British Journal of Mathematics & Computer Science
4(12): 1642-1664, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Centroid-Based and Bayesian Algorithms Performance

Ghaleb Al-Gaphari[1*], Fadl M. Ba-Alwi[1] and Saeed Abdullah M. Al Dobai[1]

[1]Computer Faculty Sana'a, University, P. O. Box 1247, Sana'a, Yemen.

Received: 15 November 2013
Accepted: 09 March 2014
Published: 05 April 2014

Original Research Article

Abstract

Since, the amount of textual information available on the web is estimated by terra bytes. Then, there should be an efficient algorithm to summarize such information. The algorithm would speed up the process of information reading, information accessing and decision making process. This paper investigates Bayesian classifier (BC) and a Centroid -Based algorithm (CBA) performance in terms of Arabic text summarization problem (ATS). Both algorithms are implemented as a software program. The Centroid -Based algorithm (CBA) extracts the most important sentences in a document or a set of documents (cluster). This algorithm starts computing the similarity between two sentences and evaluating the centrality of each sentence in a cluster based on centrality graph. Then the algorithm extracts the most important sentences in the cluster to include them in a summary. Whereas the Bayesian algorithm categorizes each sentence to be in text summary or out of text summary classes depends on its features vector. Both algorithms are evaluated by human participants and by an automatic metrics. Arabic NEWSWIRE-a corpus is used as a data set in the algorithms evaluation. The F-measure is obtained for both algorithms results. The Centroid -Based algorithm records 0.7199 and the Bayesian algorithm records 0.623.Thereforethe Centroid -Based algorithm (CBA) outperforms the Bayesian algorithm. The CBA results show that, the CBA is a robust algorithm compared to BC. It show a low deviation average that means the CBA gives similar result either contains bugs or not compared to BC. It is able to compress or reduce the text into 25% of its original size without losing the main idea behind the original text. This property makes the algorithm distinguishable among others used for the same purpose. Also, it outperforms all those techniques which are included in this paper when it is used for Arabic text summarization.

General Terms: AI Applications, NLP, Text Mining and AI Algorithms.

Keywords: Text Summarization, Text Mining and Centrality Concept.

1 Introduction

Information plays an important role in human daily life in different modern societies. Unfortunately, when large amounts of knowledge are produced and available through the web the

*Corresponding author: drghalebh@yahoo.com;

process of efficient, effective distribution and accessing this valuable information becomes very critical. In fact, people faced an orientation problem because of abundance of such information. Finding specific piece of information in this mass of data requires search engines to perform a remarkable task in providing users with subset of the original amount of information. Anyway, the subset retrieved by the search engines is still substantial in size. For example, at the time of writing the query "Text Summarization" in Google returned more than 4,000,000 results (as on 25th January 2014 from Google). Users still need to manually scan through each single item of the information retrieved by the web search engines until the information of user interest is obtained. This boring task makes automatic text summarization the task of great importance as the users can then just read the summary and get overview of the document. In another word, document retrieval is not sufficient and user need a second level of abstraction to reduce this huge amount of data, user should have text summarization technique. Text summarization is one of the basic techniques in the area of text mining. Text mining is to concern with the task of extracting relevant information, from natural language text, and to search for interesting relationships between the extracted entities [1]. In more specific, text summarization is the process of extracting the most important information from a single document / multi-documents and producing a new short version for a particular task and user without losing any important contents or overall meaning from the original document/documents. This process could be seen as a text compression; therefore, text summarization system should define the important parts based on the purpose of the summary or user needs. Text summarization techniques could be classified into two classes based on the way which summarization is going to perform on the input document/documents. Such classes are extractive and abstractive summarizations. The main objective of an extractive text summarization technique is to select the important sentences from the original input text and combine them into a new shorter version. The importance sentences selection process takes place based on linguistic features, mathematical and statistical techniques. The summary generated based on the important sentences from the original input text may not be coherent. But it gives main idea about the content of the input text. While the main idea behind an abstractive text summarization technique is to understand the original input text and then create summaries with its own words. The technique usually, depends on linguistic models to generate new sentences from the original sentences through a process called paraphrasing. The technique includes syntactic and semantic studies for specific language and is useful for meaningful applications. In fact, abstractive text summarization technique is similar to the way a human creates a summary; unfortunately this is still a challenging task for a computer program. As the matter of fact, there are increased demands in developing technologies for automatic Arabic text summarization [2,3]. Fortunately, there are several research projects to investigate and find out the techniques in automatically summarizing English documents as well as other European languages. Also, there is some of software products have been developed for English text summarization such as MEAD summarization toolkit. Unfortunately, there is a limitation in both research papers and software development in terms of automatic Arabic text summarization. The main objective of this paper is to describe results of Centroid-Based algorithm implementation [1]. It is used to capture sentence centrality based on some centrality measures such as degree and lexis ranking. Also, the paper presents a graph representation for clustering documents, where each node of the graph represents a sentence and each edge represents the similarity relation between pairs of sentences. The summarization algorithm is evaluated based on two types of documents that are AFP Arabic newswire corpus provided by LDC as well as summarization evaluations of Document Understanding Conference (DUC) [1].

