitable problem. Figure 6.1 illustrates this scenario, the ant is currently at node"a", and has a choice of which feature to select next to its path (paths are represented by dotted lines). The ant chooses feature "b", this section is based on some "probabilistic feature selection" value, then the ant chooses feature "c" and then feature "d". Upon arrival at feature "d", the ant terminates its traversal and outputs the current subset of features (a, b, c, and d). A suitable heuristic desirability of travelling between features could be any feature subset evaluation function.

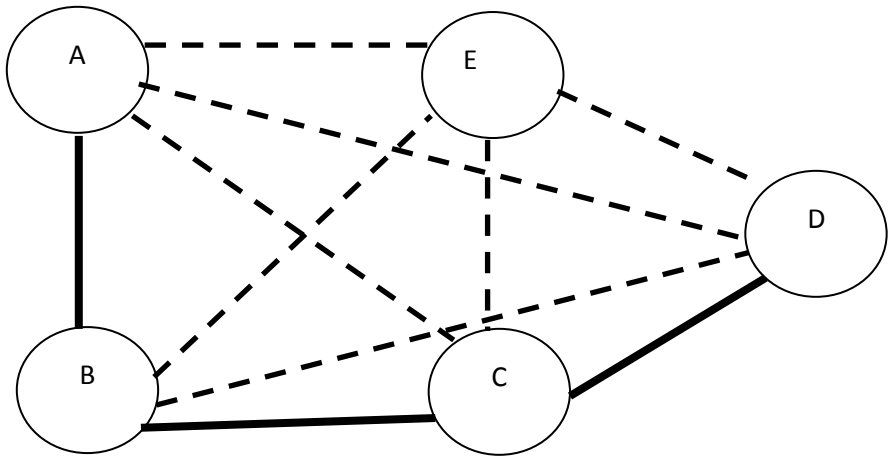

Figure 6.1 ACO Representation for Feature Subset Selection

The heuristic information and the pheromone levels associated with features are combined to form a probabilistic transition rule $P_{ij}^{k}(t)$,denoting the probability of an ant at feature i choosing to select feature j at time t :

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k}[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}} & if\ j \in J_i^k \\ 0 \qquad otherwise \end{cases} \qquad 3.6$$

Where:

- k is the number of ants.
- When an ant at feature i, η_{ij} is the heuristic information to select feature j.
- $\tau_{ij}(t)$ Is the amount of pheromone associated with the edge between feature i and feature j.
- $J_i^k$is the set of neighbor features of feature i which have not yet been visited by the k 's ant.
- α > and 0 β > are two tuning parameters to determine the relative importance of the pheromone levels and the heuristic information.

The ACO-Based FSS method starts by generating a number of ants, each ant selects one random feature. From these initial positions, each ant traverses edges probabilistically The ACO-Based FSS method starts by generating a number of ants, each ant selects one random feature. From these initial positions, each ant traverses edges probabilistically satisfied. The resulting subsets of features are evaluated by best summation of subset of features. If

an optimal subset is found or the algorithm has executed a certain number of times, then the feature subset selection process halts and outputs the best subset of features encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the feature subset selection process iterates once more.

The pheromone trails are updated according to our proposed Automatic Text Summarization Pheromone Update Formula(ATSPUF), the ATSPUF is defined as follows:

$$\tau_i = \begin{cases} \rho.\tau_i + \Delta\tau_i + \omega.\Delta\tau_i & f_i \in EBS_j \\ \rho.\tau_i + \Delta\tau_i & otherwise \end{cases} \qquad 3.7$$

Where:

- τ_i Is the pheromone level associated with each feature.
- ρ is a coefficient such that(1- ρ) represents the evaporation of the pheromone level. $0 \leq \rho < 1$, ρ must be set to a value 1 < to avoid unlimited accumulation of trails (ρ is determined experimentally).
- Elitist Best Solution (EBS_j) is any solution S_j(any subset of features) among the top best solutions. In other words, any high quality solution (any subset of features) whose performance effectiveness is better than a predefined effectiveness value is considered as a member of EBS_j As a result, extra pheromone values shall be awarded to any solution that belongs toEBS_j. This extra pheromone value is defined by $\omega.\Delta\tau_i$ where ω is the performance effectiveness of the corresponding Solution S_i .
- f_i be a feature indexed by i (feature i).
- $\Delta\tau_i$ (amount of pheromone change for each feature) is defined by:

3.7

$$\Delta\tau_i = \begin{cases} \dfrac{max_{g1:TBS}\ \left(F_{1g}\right) - F_{1j}}{max_{h=1:TBS}(\ max_{g1:TBS}\ \left(F_{1g}\right) - F_{1j})} & f_i \in TBS_j \\ 0 & otherwise \end{cases}$$

- $f_i \in TBS_j$ Means that only the top best solutions (TBS) are used to calculate $\Delta\tau_i$ values.

