Improving Laboratories Efficiency through Website Using Text Mining

Dr. Abeer Tariq
Computer Sciences Department, University of Technology/Baghdad
Email:Abeer282003@yahoo.com

Received on: 16/2/2012& Accepted on: 4/10/2012

ABSTRACT

Text mining is an emerging technology that can be used to augment existing data in corporate databases by making unstructured text data available for analysis.

This research aim to present a proposed text mining system customized to improve laboratories efficiency. This is done by taking the electronic comments and e-mails produced to the organization web as inputs for that proposed mining system. The proposed text miner is customized for emails and comments written to the organization. For that the basic text mining algorithm will almost be modified by adding new steps, modify some steps by customizing Natural Language Processing (NLP), data mining techniques and building the document database. The proposal applied on the Computer Sciences Department Web and the results obtained are suitable to be published.

Keywords*:* Text Mining, natural language processing, naive bayes classifier, text clustering.

تحسين كفاءة المختبرات من خلال الموقع الالكتروني عن طريق تحليل النص

الخلاصة

تنقيب النص هي تكنولوجيا منبثقة يمكن استخدامها لتعزيز البيانات الموجودة في قواعد بيانات الشركات من خلال جعل البيانات النصيه متاحه للتحليل.

هذا البحث يهدف إلى تقديم نظام بيانات نصي مقترح مخصص لتحسين كفاءه المختبرات من خلال اخذ التعليقات الالكترونية ورسائل البريد الالكتروني المقدمة إلى شبكة معلومات المنظمة كمدخلات للنظام التحليلي المقترح . ولقد تم تخصيص هذا النظام لاستلام رسائل البريد الالكتروني و التعليقات المكتوبة للمنظمة, ولأجل هذا يمكن تعديل خوارزمية التحليلي النصي الاساسية بإضافة بعض الخطوات و تخصيص لغات البرمجة الطبيعية و تقنيات تحليل البيانات و بناء قواعد بيانات الوثائق. البحث المقترح طبق لشبكة معلومات قسم علوم الحاسوب و النتائج التي تم الحصول عليها مناسبة للنشر.

الكلمات المرشدة. تنقيب النص, معالجة اللغات الطبيعية, المصنف, تجميع النص

https://doi.org/10.30684/etj.31.2B.5
2412-0758/University of Technology-Iraq, Baghdad, Iraq

INTRODUCTION

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets [1]. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve their performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction [2] [3].

Text Mining is an important step of Knowledge Discovery process. It is used to extract hidden information from not structured or semi-structured data [4]. This aspect is fundamental because much of the Web information is semi-structured due to the nested structure of HTML code, much of the Web information is linked, and much of the Web information is redundant. Web Text Mining helps whole knowledge mining process to mining, extraction and integration of useful data, information and knowledge from Web page contents, see Figure (1).[5]

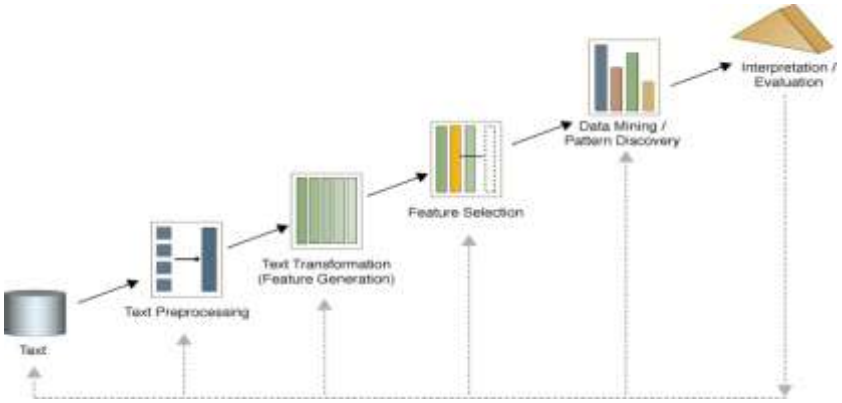

Figure (1): Text Mining.

As shown in Figure (1), the dotted lines present the idea of backtracking among text mining stages. Challenges of Text Mining are, large textual database (Web is growing, Electronic Publications), Access (No uniform access over all sources and Each source has a separate storage and algebra), Security and Authority of source (IBM is more likely to be an authorized source then my second far cousin), Ambiguity (Word ambiguity which imply Pronouns (he, she …), - Synonyms (buy, purchase…) and Words with multiple meanings (bat – is related to baseball or mammal)), Semantic ambiguity (The king saw the rabbit with his glasses (multiple meanings), Noisy data (Spelling mistakes, Abbreviations and Acronyms) ,Not well structured text (Email/Chat rooms imply "r u available ?) And Speech (Multilingual)[6].