2 Related Works

Over time there have been different methods and techniques to English text summarization and other European languages. Those methods and techniques are associated with single-document and multi-document summarization. Unfortunately, a few existing projects concerning with Arabic text summarization. The most closely related to this work are surveyed and reported:

A. Haboush et al. [1] presented and discussed a new model for automatic Arabic text summarization. They stated that the major attribute of their model is the word rooting capability. This attribute enabled the model to be semantic based rather than syntactic based. The meaning behind the root eliminated different derived structures. They reported in their conclusion that they obtained an average of recall (0.787) and precision (0.757) for the resulted summarization.

K. Thakkar and U. Shrawankar [4] suggested a model that uses text categorization and text summarization for searching a document based on user query. The model uses QDC algorithm for text categorization. The QDC algorithm is evaluated against other clustering algorithms. They stated that by using text summarization after searching the document they save the user's time required for reading the complete document.

P. Vijayapal Reddy et al. [5] investigated the problem of title word selection in the process of title generation for a given text document by using BMW approach. They stated that they tried to explore the impact of word weigh on Title word selection by using BMW model. They reported that they found F1 measure on Telugu corpus is 1.3 percent less than the F1 measure on English corpus due to Telugu has more complex morphological variations when compared with English.

V. Seretan [6] presented a novel approach to extractive summarization. The researcher reported that the method produced an abstract for an input document by selecting a subset of the original sentences. The researcher also mentioned that the method based on domain-specific collection. As well as collocation statistics are able to capture the gist of the information content in documents from a given domain, and by the fact that syntactically related co-occurrences represent a better way to model lexical meaning than surface co-occurrence. Finally, the researcher stated that the method has the ability to control the length and detail of the summary produced. Moreover, the work considered, in contrast, only syntactically related word combinations, thus eliminating the need for word sense disambiguation heuristics.

C.F. Greenbacker et al. [7] introduced an approach to automatic summarization of multimodal documents based on a semantic understanding of text and graphics. They stated that their model enabled them to construct a unified conceptual model that serves as the basis of generating an abstractive summary. They also added that they integrated the knowledge obtained from the graphic with the knowledge obtained from the text at the semantic level. They concluded that their method is able to generate summaries that are more human-like in nature, while not suffering from coherence and other readability issues related to traditional extractive techniques.

N. Nagwani and S. Verma [8] proposed a summarization algorithm that includes four phases: stop words elimination, frequent term computation, frequent term selection and semantic equivalent terms generation. They reported that all sentences in the document, which are containing the frequent and semantic equivalent terms, are filtered for summarization. They concluded that their experiment result was promising.

H. Yasin et al. [9] presented an automated Text Summarization System for multiple documents, it based on statistical factors. They stated that, Jacquard's coefficient was used to improve the worth and quality of the summarization. They also mentioned that their experiment was useful and effectual to enhance the quality of multiple documents summarization via Jacquard's coefficient. Finally, they concluded that the system represented steady correlation with the human assessment outcome.

Özsoy et al. [10] introduced the Latent Semantic Analysis (LSA) method for text summarization. They argued two LSA based summarization algorithm, also, they evaluated both algorithms on two different datasets. They concluded that, both of algorithms perform equally well on both Turkish and English datasets.

N. Zamin and A. Ghani [11] presented a hybrid approach to Malay text summarization. They indicated that the base system was built based on SUMMARIST and EstSum systems. They also emphasized that using a combination of two techniques enabled the base system to extract the most important sentences from Malay news articles.