Below are the main steps of our proposed ACO-Based FSS algorithm:

- Step 1 - Initialization: Initially, the ant colony algorithm parameters are initialized:
  - The amount of pheromone change for each feature $\Delta\tau_i$ is set to zero where i is a feature index, i ∈ [0, N] , and N is the total number of features in the feature space).
  - The pheromone level associated with each feature is initialized to some constant value τ_i=1.
  - Define the number of solutions (number of ants - NAs).
  - Define the maximum number of iterations.
- Step 2 – generation ants for initial iteration: For each solution (ant) ($ant_i : i = 1: NAs$)(in our experiments, we have five features).
- Step 3 – Evaluation solutions: for each solution ($ant_i : i = 1: NAs$) evaluate selects features in solution i.
- Step 4 – Stopping criterion: **If** a predefined stopping criterion is met(1000iterations)
  - Step 5 – Then:
    1. Stop the Ant Colony Optimization-Based Feature Subset Selection algorithm.
    2. Return the best subset of features.

- Step 6 – Else
    1. Update the pheromone levels associated with all features. Pheromone update is based on our proposed Automatic Text Summarization Pheromone Update Formula (ATSPUF).
    2. Select new features for the NAs ants for the next iteration.
    3. Go to evaluation Step 3.

3.3.3 Phase 3 Training procedure

In training level we used DUC2002 data set (70 documents from training). Each document start by preprocessing process (sentence segmentation, tokenization, stop word removal and stemming), then extracting the text features. The score of each sentence features are present a vector see table 3.2. The resulting of the features scores are used as input for PSO scoring function Equation 3.8

$$score(s_i) = \sum_{j=0}^{5} s(f_i) * \ vopp(i) \qquad 3.8$$

Where $score(s_i)$ is the score of the sentence s_i, $s(f_i)$ is the score of the j^{th} feature and vopp(i) is the value of i^{th} bit in ACO .All document sentences are scored using Eq. 3.8 and ranked in descending order. Then the top n sentences are selected as summary, where n is equal to the predefined summary length. In this study, 20% of the total number of the document sentences is used as summary length. The created summary is used as input for the fitness function. The ROUGE-1 is used as the fitness function (see chapter 2, subsection 2-1-9-3) for more details. Depend on

the summary evaluation Based on the summary evaluation, the best path of the ACO is determined, which means the evaluation value of the best summary generated by that ACO which is the evaluation value of the best summary created by a particle in the population so far. By the end of iteration, the ACO with the best path value is selected as a vector for the best selected features of each document. The final features weights are calculated over the vectors of the features weight of all documents in the data collection.

Table 3.2 Representative for Feature Score vectors

Feature / Sentence	TF	SL	SP	ND	TW	Total
S_1						
S_2						
S_3						
...						
S_i						

3.3.4 Phase 4: Testing procedure

The target of deploy the ACO to discover and optimize the corresponding weight w_j of each feature. Equation 3.8 is calculate the features weights

$$score\ weights(s_i) = \sum_{j=1}^{5} wj * score(f_j(s_i) \quad 3.8$$

Where $score\ weights(s_i)$ is the score of sentence s, wj is the weight of the feature j that produced by ACO, j is the number of feature and score_fj(si) is a function that calculate the score of the feature j. The training procedure used 100 documents from DUC2002 data set. The training procedure is begin with input document, then implementing the preprocessing process (segmentation,

tokenization, remove stop word and stem the word), then extracting features for each sentence, then modify the score of each feature based on the features weights that produced in training process, then calculate the score of each sentence in document by Equation 2.7, then order the sentence based on their score in descending order, then select top *n* sentence as summary sentence, where *n* is equal to predefined summary length, then order the summary sentences in the same order as in the original document. The ROUGE package (lin. 2004) is used as evaluation measure.

3.3.5 The Selected Methods for Comparison

To standardize evaluation of the proposed methods in this research, benchmark and similar methods are selected to demonstrate significant comparison of results. Those methods have been divided into two parts: benchmark methods and state-of-the-art methods as follows.

3.3.5.1The Benchmark Methods

The selected methods are standard benchmark methods that have been widely used (Ren Arnulfo Garca-Hernndez, 2009). They are chosen for comparison purposes.

- Binary Differential Evolution Based Text Summarization (BiDETS) Model (Albaraa Abuobieda, 2012).
- H2-H1: This method was stated before in Section 3.3.1.1.

