e-ISSN: 2349-9745
p-ISSN: 2393-8161
Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Mobile Agents for Distributed Information Retrieval
An Optimized Mobile Agent System for Distributed Information Retrieval and Analysis

Altaf Husain[1], Shreedhar Yadawad[2], Ranganath Yadawad[3]
[1]*Department of CSE, SDMCET, DHARWAD,*
[2] *Department of CSE, SDMCET, DHARWAD,*
[3] *Department of CSE, SDMCET, DHARWAD*

Abstract— The emergence of Internet with millions of electronic stores, auctions and other services created the information is overloaded and it creates obstacle to the practical use and to obtain useful information over web. The use of mobile agents in this kind of applications that represents a beneficent mechanism and solves the problems present in centralized client-server systems, these are the programs with a persistent identity, move around a network and can communicate with this environment. We use optimized Mobile Agent system for the retrieval of distributed structured information in a scenario of retrieving and analyzing a specific product's reviews and storing in database. An effective web mining technique based on review clustering proposed for assessing a customer review of a particular product stored in database. It can be performed by, clustering the reviews into four groups by applying k-means clustering technique and compute the cluster weights. Assess quality of the given reviews and classify them by considering the cluster weights. A summarized opinion is provided based on classification.

Keywords- Optimized Mobile Agents; Distributed Data Retrieval; k-means clustering; quality assessment and classification.

I. INTRODUCTION

Today, the size and the number of available information sources are growing with faster speed. Finding and combining the relevant information is becoming a critical operation. We need facilities to perform integration operations and overcome problems of distribution and heterogeneity. We can access incredible amount of information at wide-area networks, where data is stored at heterogeneous sources and the accessibility to every source may change. One of the major task of today's research is finding effective mechanisms of sorting and accessing these distributed sources. Today, most of the models for distributed sources is client-server model. These models divide the distributed application code into the server which provides fixed services at the server machine and the client that remotely requests the service. This process may be extremely complex and it can be even worse if the client and server sites belong to different administration domains [1].

In this concept, we discuss the mechanism of mobile-agent system to this problem. The approach is based on the idea of using mobile agents to integrate distributed sources into a global system to provide database connectivity, processing and communication, and consequently eliminate the overheads of the existing mechanisms. The intention of using the mobile agents is for two purposes. Firstly, the mobile agents provide an efficient, flexible and asynchronous method for searching information or services in rapidly evolving networks: mobile agents are launched into the unstructured network and roam around to gather information. Second, mobile agents support intermittent connectivity, slow networks, and lightweight devices [2].

A mobile agent is a program, which represents a user in a computer network, and is capable of migrating autonomously from node to node, to perform some computation on behalf of the user. Mobile agents are defined as objects that have behavior, state, and location. Its tasks are determined by the agent application, and can range from online shopping to real-time device control to distributed scientific computing. Applications can inject mobile agents into a network, allowing

them to roam the network either on a predetermined path, or one that the agents themselves determine based on dynamically gathered information. Having accomplished their goals, the agents may return to their "home site" in order to report their results to the user.

The concept of Web Mining involves techniques for summarizing, classification and clustering of the web contents. It can provide useful and interesting patterns about user needs and contribution behavior. It targets the knowledge discovery, in which the main objects are the traditional collections of text documents and, more recently, also the collections of multimedia documents such as images, videos, audios, which are embedded in or linked to the Web pages. It is mainly based on research in information retrieval and text mining, such as information extraction, text classification and clustering, and information visualization.

The customers can now post reviews of products at merchant sites and express their views on almost everything. In the past few years, there has been an increasing interest in mining and assessing the customer reviews [9], [10], [11].

The customer reviews posted at online shopping sites vary greatly in quality. Thus, it is very essential to have a mechanism which is capable of assessing the quality of reviews for purchase decision or marketing intelligence. Identifying the quality of customer reviews is useful for both potential buyers and product manufacturers. For a potential buyer, it is more convenient and less time consuming to see at a glance feature by feature comparison of customer reviews. For a product manufacturer, it helps to find the strengths and weaknesses of his/her own products and also that of the competitors.

There are three main review formats commonly found on the web. Different review formats may need different techniques to identify and assess the quality of the reviews.

Format 1: Pros and Cons

-The reviewer is asked to describe Pros and Cons separately. e.g., Cnet.com uses this format.