H. Saggion [12] described a language independent multi document centroid-based summarization system. The system was evaluated in the 2011 TAC Multilingual Summarization pilot task where summaries were automatically produced for document clusters in Arabic, English, French and Hindi. The system had a good performance on Arabic and Hindi documents, a medium performance for English, and a poor performance for French.

J. Delort and E. Alfonseca [13] described the task of update summarization in TAC-2011, which consists of an extension of TOPICSUM. They reported that they have observed that the method performed comparably well for very short summaries in terms of ROUGE-2. Moreover, they executed TOPICSUM on the update set B as a baseline, they shown that it also performed better on shorter summaries.

A. Kogilavani and P. Balasubramani [14] proposed an approach to cluster multiple documents using clustering method. They produced cluster wise summary based on features profile oriented sentence extraction. They concluded that the generated summary coincides with the human summary for the same dataset of documents.

H. Saggion et al. [15] presented a series of experiment in content evaluation in text summarization. They reported that they found a weak correlation among different rankings in complex summarization tasks, such as summarization of biographical information and the summarization of the opinions about an entity.

G. Erkan and D. R. Radev [2] introduced a stochastic graph-based method for computing relative importance of textual units for Natural Language Processing. They evaluated the technique on the problem of text summarization. They stated that the results of applying this method on extractive summarization were quite promising. The main goal of the proposed paper is to investigate the efficiency of the Centroid-Based and Bayesian algorithms [2]. The Centroid -Based algorithm computes each sentence importance in a cluster and extracts the most important sentences to include in the text summary. The process of extraction and combination is based on the concepts of similarity matrix in sentences graph representation. While the Bayesian algorithm computes the vector of sentence features as well as the maximum probability of each sentence.

3 Centroid-Based and Bayesian Algorithms Specifications

In this section, both of Centroid -Based Algorithm (CBA) and Bayesian Classifier (BC) were proposed and implemented. On the one hand, the Centroid-Based algorithm performs the process of computing and combining the sentence centrality scores. Such a process is based on the presence of particular important words and similarity to a central sentence. Some measures were used for centrality such as degree and lexes rank. On the other hand, the Bayesian Classifier computes sentence attributes vector as well as the maximum probability of each sentence. The implementation details of both algorithms are reflected in sections 3 and 4 as follows:

3.1 Graph Representation

In a text the sentences are connected to each other. This connectivity can be realized as lexical overlap. In lexical connectivity, two sentences sharing same lexis are connected to each other. This concept is used to compute the sentence importance in a text. Since, a sentence importance in a text is associated with other sentences in the same text. Thus the graph is a suitable technique for representing the relationship and computing the relative importance of sentences by analyzing the graph structure. To implement this concept, the text should be represented as a fully connected graph G= (V, E). Where V is a set of a graph vertices and E is a set of a graph edges. In this study sentences used as the graph vertices at the same time the graph edges represent the lexical similarity between pairs of sentences. As soon as the fully connected graph is constructed the edge reduction algorithms can be used to reduce the graph to include only important edges. The most important edge reduction algorithm is the threshold algorithm. This algorithm eliminates an edge if its weight exceeds some thresholds [2].

3.1.1 Centrality of a Sentence

A Sentence centrality means the centrality of all words that it includes. The evaluation of award centrality is to search for the central of the document cluster in a vector space. The centroid of a cluster is a pseudo-document which contains words that have **tf×idf** scores greater than a predefined threshold [2].

3.1.2 Centroid -Based Summarization

The sentences that contain more words from the centroid of the cluster are called central as in Figure 1.