The problem is that there are a huge number of texts submitted to the enhancement of the laboratories. The method of processing the gigantic amount of text is by extracting an abstract from these comments to obtain the important meaningful and desired words out of these comments which are related to the enhancement process.

THEORETICAL BACKGROUND

The naive Bayes text classifier

Text classification is the problem of assigning a document D to one of a set of $|C|$ predefined categories $C = \{c1, c2, c|C|\}$. Normally a supervised learning framework is used to train a text classifier, where a learning algorithm is provided a set of N labeled training examples $\{(di , ci) : i = 1, . . . , N\}$ from which it must produce a classification function $F:D \rightarrow C$ that maps documents to categories. Here di denotes the ith training document and ci is the corresponding category label of di. We use the random variables D and C to denote the document and category values respectively. A popular learning algorithm for text classification is based on a simple application of *Bayes' rule*:

$$P(C = c \mid D = d) = \frac{P(C = c) \times P(D = d \mid C = c)}{P(D = d)} \quad (1)$$

Where D and C are instances of D and C, to simplify the presentation, we re-write Eq. (1) as:

$$P(c \mid d) = \frac{P(c) \times P(d \mid c)}{P(d)} \quad (2)$$

Bayes' rule decomposes the computation of a posterior probability into the computation of likelihood and a prior probability. In text classification, a document d is normally represented by a vector of K attributes $d = (v1, v2, ...vK)$[2] Computing $p(d \mid c)$ in this case is not generally trivial, since the space of possible documents $d = (v1, v2, ...vK)$ is vast. To simplify this computation, the naive Bayes model introduces an additional assumption that all of the attribute values, vj , are independent given the category label, c. That is, for $i \neq j$, vi and vj are conditionally independent given c. This assumption greatly simplifies the computation by reducing Eq. (2) to:

$$P(c \mid d) = P(c) \times \frac{\prod_{j=1}^{K} P(v_j \mid c)}{P(d)} \quad (3)$$

Based on Eq. (3), maximum a posterior (MAP) classifier can be constructed by seeking the optimal category which maximizes the posterior $P(c \mid d)$:

$$c^* = \arg\max_{c \in C}\{P(c \mid d)\} \quad (4)$$

$$= \arg\max_{c \in C}\left\{P(c) \times \frac{\prod_{j=1}^{K} P(v_j \mid c)}{P(d)}\right\} \quad (5)$$

$$= \arg\max_{c \in C}\left\{P(c) \times \prod_{j=1}^{K} P(v_j \mid c)\right\} \quad (6)$$

RELATED WORKS

Many research performed to enhance text mining some of these are:

1. Give a survey on text mining facilities in XML and explain how typical application tasks can be carried out using proposed framework. Present techniques for count-based analysis methods, text clustering, text classification and string kernels [7].
2. The application of neural networks in the data mining has become wider. Although neural networks may have complex structure, long training time, and uneasily understandable representation of results, neural networks have high acceptance ability for noisy data and high accuracy and are preferable in data mining [8].

THE PROPOSED TEXT MINING

The proposed system consists of three steps (phases) that work simultaneously to fulfill the goal of this work:

1. The first step is the training phase.
2. The second step is the classification phase.
3. The third phase is the clustering phase.

The structure of the proposed system design is shown in figure(2).