Format 2: Pros, Cons and detailed review

-The reviewer is asked to describe Pros and Cons separately and also write a detailed review. e.g., Epinions.com uses this format.

Format 3: Free format

-The reviewer can write freely, i.e., no separation of Pros and Cons. e.g., Amazon.com uses this format.

The opinion orientations (positive or negative) of features are known from Format 1 and 2 because pros and cons are separated and thus there is no need to identify them.

In this paper, we propose an effective web mining technique for assessing the customer review of a particular product based on the review clustering. Given a product name and a set of URL's of web pages that contain customer reviews on the product, it works in two stages:

Stage1: Cluster the reviews into four groups by applying k-means clustering technique and compute the cluster weights.

Stage2: Assess quality of the given reviews and classify them by considering the cluster weights.

II. RELATED WORK

This work is inspired by current research in many fields. We draw from other agent projects using functionality ideas, various topics in mobility and ideas from distributed-systems process and optimization. The capabilities of agents have been applied to electronic commerce, promising a revolution in the way we conduct transactions [4].

As we've seen, there are several agent based applications that help the user to find the wanted information on the web, but however, we've found a few applications [5], [6] which use mobile agents for distributed database access. Because of this, the main purpose of this paper is to look at how the mobile agent paradigm can improve some distributed database and information retrieval related problems, such as the performance of an ecommerce prototype.

Research that is done in feature extraction and summarization, authors P. S. Hiremanth et. al. provides a novel approach for clustering the product reviews [9]. The authors Dejong [12], Tait [13] and Radev and McKeown [14] propose text summarization using template instantiation. This technique needs to design a template by identifying and extracting primary elements and facts in a document. Paice [15], Kupiec, Pederson and Chen [16], Hovy and Lin [17] have focused on text summarization using text extraction, which is based on representive sentences.

2.1. Java as Technology Base for Mobile Agents

In the current trend towards heterogeneous networks, which are composed by several different platforms, the mobility of binary code is problem hardly to overcome. An answer can be found in Java programming language, which combines the object-oriented programming style with the use of intermediate format called bytecode, which can be executed on each platform that hosts a Java virtual machine (JVM). Java is strongly network oriented and provides some support for mobility of code from the dynamic class loading to the definition of applets. Java implements a form of weak mobility, by serializing objects and sending them to another JVM.

The serialization mechanism permits to maintain the values of the instance variables, but it cannot keep track of the execution flow. Some of the properties of Java that make it a good language for mobile agent programming are:

1. **Platform-independence:** Java is designed to operate in heterogeneous networks. To enable a Java application to execute anywhere on the network, the compiler generates architecture-neutral byte code, as opposed to non-portable native code.
2. **Object serialization :** A key feature of mobile agents is that they can be serialized and de-serialized. Java conveniently provides a built-in serialization mechanism that can represent the state of an object in a serialized form sufficiently detailed for the object to be reconstructed later. The serialized form of the object must be able to identify the Java class from which the object's state was saved, and to restore the state in a new instance.
3. **Reflection:** Java code can discover information about the fields, methods, and constructors of loaded classes, and can use reflected fields, methods, and constructors operate on their underlying counterparts in objects, all within the security restrictions.

2.2. IBM Aglets: Java Mobile Agent Technology

The Aglets Software Developer Kit (ASDK) [7] developed at IBM Research Laboratory is a framework for programming mobile network agents in Java. From a technical point of view, the IBM's mobile agent called "aglet" (agile applet), is a lightweight Java object that can move autonomously from one computer to another for execution, carrying along its program code and state as well as the so far obtained data. An aglet can exist and execute tasks forever. One of the main differences between an aglet and the simple mobile code of Java applets, is the itinerary that is carried along with the aglet. By having a travel plan, aglets are capable of roaming the Internet collecting information from many places. The itinerary can change dynamically giving the aglet the sense of self-governing and the look of an intelligent agent (that of course is in the hands of the programmer) [7].

An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This requires from the remote host to have preinstalled Tahiti, a tiny aglet server program implemented in Java and provided by the Aglet Framework. A running Tahiti server listens to the host's ports for incoming aglets, captures them, and provides them with an aglet context (i.e., an agent execution environment) in which they can run their code from the state that it was halted before they were dispatched. Within its context, an aglet can communicate with other aglets, collect local information and when convenient halt its execution and be dispatched to another host. An aglet can also be cloned or disposed [1].