3.1.3 Sentence Salience Concept

A cluster of documents can be seen as a network of sentences which are connected to each other. Some sentences share a lot of information with each other while some others may share a little information with the rest of the sentences. Assume that the sentences which are similar to each other sentence in a cluster are more salient or central to the topic [2]. This concept is implemented in this experiment based on computing the similarity between two sentences as well as computing the overall prestigious of a sentence given its similarity to other sentences. The model bag of words is used to represent each sentence as an N-dimensional vector, where N is the number of all possible words in the target language. Cosine similarity measure is used to compute the similarity between two sentences as follows:

$$\cos\,(\mathrm{x},\,\mathrm{y}) = \frac{\sum_{\mathrm{w} \in \mathrm{s1,s2}} \mathrm{tf}_{\mathrm{w,x}} * \mathrm{tf}_{\mathrm{w,y}} (\mathrm{idf}_{\mathrm{w}})\ \ ^{2}}{\sqrt{\sum_{\mathrm{x_i} \in \mathrm{x}} (\mathrm{tf}_{\mathrm{x_i,x}} * \mathrm{idf}_{\mathrm{x_i}})^2} \sqrt{\sum_{\mathrm{y_i} \in \mathrm{y}} (\mathrm{tf}_{\mathrm{y_i,y}} * \mathrm{idf}_{\mathrm{y_i}})^2}} \qquad (1)$$

Where $\mathrm{tf}_{\mathrm{w,s}}$ is the frequency of the word w in the sentence s and $\mathrm{idf}_{\mathrm{w}}$ is the inverse document frequency.

A cosine similarity matrix is computed and used for a cluster representation, where each item in the matrix represents the similarity between the corresponding sentences pair. Figure 2, Figure 3 show the algorithms used to compute the vector length, the similarity and the centroid node. While Figure 4 show the algorithm that associate with sentences summary.

```
INPUT:  An array S of n sentences, cosine threshold t
OUTPUT: An array C of Centroid scores hashes   WordHash; Array C;
/* compute $df_i$ which is document frequency of term $i$= number of documents containing term $i$ Compute
$idf_i$  which is the inverse document frequency of term $i$, = $\log_2 (N/ df_i)$  where N is the total number of
documents*/
FOR =1 TO n DO
 BEGIN
   FOREACH word w of S[i]   DO
        WordHash{w}{“tfidf”} = WordHash{w}{“tfidf”} + idf{w};
  END-FOREACH
END-OF-FOR
/* construct the centroid of the cluster By taking the words that are above the hreshold*/
 FOREACH word w of WordHash DO
   IF   (WordHash{w}{“tfidf”} > t ) then
        WordHash{w}{“centroid”} = WordHash{w}{“tfidf”};
  END-IF
  ELSE
     WordHash{w}{“c centroid"”} = 0;
 End-IF
END-FOR
/* compute the score for each sentence */
FOR i =1 TO  n  DO
 BEGIN
    C[i] = 0;
   FOREACH word w OF S[i] DO
        C[i] = C[i] + WordHash{w}{“c centroid"”};
  END-FOREACH
END
return C;
```

Figure 1. Computing Centroid Scores Algorithm

```
int[] CentralityNodes(decimal[,] cosineMatrix, decimal threshold, int size)
      {
         int max = 0;
         cenNode = -1;
         cosineMatrixDeg = new decimal[size, size];
         int[] degree = new int[size];
         decimal[] lR = new decimal[size];
         for (int i = 0; i < size; i++)
         {
            for (int j = 0; j < size; j++)
            {
               if (cosineMatrix[i, j] > threshold)
               {
                  cosineMatrixDeg[i, j] = cosineMatrix[i, j];
                  degree[i]++;
               }
               else
                  cosineMatrixDeg[i, j] = 0;
            }
         }
         for (int i = 0; i < size; i++)
         {
            if (degree[i] > max)
            {
               max = degree[i];
               cenNode = i;
            }
                  }
        return summarizatinNodes(cosineMatrixDeg,degree,cenNode,size);
     }
```

Figure 2. Computing Central Node Degree Algorithm

This is a measure of how close the sentence is to the centroid of the cluster [2,3].

```
Vector_Length()
        {
            DB.conn.Open();
           countS = this.countSP;
           for (int i = 1; i <= countS; i++)
            {
              DB.da.SelectCommand.CommandText = "select term_w from weights where sen_no=" + i;
              DB.da.Fill(DB.ds);
              DB.dt = DB.ds.Tables[0];
              if (DB.dt.Rows.Count > 0)
               {
                 VecLength = 0;
                 for (int j = 0; j < DB.dt.Rows.Count; j++)
                  {
                    if ((decimal)DB.dt.Rows[j][0] > 0)
                      VecLength += Math.Pow(Convert.ToDouble((decimal)DB.dt.Rows[j][0]), 2);
                  }
              }
            VecLength = Math.Sqrt(VecLength);
            DB.da.InsertCommand.CommandText = "insert into vector values(" + i + "," + VecLength + ")";
            DB.da.InsertCommand.ExecuteNonQuery();
            DB.ds.Clear();
            DB.dt.Clear();
           }
          DB.conn.Close();
        }
        }
```