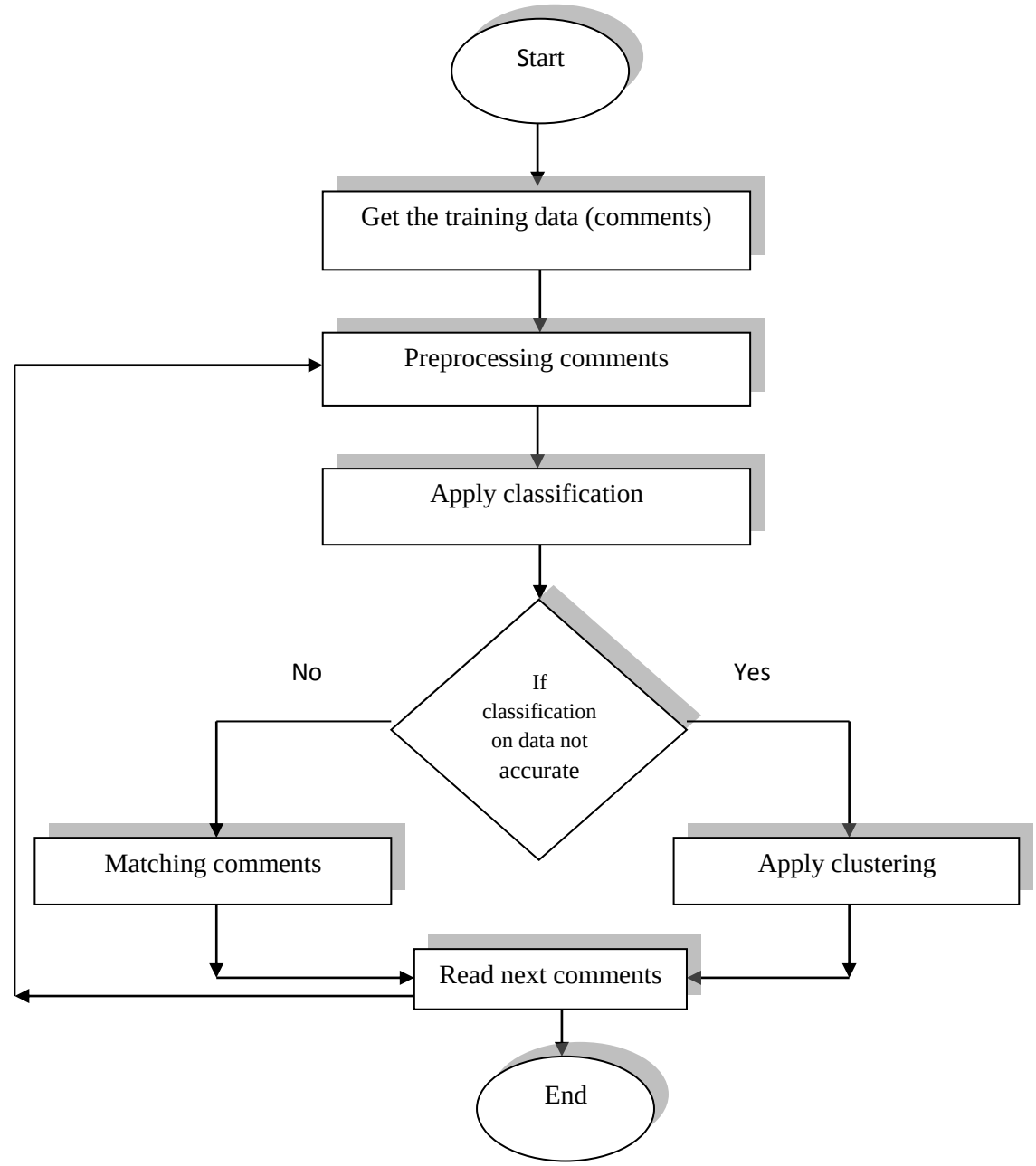

Figure (2) Steps of Research Methodology.

1. General Proposed Text Mining Algorithm
Input: all e-mails and e-comments related with organization
Output: relations and predictions enhancing organization
Process:
Step1: Training phase

1. Get all e-mails and e-comments related with organization.
2. Each one deal with it as document and apply the following customized proposed algorithms:
   - Tokenize the documents and then apply the part of speech on all tokens (such as read (verb)), explained in algorithm (1).

Algorithm(1) tokenization

Input: email or comment
Output: tokens
Process: 1. while not EOF do 2. collect characters 3. if char is not space or punctuation mark then with each space or punctuation consider collected characters as token. 4. End {while}

   - Clean up documents from such as (stop words, funny faces, collection of stars character and others), explained in algorithm(2).
   -

Algorithm (2) removing undesired tokens

Input: tokens
Output: related tokens
Process: 1. while not EOF do 2. take next token 3. if token is stop word or funny face or any trash then delete the token else keep it 4. end {while}

   - Treat each token with ambiguity (such as the boy saw the dog with his glasses).
   - Convert the parse tree of each statement in documents into graphs.
3. Return each word in document to it is root (such as walking, it is root walk).
4. Features extraction from each document which presented by considering all good words.
5. Build the final training database of documents to text mining.