III. PROPOSED ARCHITECTURE

3.1. Product Agent System

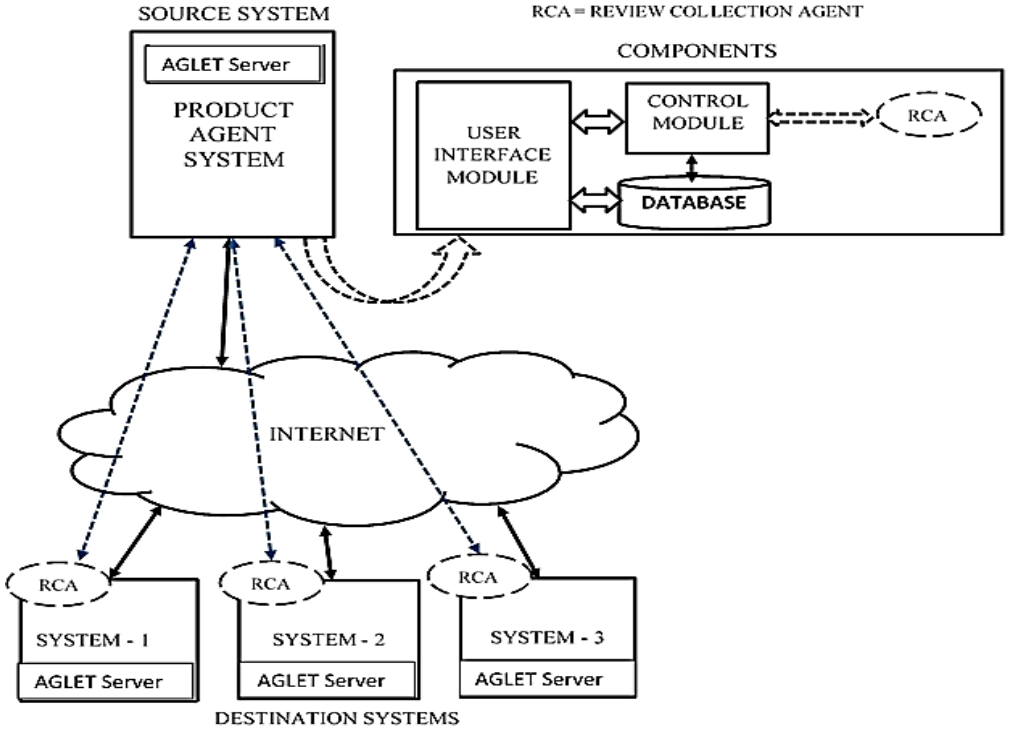

Figure 1: Product Agent System

The system is proposed to enable the Product Agent system to have the functionality mentioned above in the Figure 1. The system presented has the following: a) User Interface Module, b) Control Module and c) Review Collection Agents.

a) User Interface Module: In this component, the user contacts the system and inputs the product name for search. Once search completes, it presents the results obtained by the agents to the user. Product Name, and model are the information that user provides to the User Interface Module so that it may request the Control Module to create and dispatch the Review Collection agents according to the search criteria by the user.

b) Control Module: It controls the creation and release of Review Collection Agents to begin the search requested by the user. This module also aggregates the results found by the different agents. When the Control Module receives a request to send an agent, the latter is created on the "aglet" layer according to user's requirements and travels through the runtime layer, which converts the agent into an array of bytes and such array, on its turn, passes on to the ATP layer – Agent Transfer Protocol, to be sent to its destination.

c) Review Collection Agent: communicate with stores by accessing their databases and interpret the answers generated, converting them into a format that is understood by the control module. Before proceeding to their destination, the agents are coded in bit stream: the first segments are general information, such as the agent's identification, and the last segment is the byte array, the agent per se: code and state. The goal of the agents is to check the information found at their destination address, selecting only the information considered relevant and recommended according to the pre-determined rules.