Figure 3. Computing Vector Length Algorithm

Table 1 shows the similarity matrix which represents a subset of a cluster used in Arab newswire 2004. The same matrix also is represented as a weighted graph where each link represents the cosine similarity between a pair of sentences Figure 4.

```
decimal[,] cosSimilarity()
 {
   cosineMatrix = new decimal[this.countSP, this.countSP];
   DataTable dt1 = new DataTable();
   DataSet ds1 = new DataSet();
   DataTable dt2 = new DataTable();
   DataSet ds2 = new DataSet();
   DB.da.SelectCommand.CommandText = "select vec_length from vector";
   DB.da.Fill(ds2);
   dt2 = ds2.Tables[0];
   for (int i = 1; i <= countS - 1; i++)
   {
     DB.da.SelectCommand.CommandText = "select term_no,term_w from weights where sen_no=" + i;
     // initial dataset & dataTable
     ds1.Clear();
     dt1.Clear();
     DB.da.Fill(ds1);
     dt1 = ds1.Tables[0];
     for (int j = i + 1; j <= countS; j++)
     {
       CosSim = 0;
       X_Y = 0;
       //DW for Di
       DB.da.SelectCommand.CommandText = "select term_w,term_no from weights where sen_no=" + j;
      // initial dataset & dataTable
       DB.ds.Clear();
       DB.dt.Clear();
       DB.da.Fill(DB.ds);
       DB.dt = DB.ds.Tables[0];
       if (DB.dt.Rows.Count > 0 && dt1.Rows.Count > 0)
        {
         for (int j0 = 0; j0 < DB.dt.Rows.Count; j0++)
          {
           for (int j1 = 0; j1 < dt1.Rows.Count; j1++)
            {
             if (IsTermEqual(term1,term2))
              X_Y += Convert.ToDouble((decimal)(DB.dt.Rows[j0][0]) * (decimal)(dt1.Rows[j1][1]));
            }
          }
        if (((decimal)dt2.Rows[i - 1][0] * (decimal)dt2.Rows[j - 1][0])) == 0)
           CosSim = 0;
        else
           CosSim = X_Y / Convert.ToDouble(((decimal)dt2.Rows[i - 1][0] * (decimal)dt2.Rows[j - 1][0]));
        cosineMatrix[i - 1, j - 1] = (decimal)CosSim;
        cosineMatrix[j - 1, i - 1] = (decimal)CosSim;
       }
     }
   }
 for (int i = 0; i < cosineMatrix.GetLength(0); i++)
     {
        for (int j = 0; j < cosineMatrix.GetLength(0); j++)
          if (i == j)
              cosineMatrix[i, j] = 1;
    }
  return cosineMatrix;
}
```