Step2: Classification

Customized Naive Bayes Classifier Algorithm

Input: new document d;

Output: classified d

Process:

1. determine classes C={c_1,c_2,….,c_l};
2. Compute the probability that *d* is in each class $c \in C$
3. for($c_i \in C$)
4. begin
5. compute the probability by the following equation:

$$\Pr(c_i|d) = \frac{\Pr(d|c_i)\Pr(c_i)}{\Pr(d)} = \frac{\Pr(d|c_i)\Pr(c_i)}{\sum_{c_j \in C} \Pr(d|c_j)\Pr(c_j)} \quad \text{..... (1)}$$

6. terms w_i in document are independent each other:

$$\Pr(ci|d) = \frac{\Pr(c_i)\prod_{J=1}^{|d|} \Pr(w_j|c_i)}{\sum_{c_k \in C}\left(\prod_{j=1}^{|d|} \Pr(w_j|c_k)\right)\Pr(c_k)} \quad \text{.....(2)}$$

7. end
8. determine to *d* the **class** *c* with the highest probability:

$$\Pr(d|c) = \max_{\vec{c} \in C}\left(\Pr(d|\vec{c})\right) \quad \text{.....(3)}$$

9. end process

Step 3: Clustering

Customized Hierarchic agglomerative clustering (HAC) Algorithm and the output will be dendrogram of clusters. A dendrogram (from Greek dendron "tree", -gramma "drawing") is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering.

Input: D:={d_1,d_2,…,d_n };

Output: dendrogram of clusters

Process

1. Calculate similarity matrix SIM[i,j]

2. Repeat

3. Merge the most similar two clusters, K and L, to form a new cluster KL

4. Compute similarities between KL and each of the remaining cluster and update SIM[i,j]
5. Until there is a single (or specified number) cluster
6**.** End process

Example:
The following database is a general training database, see Table(1):

Table (1): general training database.

	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature(m)
d1	W1	W2	W3	W4	W5	W6	Wn
d2	W1	W2	W3	W4	W5	W6	Wn
d3	W1	W2	W3	W4	W5	W6	Wn
d4	W1	W2	W3	W4	W5	W6	Wn
dn	W1	W2	W3	W4	W5	W6	Wn

Some Comments from the users (examples), as known comments may have many of noisy characters such as *, funny faces and unstructured sentences.

1. ***The computer is infected with strong viruses***
2. The manager of the laboratories not a good one :((
3. The courses that teaches in the laboratories is not updated
4. Teachers not good
5. The computers in the laboratories must replace with new one
6. Every laboratory must contain wire and wireless connection
7. My degree is bad in the laboratory
8. The computer is not working

In computer science department every laboratory contains the following
1. Personal Computers (PC)
2. Wire or wireless connecter
3. At least two lecturers for any subject
4. Printers
5. One manager for the all laboratories
6. One sheet for each subject which contains the experiments

Example (1) : Suppose we take the first comment:
The computer is infected with strong viruses
Apply the suggested algorithm on it (training phase)

1. Read the comment and apply the following algorithms:
2. Apply algorithm 1. (tokenization)

Token1 = *** Token2 = the
Token3 = computer Token4 = is
Token5 = infected Token6 = with
Token7 = strong Token8 = viruses

Token9 = ***

3. Apply algorithm 2. (clean up)
   Remove *** , and ***
   Remove the, is, with
4. The comment is unambiguous
5. Convert the parse tree into graphs
6. Return word into it's root
   Infected = infect
7. The result contain the following features
   D1: Feature1 = word1= computer
   Feature2 = word2= infect
   Feature3 = word3= strong
   Feature4 = word4= viruses
8. The final training database is as in Table (1):
9. apply either classification or clustering according to the feature extraction

Implementation

The first step in the proposal is to convert all emails and document to uniform file with extension .txt, that by open all files in the proposed subprogram and click save command to save it in specific store with predefined extension, see Figure (3).