3.2. Quality Assessment of Retrieved Distributed Data

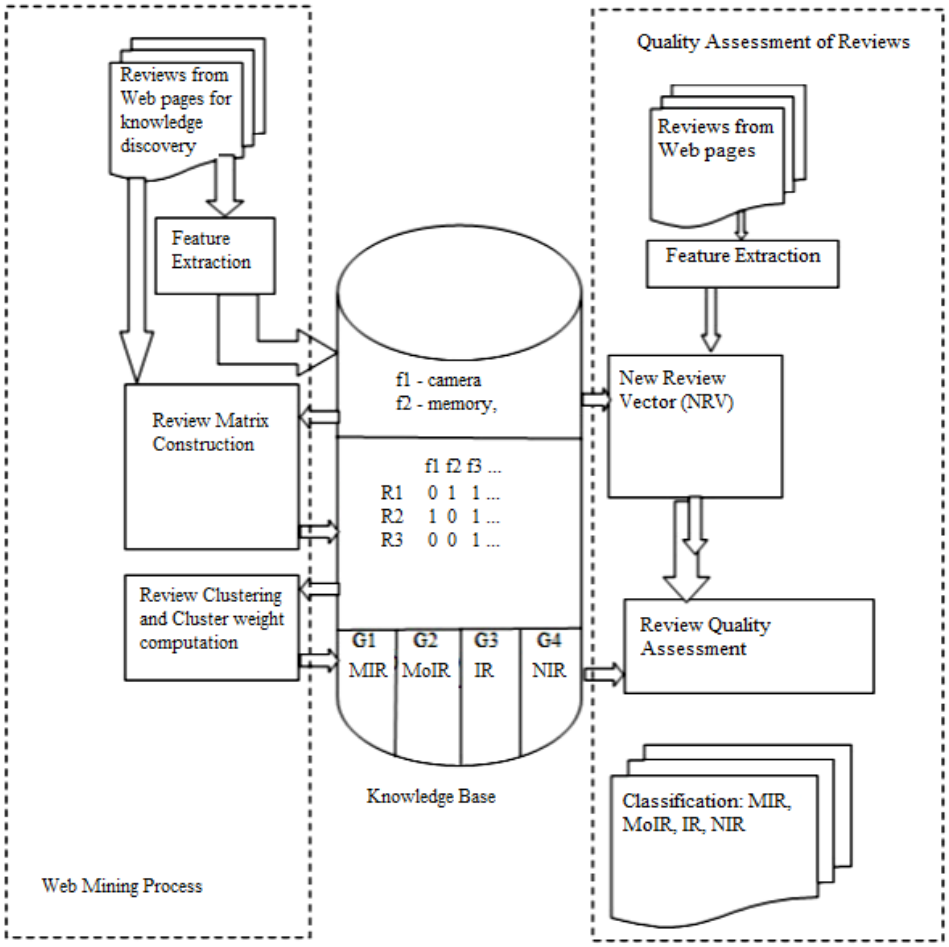

Figure 2: A review clustering Process

To assess and classify the customer reviews for a specific product, features must be identified and extracted accurately from the Web pages and stored in database. Since the product features are often domain dependent, it is desirable that the feature extraction system is as flexible as possible. We propose a novel and effective technique to extract the customer reviews from the web pages and classify them into different groups based on their quality using clustering technique. We define four types of review qualities, which are determined by applying the k-means clustering technique. The clustering algorithm partitions the features into four clusters (k=4), i.e., groups such that the similarity within a group is larger than that among other groups. The four types of reviews are:

(i) Most important Review (MIR): It is the one which corresponds to the maximum cluster weight. A feature weight is defined as the sum of corresponding feature values in the reviews of a cluster divided by the total number of reviews in that cluster. A cluster weight is defined as the sum of all the feature weights of that cluster.

(ii) More Important Review (MoIR): It is the one which corresponds to the next maximum cluster weight and whose value is less than the first one.

(iii) Important Review (IR): It is the one which corresponds to the next maximum cluster weight and whose value is less than the second one.

(iv) Non Important Review (NIR): It is the one which corresponds to all the remaining reviews and whose cluster weight is less than the third one.

The system model of the proposed technique, namely**,** Quality Assessment of Customer Reviews Extracted from Web Pages. As shown in Figure 2, a Review Clustering Approach consists of the following components [17]:

1. **Feature extraction**
2. **Review matrix construction**
3. **Review clustering and cluster weight computation.**
4. **Review quality assessment and classification.**

The output of each component is the input for the next component.