Figure 4. Computing Cosine Matrix Algorithm

Table 1. Intra-sentence cosine similarities in a subset of cluster from Arabic Newswire-a (2004)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1	1.00	0.08	0.04	0.07	0.05	0.03	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.05	0.02	0.05	0.04
2	0.08	1.00	0.04	0.08	0.16	0.13	0.02	0.00	0.01	0.01	0.00	0.01	0.00	0.06	0.19	0.05	0.04	0.13
3	0.04	0.04	1.00	0.02	0.05	0.03	0.48	0.00	0.05	0.02	0.01	0.00	0.01	0.02	0.12	0.06	0.00	0.01
4	0.07	0.08	0.02	1.00	0.45	0.07	0.00	0.00	0.04	0.00	0.00	0.01	0.00	0.00	0.09	0.05	0.02	0.08
5	0.05	0.16	0.05	0.45	1.00	0.08	0.00	0.06	0.02	0.00	0.03	0.01	0.00	0.16	0.22	0.12	0.03	0.12
6	0.03	0.13	0.03	0.07	0.08	1.00	0.02	0.11	0.00	0.00	0.04	0.00	0.00	0.00	0.02	0.07	0.00	0.00
7	0.03	0.02	0.48	0.00	0.00	0.02	1.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00
8	0.00	0.00	0.00	0.00	0.06	0.11	0.00	1.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.03	0.00	0.00
9	0.00	0.01	0.05	0.04	0.02	0.00	0.00	0.00	1.00	0.01	0.00	0.00	0.00	0.02	0.01	0.09	0.00	0.00
10	0.00	0.01	0.02	0.00	0.00	0.00	0.03	0.00	0.01	1.00	0.00	0.00	0.00	0.02	0.01	0.01	0.00	0.00
11	0.00	0.00	0.01	0.00	0.03	0.04	0.00	0.04	0.00	0.00	1.00	0.06	0.03	0.03	0.01	0.00	0.00	0.01
12	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.06	1.00	0.00	0.00	0.01	0.00	0.01	0.01
13	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	1.00	0.03	0.02	0.00	0.00	0.02
14	0.00	0.06	0.02	0.00	0.16	0.00	0.00	0.00	0.02	0.02	0.03	0.00	0.03	1.00	0.24	0.02	0.00	0.10
15	0.05	0.19	0.12	0.09	0.22	0.02	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.24	1.00	0.01	0.01	0.17
16	0.02	0.05	0.06	0.05	0.12	0.07	0.05	0.03	0.09	0.01	0.00	0.00	0.00	0.02	0.01	1.00	0.00	0.00
17	0.05	0.04	0.00	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	1.00	0.44
18	0.04	0.13	0.01	0.08	0.12	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.10	0.17	0.00	0.44	1.00

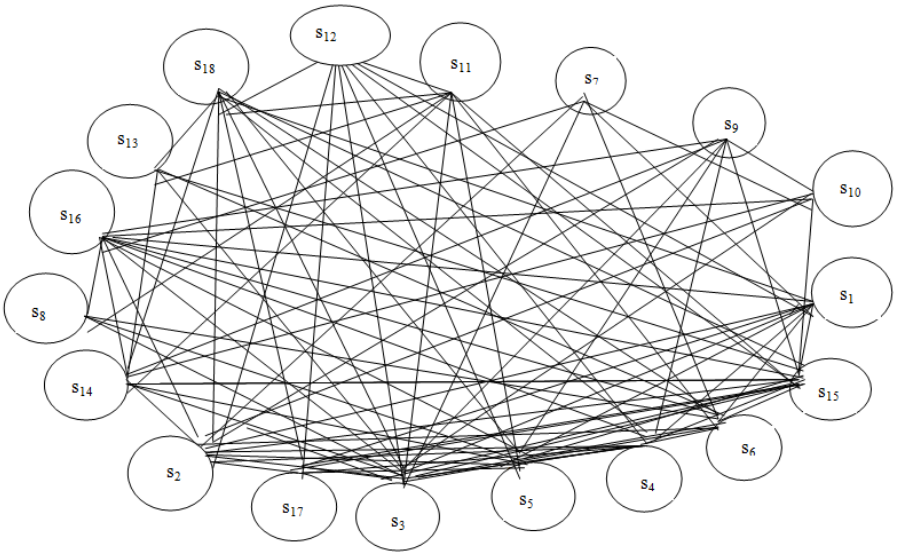

Figure 5. Cosine similarity graph for the cluster in Table 1.