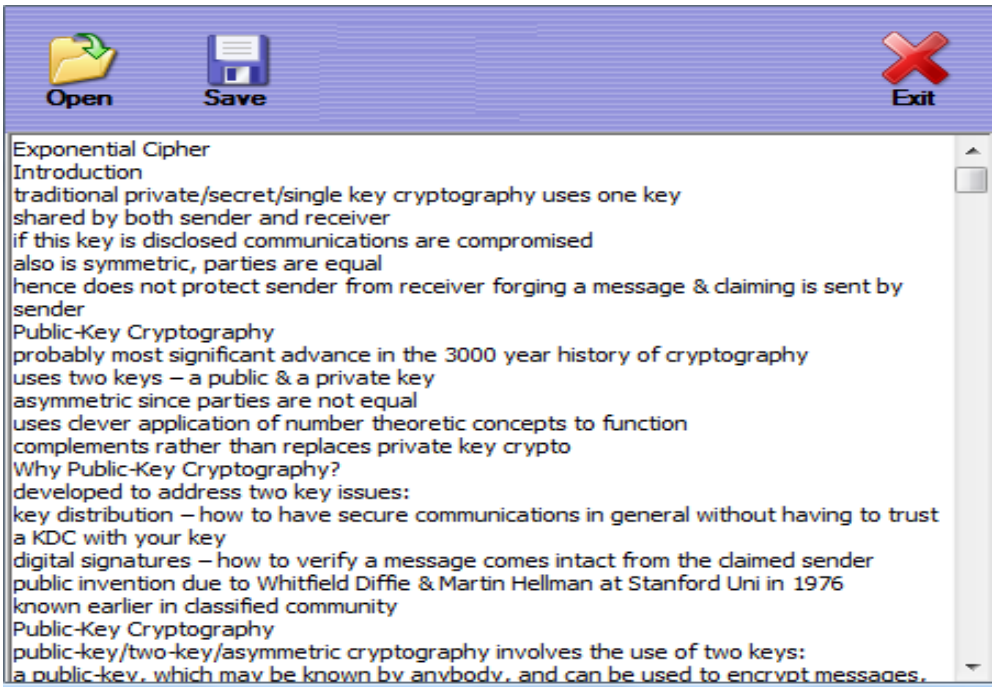

Figure (3): Editor Program.

After uniform files begin with tokenization and all traditional Natural Language Processing (NLP) steps, then begin with feature selection, using Artificial Neural

Network (ANN), by specified programs NuClass, see Figure (4), this program provides basic architecture for machine learning algorithms. Weka, see Figure (5), is a program which consists of a collection of machine learning algorithms for data mining tasks. The advantages of such platforms for the programmer are their availability for the said programmer and the fact that they are compact and portable, thus allowing for ease of use. This is due to the fact that they are fully integrated into JAVA, therefore enabling them to run on virtually any modern computing platform. Another element that contributes to the ease of use is the user-friendly graphical interface.

These platforms are comprehensive collections of data preprocessing techniques. That is why they can be used for this proposal.

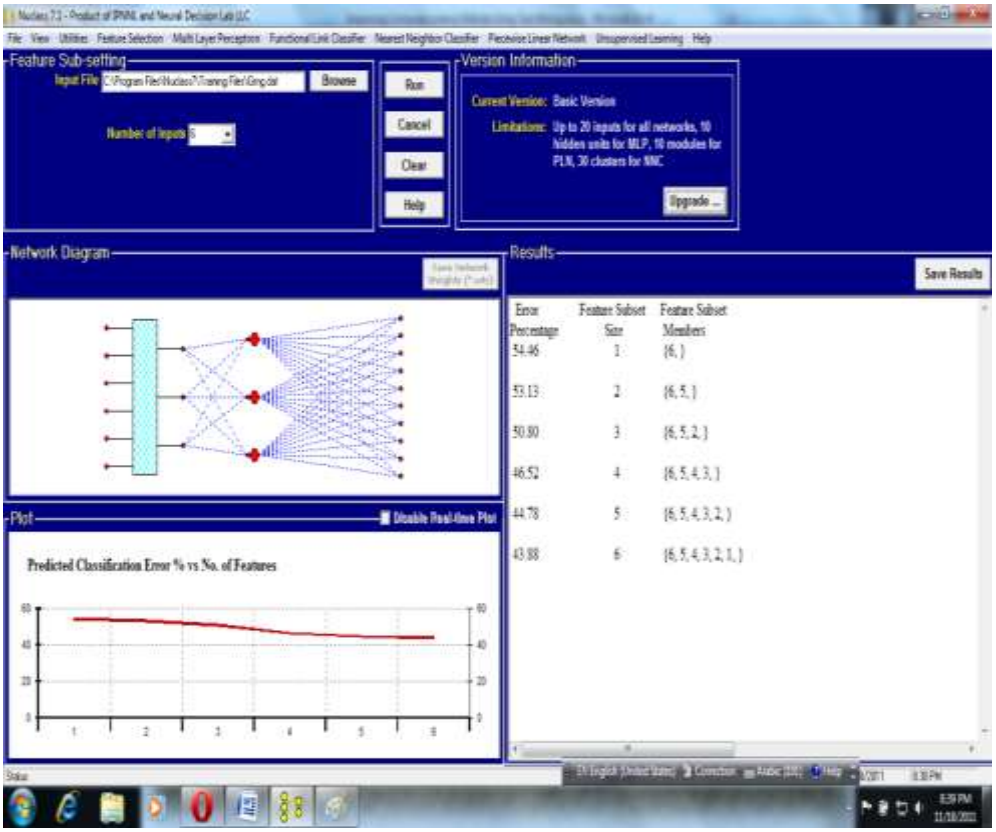

Figure (4): Feature extraction using NuClass Program.