1. Feature Extraction

It extracts the features from the given set of reviews (pros and cons as separate reviews) extracted from the web pages. The features extracted from the customer reviews are stored in the feature database after checking whether that feature is already existing or not. A sample of extracted features stored in the database.

2. Review matrix construction:

The inputs for this component are the set of raw reviews and the feature set extracted in the earlier step. Consider that there are a total of m customer reviews for a particular product and n features are extracted from each of the reviews. We construct a review matrix M of order of m x n using the Algorithm 1.

Algorithm 1: Algorithm for review matrix construction.

```
for each review Ri in the raw review database
{
for each feature fj in the review
{
if fj is present in Ri then Mij = 1
else Mij = 0
}
}
```

3. Review clustering and cluster weight computation

Now, we propose to group the reviews into four groups by applying a k-means clustering technique with k = 4 and absolute difference of two data values as the distance measure, for the data set of reviews present in review matrix. The input to this component is the review matrix constructed in the Algorithm 1. The algorithm for grouping reviews based on clustering technique is given in Algorithm 2.

Algorithm 2: Proposed algorithm for grouping of reviews by clustering

Step 1: Construct the review matrix M for the review set using Algorithm 1.

Step 2: Apply k-means clustering technique with k = 4 for the review set and obtain four clusters of reviews.

Step 3: For each cluster, compute cluster weight Wg, g=1 to 4, as shown below:

a) Compute feature wise sum of the reviews in gth cluster, given by

$$Y_{gj} = \sum_{i=1}^{p} X_{ij}, \text{ for } j=1 \text{ to } n$$

where

n = number of features of the product

p = number of reviews of product in the gth cluster

Y_{gj} = sum of jth feature of all the reviews belonging to gth cluster.

X = sub matrix of review matrix M for gth cluster.

b) Compute the feature weight vector WV_g for gth cluster given by

$WV_g = (WV_{g1}, WV_{g2}, \ldots. WV_{gn})$

where $WV_{gj} = Y_{gj} / p$, for j = 1 to n, are the jth feature weights for gth cluster of reviews

c) Compute cluster weight W_g for gth cluster given by

$$Wg = \sum_{j=1}^{n} WV_{gj}$$

Step 4: Label the clusters as G1, G2, G3 and G4 groups with decreasing order of their cluster weights W_g, g = 1, 2, 3, 4. The corresponding feature weight vectors WV_g, g = 1, 2, 3, 4, are the representative vectors of the clusters G1, G2, G3 and G4, respectively. The G1, G2, G3 and G4 contain MIRs, MoIRs, IRs and NIRs, respectively. The feature weight vectors WVg, g = 1, 2, 3, 4, are used for the quality assessment of reviews, described in the next section.

4. Review Quality Assessment

The third step of the proposed technique is to find out the group to which a given review belongs based on its quality. A review from the raw review database, and the feature weight vector of each group are the inputs for rating the review quality assessment. The algorithm for review quality assessment is given in the Algorithm 3.

Algorithm 3: Algorithm for Review quality assessment.

1. Identify and extract the features appearing in the given review and store it in the New Review Vector (NRV).
2. Compute dot product NRVSg of NRV and WVg, g = 1, 2, 3, 4, given by

$$NRVS_g = \sum_{j=1}^{n} (WV_{gj})\,(NRV_j)$$

3. Let NRVSmax = max (NRVS1, NRVS2, NRVS3, NRVS4)
4. Determine the review quality assessment using the following criteria:

If NRVSmax = NRVS1, then the Review is Most Important (MIR).
If NRVSmax = NRVS 2, then the Review is More Important (MoIR).
If NRVSmax = NRVS 3, then the Review is Important (IR).
If NRVSmax = NRVS 4, then the Review is NonImportant (NIR).

IV. PERFORMANCE

The proposed system is to be experimented. At any time, a user can delegate tasks to agent. Such task is to look for reviews posted on a product by customers, collect them and perform mining process to obtain certain results. Users are notified through the User Interface Module about the quality and opinion based on reviews.

The Review Collection Agent can be launched to query multiple databases in parallel. The agents are launched to different hosts on the web, cooperate and communicate with each other to perform complicated tasks efficiently. By using the agents to encapsulate all interactions between the client and the server machine, the client becomes light and portable. This is possible because the only responsibility of the user machine is to specify the URL address of the database server, the query to be performed and the itinerary. The rest, such as the loading and initialization of JDBC drivers, is the responsibility of Control agent. The effect on performance is quite significant. The outcome of Review collection agent is taken for the mining process through User Interface Module.