3.3.5.2The State-of-the-art Methods

The second part includes state-of-the-art methods selected to show the performance rate of proposed methods in this study as compared the other proposed methods. The proposed method is related to method(s) from the literature review it/they will be tested and compared based on its/their ROUGE evaluation measures.

- Feature Selection methods: PSO
- Feature Selection methods: GA

Selected Benchmarks and Similar Methods

3.3.6 Selected Benchmarks and Similar Methods

According to Section 3.3.5 (BiDETS), (H2-H1), and (PSO and GA) should be added for purposes of comparison. The composed of similar published optimization based text summarization methods: Genetic Algorithm (GA) (Albsharee., 2014) and Particle Swarm Optimization methods (PSO)(Abd rhman ,2014) thos are team work algorithms. To the best of this author's knowledge, the ACO has never been presented before to tackle text summarization features weighting problem. In the literature, the ACO has been proposed as a clustering approach for text summarization problems (Al-Ani,2005) but not for feature weighting problems. However, both GA and PSO have been employed for the problem of weighting features in text summarization.

CHAPTER 4

RESULTS AND DISCUSSION

4-1 Introduction

This chapter describes the proposed model testing results are presented accompanied with a comparison between the (particle swarm optimization algorithm, Genetic algorithm) Binary Differential Evolution Based Text Summarization (BiDETS) for text summarization used same five statistical features

4-2 Results

Table 4.1 an algorithm comparison using ROUGE-1 result

algorithm	Avg-R	Avg-P	Avg-F
H2-H1	**0.51642**	**0.51656**	**0.51627**
BiDETS	0.45610	0.52971	0.48495
ACO	0.3105	0.4508	0.3289
PSO	0.2871	0.4101	0.3011
GA	0.2918	0.3362	0.2782

Table 4.2: An algorithm comparison using ROUGE-2 result

algorithm	Avg-R	Avg-P	Avg-F
BiDETS	**0.24026**	**0.28416**	**0.25688**
H2-H1	0.23394	0.23417	0.23395
ACO	0.1422	0.2318	0.1589
PSO	0.1023	0.1317	0.1017
GA	0.1182	0.1493	0.1173

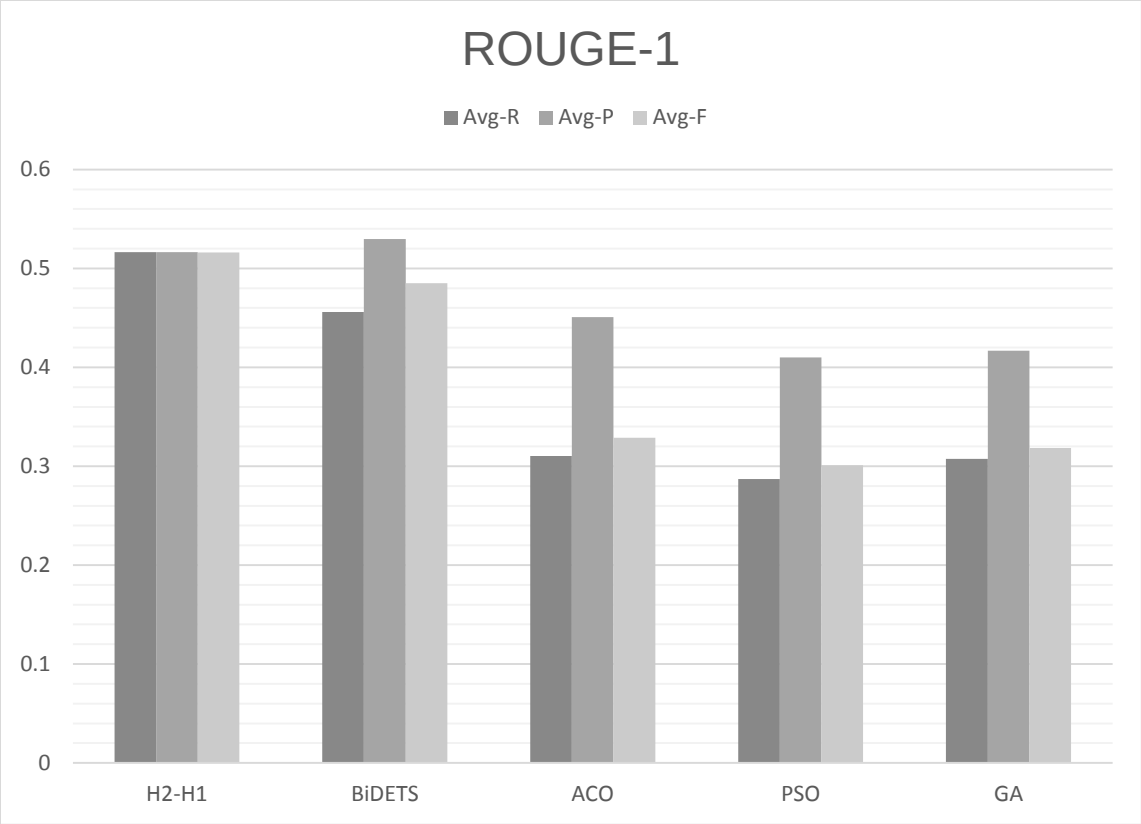

Figure 4.1: The comparison of average precision, recall and f-measure scores of Different summarizer using ROUGE-1 result.