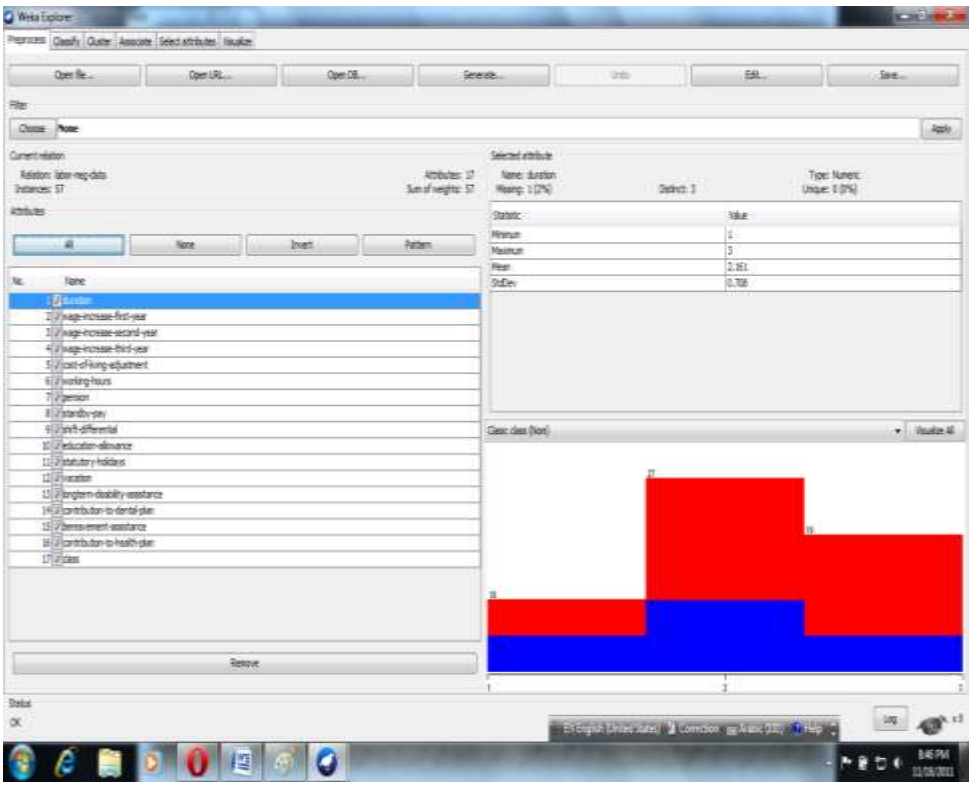

Figure (5): Feature extraction using Weka Program.

From the extracted information, the classification begins by using an algorithm of Customized Naive Bayes Classifier and ANN, by specified programs NuClass, see Figure (6) and Weka, and see Figure (7).