As compared to Client – Server type of communication, Mobile agent system provides huge bandwidth saving.

V. CONCLUSION

In this concept, we introduce a new approach for developing client/server applications on the web using java mobile agents. The product agents system implements an application to investigate and test the convenience of using the mobile agent technology at a distributed environment. This application is able offer information services where the agents under user's thumb, liberating him of doing routine tasks of searching information.

The collected reviews by Review Collection agent used in web mining technique for assessing the customer review extracted from a web page for a particular product based on the clustering of reviews. The quality assessment of a customer review is categorized as most significant review, more significant review, significant review or insignificant review. This is performed in two steps: (1) Extract the features of reviews and group the reviews by a clustering technique and then compute cluster weight for each of the groups, (2) Assess the review by projecting its review vector on to the feature weight vectors of all the four groups. The reviews belonging to insignificant group may be ignored while review summarization, and thus optimizing the process of review quality assessment. The clustering of reviews is found to be more effective.

REFERENCES

[1] Rosane Maria Martins, Luci Pirmez And Luiz Fernando Rust Da Costa Carmo, "Databases And Information Retrieval: Taking Advantage Of Mobile Agent Technology", NCE/UFRJ, Brasil.
[2] Pittura, E., Samaras, G. "Data Management For Mobile Computing", Kluwer Academic Publisher, 1997.
[3] Rahul Jha, "Mobile Agents For E-Commerce", Kr School Of Information Technology Indian Institute Of Technology, Bombay.
[4] Prof. Y M Naik, Shilpa Tarihal, Roopali Swami, Ashwini Purandare, Kiran Adike, "Distributed Information Retrieval Using Mobile Agent, Department Of Computer Science And Engineering, Hirasugar Institute Of Technology Nidasoshi-591236 Dist: Belgaum Karnataka, India, Aijrstem, Issn (Print): 2328-3491, Issn (Online): 2328-3580, Issn (Cd-Rom): 2328-3629, Aijrstem 13-159, 2013
[5] Mattern, F., Fünfrocken,S, "Mobile Agents As An Architectural Concept For Internet-Based Distributed Applications". http://informatik.tudarmstadt.de/vs/publikationen/papers/kivs99-html/kivs99.html
[6] Vlach, R, "The Porter Mobile Database Agent". http://aglaja.ms.mff.cuni.cz/~vlach/ porterdemo.html
[7] "Aglets Workbench", IBM Japan Research Group, http://aglets.trl.ibm.co.jp
[8] P.S Hiremath, Siddu P. Algur, S. Shivashankar, "Quality Assessment of Customer Reviews Extracted From Web Pages: A Review Clustering Approach", International Journal on Computer Science and Engineering, Vol. 02, No. 03, pages: 600-606, 2010.
[9] Hu Minqing and Bing Liu, "Mining opinion features in customer reviews". AAAI. Vol.3 No.2, pp. 76-85, 2004.
[10] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques". EMNLP- 2002, pp.79-86, 2002.
[11] Ana-Maria Popescu, O. Etzioni, "Extracting product features and opinions from reviews". HLT-EMNLP, pp. 32-33, 2005.
[12] G. Dejong, "An Overview of the FRUMP System. In Stratregies for Natural language Parsing publisher", pp. 149-176, 1982.
[13] J. Tait, "Automatic summarising of english text. PhD Dissertation", Cambridge: University of Cambridge, 1983.
[14] D. Radev, K McKeown, "Generating natural Language summaries from multiple on-line sources". Cisticsomputational Lingu 24, No. 3, pp.469-500, September 1998.
[15] C. D. Paice, "Constructing Literature Abstracts by Computer: techniques and prospects. Information Processing and Management", pp.171-186, 1996.
[16] J. Kupiec, J. Pedersen, F. Chen, "A Trainable Document Summerizer", SIGIR, Vol No. (1995), pp. 68-73, 1995.
[17] E. Hovy, C. Y. Lin, "Automated Text Summarization in Summarist". ACL Workshop on Intelligent, Scalable Text Summarization, pp.89-96, 1997.