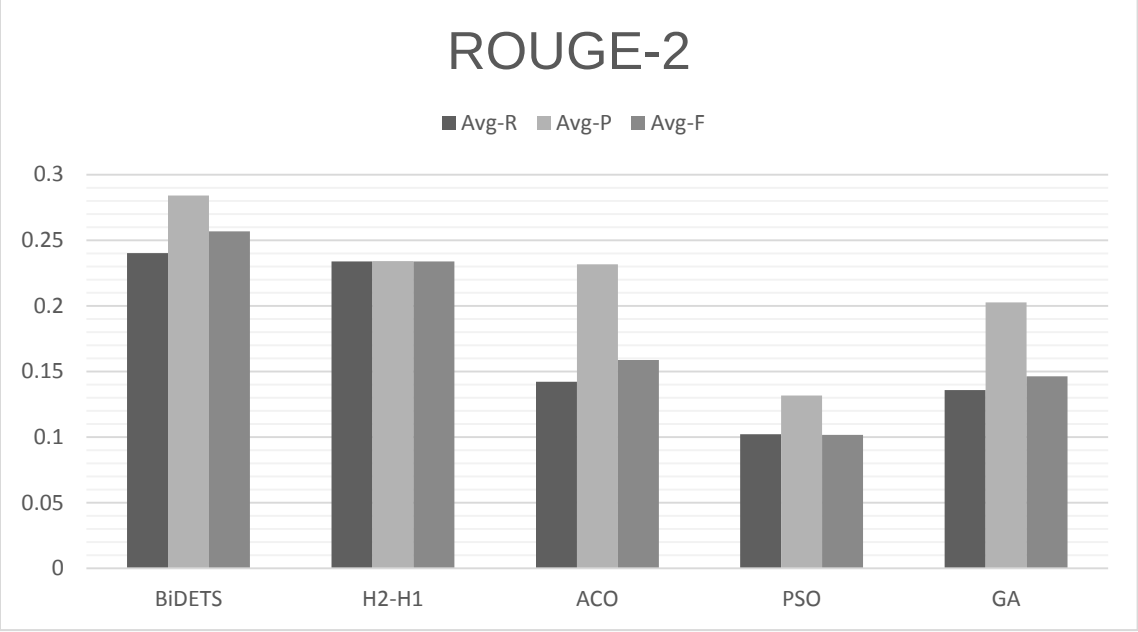

Figure 4.2: The comparison of average precision, recall and f-measure scores of Different summarizer using ROUGE-2 result.

4-3 Discussion

The experimental results showed good performance for the proposed model compared with algorithms (PSO, GE).we used ROUGE-N evaluation measure. The results of all summarizer were compared with human generated summaries using ROUGE-1. The compression rate is defined for summarizers as 20%. when comparing with the benchmark methods used in this study (BiDETS) still have the not bad result in DUC 2002 system and a human summarizer against another human summarizer (H2-H1).Table 4.1 and Table 4.2 shows the comparison of the average recall ,precision and f-measure score between (Binary Differential Evolution Based Text Summarization, Ant Colony System, particle swarm optimization and Genetic) using ROUGE-1 and ROUGE-2 . Finally the Figures 4.1, 4.2 visualize the same results of comparison of ROUGE-1 and ROUGE-2 .

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1Introduction

This chapter reviews the purpose method in this research. The Ant Colony algorithm was extensively investigated for the text summarization problem. A proposed method was investigated using a standard dataset used for text summarization; i.e. DUC 2002. The DUC 2002 was used since it was produced for single document summarization. In addition, the standard evaluation toolkit (ROUGE) was used during this study find out the significance of test results.

5.2 The Proposed Methods (Study Contributions)

The proposed methods for text summarization research are used primarily for addressing the problem of generating high quality text summaries. This can be Realized by knowing very important sentences for selection that best represent the document. The main philosophy of this research is to investigate a single algorithm (ACO) for solving the problem of extractive-based text summarization. To achieve this goal, the ACO algorithm was supplied with learning techniques feature-based approach. Therefore, this research presented a number of contributions and aims to investigate the hypothesis:

"Developed Ant Colony based text summarization method is are able to extract optimal sentences for inclusion in the summary, thus generating higher quality summaries."