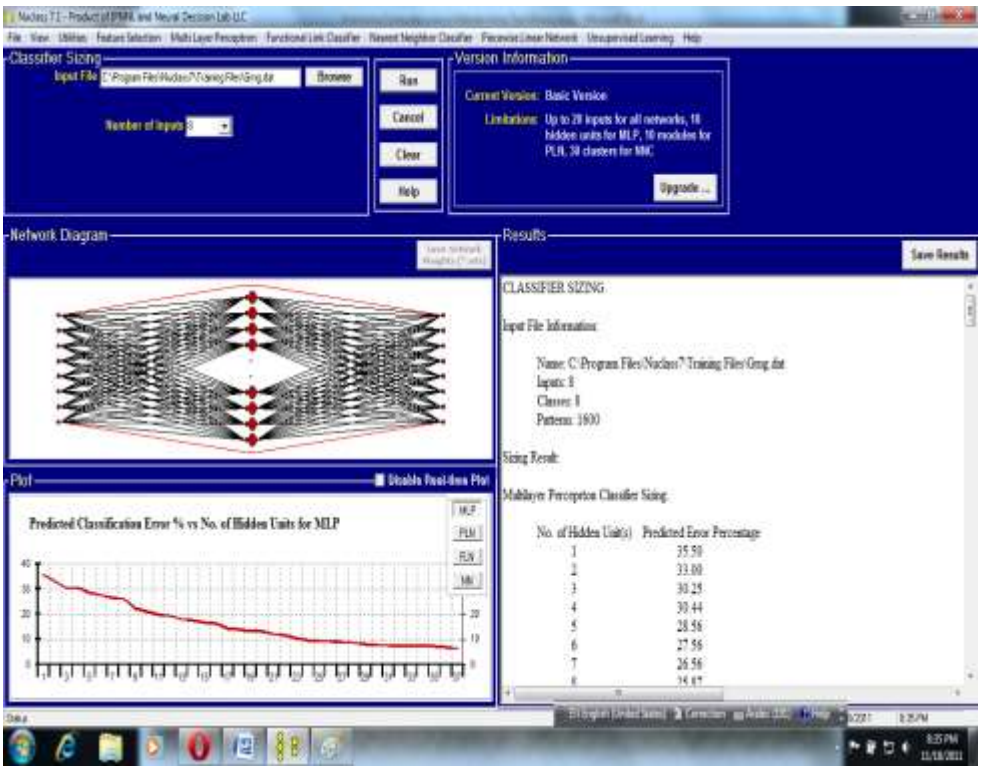

Figure (6): Classification using NuClass Program.

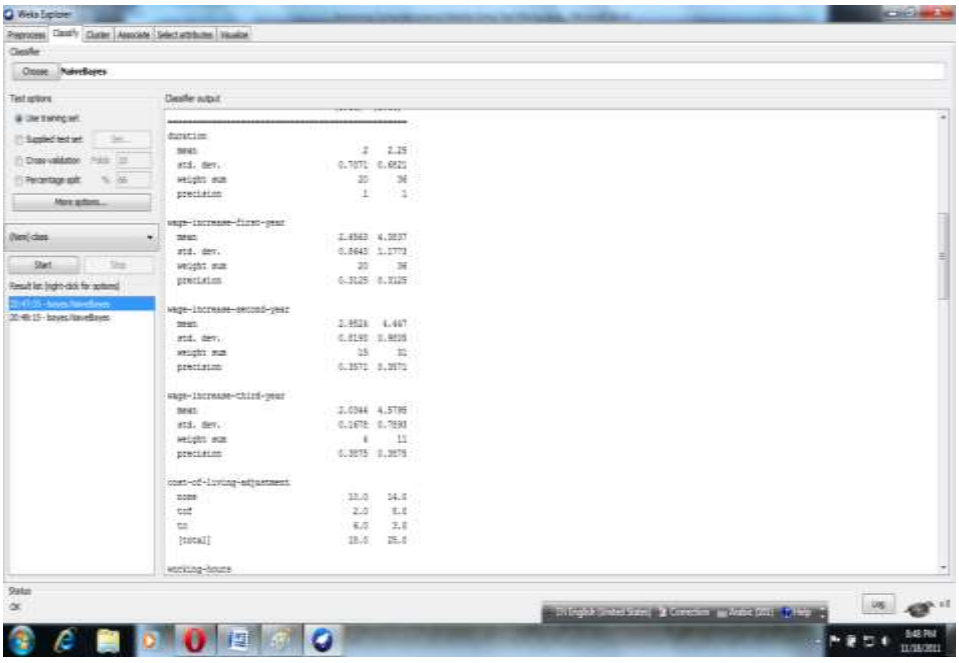

Figure (7): Classification using Weka Program.

The clustering process begins after the data is extracted using Customized Hierarchic agglomerative clustering *(*HAC*)*, by specified programs NuClass, see Figure (8) and Weka, and see Figure (9).