In this research, the design of our proposed methods allowed for the feature-based approach to accompany the ACO algorithm which then allowed it to better select top "n" representative sentences.

The proposed method also outperformed is better of evolutionary algorithms (GA and PSO).

This led us to further conclude that optimizing feature weights using robust evolutionary algorithms can generate better quality summaries compared to the other methods.

REFERENCES

Abuobieda, A., Salim, N., Albaham, A. T., Osman, A. H., & Kumar, Y. J. (2012, March). Text summarization features selection method using pseudo genetic-based model. In Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on (pp. 193-197). IEEE.

Abuobieda, A., Salim, N., Kumar, Y. J., & Osman, A. H. (2013). An improved evolutionary algorithm for extractive text summarization. In Intelligent Information and Database Systems (pp. 78-89). Springer Berlin Heidelberg.

Afantenos, S., Karkaletsis, V., & Stamatopoulos, P. (2005). Summarization from medical documents: a survey. Artificial intelligence in medicine, 33(2), 157-177.

Ahmed, A. A. (2005). Feature subset selection using ant colony optimization.

Aone, C., &Larsen, B. (1999, August). Fast and effective text mining using linear-time document clustering. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 16-22). ACM.

Baxendale, P. B. (1958). Machine-made index for technical literature: an experiment. IBM J. Res. Dev. 2(4), 354–361.

Blesa, M., & Blum, C. (2003). Ant Colony Optimization for the maximum disjoint paths problem. Technical Report ALCOMFT-TR-03-105, ALCOM-FT, Barcelona.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant System algorithm for thevehicle Routing Problem. Annals of operations research, 89, 319-328.

Casillas, J., Cordón, O., & Herrera, F. (2000, September). Learning fuzzy rules using ant colony optimization algorithms. In Abstract proceedings of ANTS2000-From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms (pp. 13-21).

Chen, L., Tu, L., & Chen, Y. (2006). An ant clustering method for a dynamic database. In Advances in Machine Learning and Cybernetics (pp. 169-178). Springer Berlin Heidelberg.

Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science, 34(1), 39-53.

Conroy, J. M., & O'leary, D. P. (2001, September). Text summarization via hidden markov models. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 406-407). ACM.

Cordon, O., de Viana, I. F., Herrera, F., & Moreno, L. (2000). A new ACO model integrating evolutionary computation concepts: The best-worst Ant System.

Cordone, R., & Maffioli, F. (2001). Coloured Ant System and local search to design local telecommunication networks. In Applications of Evolutionary Computing (pp. 60-69). Springer Berlin Heidelberg.

Costa, D., & Hertz, A. (1997). Ants can colour graphs. Journal of the operational Research Society, 48(3), 295-305.

De Campos, L. M., Fernandez-Luna, J. M., Gámez, J. A., & Puerta, J. M. (2002). Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning, 31(3), 291-311.

Dorigo, M., & Gambardella, L. M. (1996). A study of some properties of Ant-Q. In Parallel Problem Solving from Nature—PPSN IV (pp. 656-665). Springer Berlin Heidelberg.

Dorigo, M., Birattari, M., Blum, C., Gambardella, L. M., Mondada, F., & Stützle, T. (2004). ANTS 2004. LNCS, vol. 3172.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991). The ant system: An autocatalytic optimizing process (No. 91-016). Technical report.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26(1), 29-41.

Dorigo, M.,Colorni, A. , & Maniezzo, V. (1992). An Investigation of some Properties of an``Ant Algorithm". In PPSN (Vol. 92, pp. 509-520).

Edmundson, H. P. (1969). New Methods in Automatic Extracting. J. ACM. 16(2), 264–285.

Edmundson, H. P. (1969). New methods in automatic extracting. Journal of the ACM (JACM), 16(2), 264-285.

Faloutsos, C., Lafferty, J., Hauptmann, A., & Yang, Y. (2001). Informedia-II: Auto-Summarization and Visualization Over Multiple Video Documents and Libraries.

Fattah, M. A., & Ren, F. (2008). Automatic text summarization. World Academy of Science, Engineering and Technology, 37, 2008.

Feinstein, H., Brody, A., Leguizamo, J., & Lee, S. (1999). SUMMER OF SPIKE. Advocate, (789), 57.

Gambardella, L. M., & Dorigo, M. (1995, July). Ant-Q: A reinforcement learning approach to the traveling salesman problem. In ICML (pp. 252-260).

Gambardella, L. M., & Dorigo, M. (2000). An ant colony system hybridized with a new local search for the sequential ordering problem. INFORMS Journal on Computing, 12(3), 237-255.

Gambardella, L. M., Taillard, E. D., & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. Journal of the operational research society, 167-176.

Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence, 2(3), 258-268.

Hovy, E. and Lin, C.-Y. (1998). Automated text summarization and the SUMMARIST System

Hovy, E. and Lin, C.-Y. (1998). Automated text summarization and the SUMMARIST system.

Jing, R., H. Barzilay, McKeown, K. and Elhadad, M. (1998). Summarization evaluation methods: Experiments and analysis. In In AAAI Symposium on Intelligent Summarization. 60–68.

Julian Kupiec, F. C., Jan Pedersen (1995). A trainable document summarizer.

Jun,H.Zhang, J., Zhan, Z. H., Lin, Y., Chen, N., Gong, Y. J., Zhong, J. H., ... & Shi, Y. H. (2011). Evolutionary computation meets machine learning: A survey. Computational Intelligence Magazine, IEEE, 6(4), 68-75.

Li, L., Zhou, K., Xue, G. R., Zha, H., & Yu, Y. (2009, April). Enhancing diversity, coverage and balance for summarization through structure learning. In Proceedings of the 18th international conference on World wide web (pp. 71-80). ACM.

Lin, C. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram cooccurrence statistics. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 71–78.

Lin, C. Y. (1999, November). Training a selection function for extraction. In Proceedings of the eighth international conference on Information and knowledge management (pp. 55-62). ACM.

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (pp. 74-81).

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. 74 – 81.

Lin, C.-Y. and Hovy, E. (1997). Identifying topics by position. In Proceedings of the fifth conference on Applied natural language processing. 283–290.

Lin, C.-Y. and Hovy, E. (2002). From single to multi-document summarization: a

prototype system and its evaluation. In Association for Computational Linguistics. 457–464.

Lin, H., & Bilmes, J. (2011, June). A class of submodular functions for document summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 510-520). Association for Computational Linguistics.

Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165.

Luo, W., Zhuang, F., He, Q. and Shi, Z. (2010). Effectively Leveraging Entropy and Relevance for Summarization. In Information Retrieval Technology: 6th Asia Information Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 1-3, 2010, Proceedings, vol. 6458. Springer, 241.

Machnik, L. (2005). ACO-based document clustering method. In Annales UMCS, Informatica (Vol. 3, No. 1, pp. 315-323).

Mani, I. (2001). Automatic summarization (Vol. 3). John Benjamins Publishing.

Mani, I., & Bloedorn, E. (1997). Multi-document summarization by graph search and matching. arXiv preprint cmp-lg/9712004.

Mani, I., & Maybury, M. T. (Eds.). (1999). Advances in automatic text summarization (Vol. 293). Cambridge: MIT press

Mani, I., & Maybury, M. T. (Eds.). (1999). Advances in automatic text summarization (Vol. 293). Cambridge: MIT press.

Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. Knowledge and Data Engineering, IEEE Transactions on, 11(5), 769-778.

Marco Dorigo and Thomas Stützle. Ant Colony Optimization. Bradford Book, 2004. ISBN 0262042193.

Móro, R., & Bielikov, M. (2012, September). Personalized text summarization based on

important terms identification. In Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on (pp. 131-135). IEEE.

Neto, J. L., Freitas, A. A., & Kaestner, C. A. (2002). Automatic text summarization using a machine learning approach. In Advances in Artificial Intelligence (pp. 205-215). Springer Berlin Heidelberg.

Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). An ant colony algorithm for classification rule discovery. Data mining: A heuristic approach, 208, 191-132.

Porter, M. F. (1997). An algorithm for suffix stripping, Morgan Kaufmann Publishers Inc. 313–316.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.

Schoonderwoerd, R., Holland, O., Bruten, J., & Rothkrantz, L. (1996). Ants for Load Balancing in Telecommunication Networks, Hewlett Packard Lab., Bristol. UK, Tech. Rep. HPL-96-35.

Sekine, S., & Nobata, C. (2001). Sentence extraction with information extraction technique. In Proceedings of the Document Understanding Conference.

Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007, January). Document Summarization Using Conditional Random Fields. In IJCAI (Vol. 7, pp. 2862-2867).

Socha, K., Sampels, M., & Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In Applications of evolutionary computing (pp. 334-345). Springer Berlin Heidelberg.

Spärck Jones, K. (1999). Introduction to Text Summarisation.

Spark Jones, K. (1999). Invited keynote address. In Workshop on Intelligent Scalable Text Summarization.