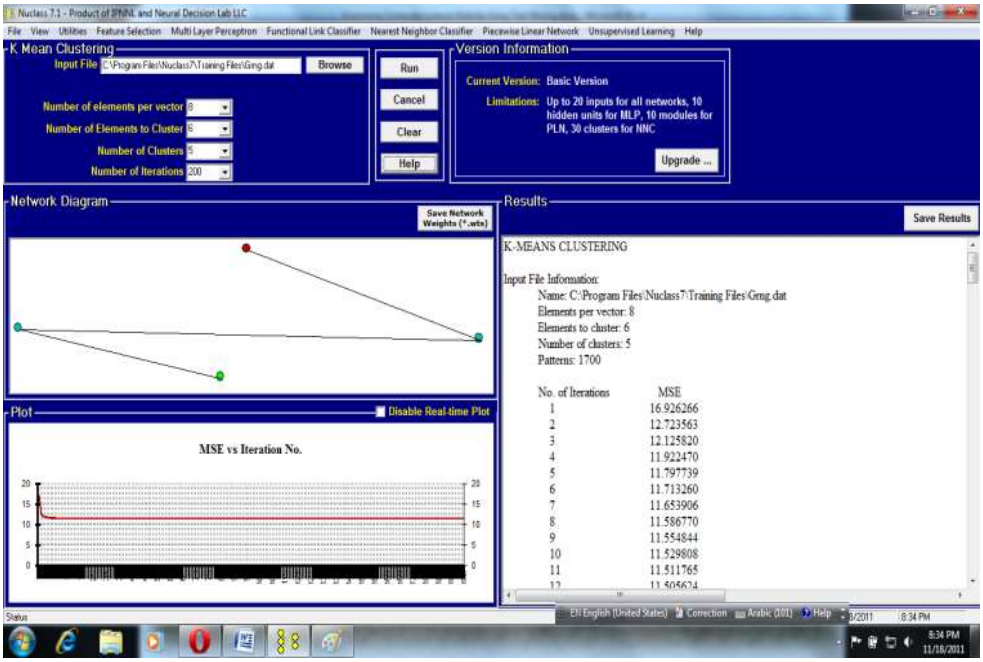

Figure (7): Clustering using NuClass Program.

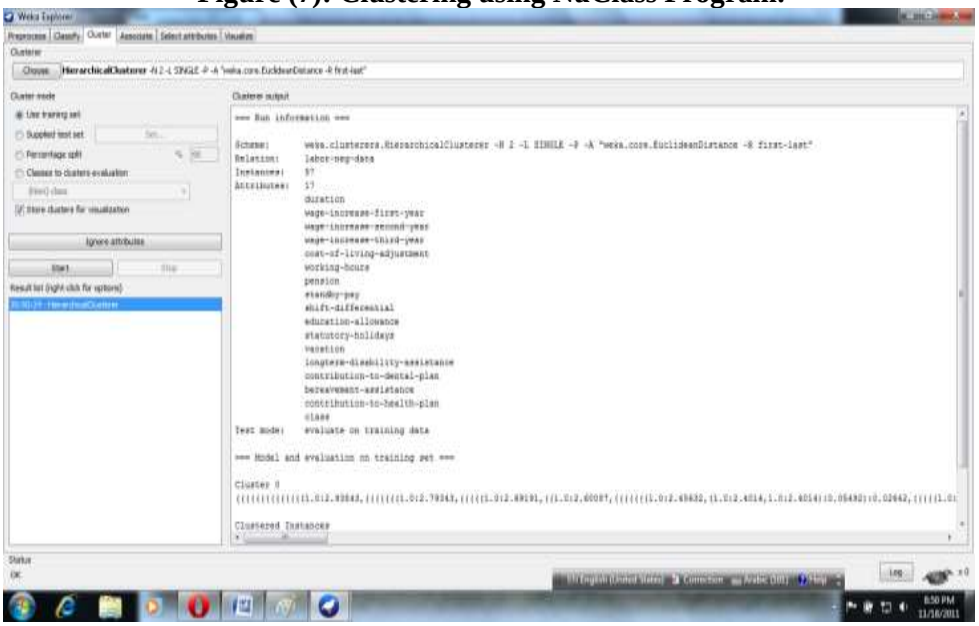

Figure (8): Clustering using Weka Program.

DISCUSSION AND CONCLUSIONS

This research reached to the following points:

1. Text mining success depending on strong methodology used in natural language processing (NLP).
2. Feature selections depend on the words extracted from text make the proposal much more flexible since the words differ from text to another.
3. Using these techniques (two clustering methods and two classification methods) for feature selections give support and strength to the obtained results.

REFERENCES

[1].Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques",2005, www.cs.sfu.ca/~han/DM_Book.html

[2]. Feldman,R. J. Sanger, "Advanced Approaches in Analyzing Unstructured Data", Cambridge Univ. Press, 2007

[3].Patil, K., Brazdil, P., and SumGraph: "Text Summarization using Centrality in the Pathfinder Network", International Journal on Computer Science and Information Systems, 2(1), pp. 18-32, 2007.

[4].Bilisoly, R., "Practical Text Mining with Perl", Wiley Publishing, 2008.

[5].Feinerer, I., K. Hornik, and D. Meyer, "Text mining infrastructure in R," Journal of Statistical Software 25:5, Mar. 2008, http://www.jstatsoft.org/v25/i05

[6].Kolyshkina, I. and M. van Rooyen, "Text Mining For Insurance Claim Cost Prediction", Presented at the XVth General Insurance Seminar Institute of the Actuaries of Australia, October 2005.

[7].Ingo Feinerer, et al, "Text Mining Infrastructure in R", Journal of Statistical Software March 2008, Volume 25, Issue 5.
http://www.jstatsoft.org/

[8].Xianjun Ni, "Research of Data Mining Based on Neural Networks", World Academy of Science, Engineering and Technology 39 2008.