Stützle, T., & Dorigo, M. (1999). ACO algorithms for the quadratic assignment problem. New ideas in optimization, 33-50.

Stützle, T., & Hoos, H. (1996). Improving the Ant System: A detailed report on the MAX-MIN Ant System.

Stützle, T., & Hoos, H. (1998, January). Improvements on the ant-system: Introducing the MAX-MIN ant system. In Artificial Neural Nets and Genetic Algorithms (pp. 245-249). Springer Vienna.

Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. Future generation computer systems, 16(8), 889-914.

Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. Future generation computer systems, 16(8), 889-914.

Suanmali, L., Salim, N., & Binwahlan, M. S. (2011). Genetic algorithm based sentence extraction for text summarization. International Journal of Innovative Computing, 1(1).

Taweesiriwate, A., Manaskasemsak, B., & Rungsawang, A. (2012, March). Web spam detection using link-based ant colony optimization. In Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on (pp. 868-873). IEEE.

Thu, H. N. T., Huu, Q. N., & Ngoc, T. N. T. (2013, April). A supervised learning method combine with dimensionality reduction in Vietnamese text summarization. In Computing, Communications and IT Applications Conference (ComComAp), 2013 (pp. 69-73). IEEE.

APPENDIX A

COLLECTION THE DUC2002 DATA SET

Table A.1: The 100 documents used in all carried out methods

No.	Folder	Set of Documents
1	D075b	AP880428-0041; AP880818-0088; AP880829-0222; AP881115-0113; AP890115-0014;
		AP900322-0112; AP900705-0149; AP901003-0006; WSJ880603-0129; WSJ910418-0105
2	D077b	AP891017-0195; AP891017-0199; AP891017-0204; AP891018-0084; AP891019-0037;
		LA101889-0066; LA101889-0108; LA102089-0172; LA102089-0177; LA102389-0075
3	D078b	AP880217-0100; AP880325-0239; AP880328-0206; AP890323-0218; AP890324-0014;
		AP890330-0123; AP891110-0043; AP900220-0065; LA033089-0190; LA033189-0114
4	D082a	AP880512-0096; AP880512-0157; AP881109-0161; AP881110-0227; AP890320-0158;
		AP891216-0037; AP891217-0053; LA012189-0060; LA051589-0055; LA121589-0192
5	D087d	AP880228-0013; AP880228-0097; AP880929-0042; AP881002-0048; AP881003-0066;
		AP900328-0128; FT923-8765; LA040790-0121; LA082889-0067; WSJ881004-0111
6	D089d	AP891115-0199; AP891116-0035; AP891116-0115; AP891116-0133; AP891116-0184;
		AP891116-0191; AP891116-0198; AP891117-0002; AP891118-0136; LA111689-0160
7	D090d	AP880625-0142; AP890519-0060; AP890519-0117; AP890710-0170; AP900408-0059;
		AP900829-0044; LA052089-0075; LA101390-0087; LA120189-0122; LA120389-0170
8	D092c	AP900621-0186; AP900622-0025; AP900623-0022; AP900624-0011; AP900625-0036;
		AP900626-0010; LA062290-0134; LA062290-0169; LA062390-0068; LA062590-0096
9	D095c	AP890117-0004; AP890117-0160; AP890118-0013; AP890118-0051; AP890118-0094;
		AP890119-0221; AP890121-0050; AP890121-0123; LA011889-0131; LA012189-0073
10	D096c	AP890122-0087; AP890203-0164; AP891117-0248; AP900128-0063; AP900130-0113;
		LA013090-0161; LA020890-0197; SJMN91-06025182; SJMN91-06025282; WSJ870122-0100

APPENDIX B

ORIGINAL DOCUMENT

Table B.1: Example Document from DUC2002

The 1988 Summer Olympics may well be remembered for the glory enjoyed by U.S. diver Greg Louganis and the disgrace experienced by Canadian sprinter Ben Johnson. Louganis won his second pair of gold medals after striking his on a springboard, which opened a cut requiring five stitches. Johnson had his gold medal and world record stripped because of drug use. Olympic officials believe that catching Johnson and nine other athletes with positive drug tests indicates that we will successful stop "doping". Surprisingly, the Soviets apparently won the hearts and cheers of the host South Koreans, while U.S. athletes experienced some anti-U.S. sentiment.

The Summer Olympics will be remembered for moments of glory like that enjoyed by U.S. diver Greg Louganis and the startling moment of disgrace when the gold was stripped from Canadian sprinter Ben Johnson. The Soviet Union won the first U.S.-Soviet Olympic medal contest since 1976, getting 55 gold medals to 37 for East Germany and 36 for the United States. Host South Korea rose to fourth in the world with 12 golds .The fear of terrorism and massive civil unrest prompted extraordinary security, but neither bogey materialized as nearly 10,000 athletes from 160 countries tested their mettle in 16 days of competition ended Sunday. As always, there were shining moments of glory, from an opening ceremony with exotic dancers and parachutists to a closing with hugs and tears, fireworks and dances, and the mascots of Seoul and Barcelona, the site of the 1992 Games, floating together into the starry night. Louganis claimed his second pair of gold medals after hitting his head on a springboard. He said he talked with his coach about quitting the Olympics after hitting the board and opening a cut that needed five stitches to close. ``We walked and discussed all the things we had gone through to get there,'' he said. ``I decided to stay in, and I'm glad I did.'' Louganis said Sunday he is retiring from diving to begin an acting career. What had been the highest moment of the Games _ Ben Johnson rocketing to victory over U.S. great Carl Lewis in a 100-meter dash world record _ led to the deepest pain when the Canadian was caught cheating with muscle-building anabolic steroids. Twenty years from now, when most of the records set in Seoul are broken, the impact of Johnson's disgrace will still be felt if athletes and trainers heed the events here and end doping. `There have been high points and some low points, and the most important low point was Ben Johnson,'' Juan Antonio Samaranch, president of the International Olympic Committee, said today. `That was indeed a blow.'' But catching Johnson and expelling him and nine other athletes with positive drug tests was an indication that ``we have won the battle against doping,'' Samaranch said. Sports officials from the United States and the Soviet Union on Sunday announced Sunday they would join forces to work towards the elimination of drugs from sport. A statement issued by the U.S. and Soviet Olympic committees said the groups would investigate using the exchange of testing teams, lab results and technical data; education programs; and uniform enalties. For the United States, a serious problem at the Seoul Games was a rising tide of anti-American sentiment. It was exacerbated by NBC's coverage, which the Koreans saw as anti-Korean and insensitive to local culture; the arrest of several American athletes; and the perceived rudeness of the U.S. team at the opening ceremony. The Soviets, meanwhile, cultivated friendship by bringing in the Bolshoi Ballet, the Moscow Philharmonic, films, a photo exhibit and copies of the Communist Party newspaper Pravda. The announcement of an unprecedented sports exchange program between South Korea and the Soviet Union and the arrival of the first Soviet diplomats since World War II also warmed relations. Soviet athletes, as they did in Calgary, Canada, during the Winter Games, made a special effort to meet with local people. American athletes tended to isolate themselves. All those differences became apparent at the sporting events, where Koreans often cheered louder for the Soviets or East Germans than they did for Americans, despite a close 40-year relationship with the United States. Still, there were Louganis and other Americans to unabashedly cheer for _ like sisters-in-law Florence Griffith Joyner and Jackie Joyner-Kersee, who led an assault on the track and field record books, and Louganis. Griffith Joyner won golds in the 100, the 200 and the 400-meter relay, and silver in the 1,600-meter relay.She set a world record in the 200 and an Olympic record in the 100.Joyner-Kersee won two golds, taking the heptathlon with a world record 7,291 points and the long jump with an Olympic mark of 24 feet, 3{inches.

APPENDIX C

Human1 Summary

APPENDIX D

Human2 Summary

Table D.1: Example of human 2 summary

There were several shining moments for US Olympians in the 1988 Summer Games. Diver Greg Louganis won a second pair of gold medals after a springboard accident, track star Florence Griffith Joyner won three golds and a silver medal, and her sister-in-law, Jackie Joiner-Kersee won two golds. Canadian Ben Johnson had rocketed to a world record performance overUS sprinter Carl Lewis in the 100 meter but was stripped of the gold medal when he tested positive for anabolic steroid use. The Games also saw increasing anti-American sentiment by the host Koreans because of perceived rudeness of the US team and NBC coverage of events.

APPENDIX E

List of Stop Words

Table E.1: Sample of List of Stop Words

a	again	although	anyone	around	against	always	anything
because	before	below	between	by	beforehand	beside	beyond
came	causes	com	considering	couldn't	can	certain	come
do	despite	different	doesn't	done	doing	down	definitely
each	else	et	everybody	exactly	elsewhere	etc	everyone
first	follows	formerly	from	for	forth	further	far
getting	go	gone	greetings	get	given	goes	got
have	hence	hereupon	his	haven't	her	hers	hither
if	indeed	instead	it'd	ignored	indicate	into	it'll
mainly	me	might	mostly	myself	mean	more	much
name	need	next	needs	nine	nor	nowhere	Namely
ones	otherwise	outside	ok	of	old	onto	our