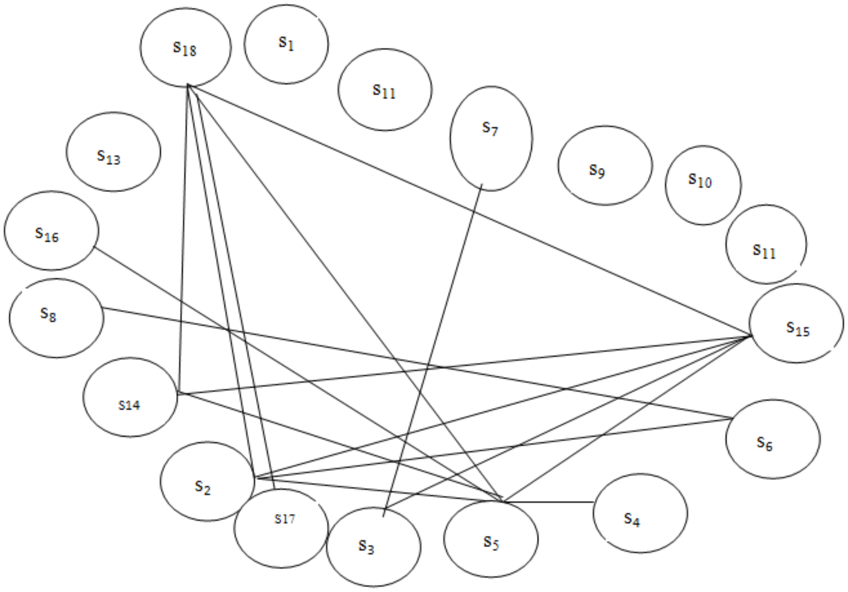

Figure 6. Similarity graphs which correspond to thresholds 0.1 for the cluster in Table 1.

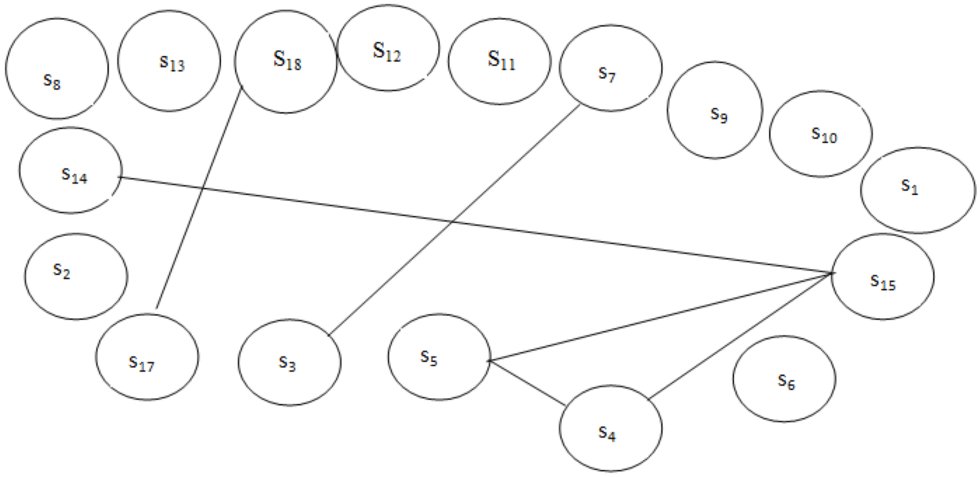

Figure 7. Similarity graph which correspond to thresholds 0.2 for the cluster in Table 1.

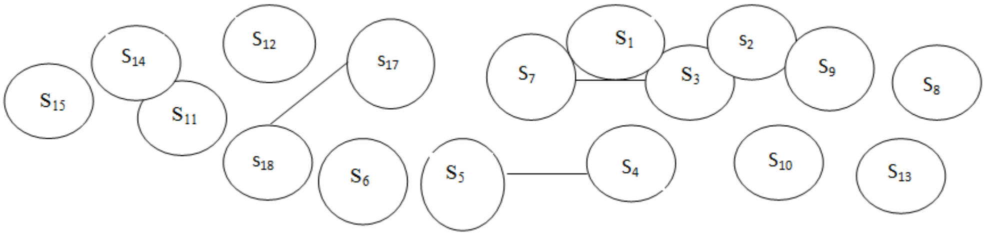

Figure 8. Similarity graph which correspond to thresholds 0.3 for the cluster in Table 1.

3.1.3 Degree Centrality

The degree centrality of a sentence is the degree of the corresponding node in the similarity graph Table1, shows the effect of cosine threshold selection. Too high thresholds may cause losing many of the similarity weights in a set of documents while too low thresholds may cause the weak similarity weights into consideration [16,17].

Table 2. Degree centrality scores for the graphs in Figure 3. Sentence s15 is the most central for thresholds 0.1 and 0.2

Id	Degree(0.1)	Degree(0.2)	Degree(0.3)
S_1	1	1	1
S_2	5	1	1
S_3	3	2	2
S_4	2	2	1
S_5	7	3	2
S_6	3	1	1
S_7	2	2	2
S_8	2	1	1
S_9	1	1	1
S_{10}	1	1	1
S_{11}	1	1	1
S_{12}	1	1	1
S_{13}	1	1	1
S_{14}	4	2	1
S_{15}	6	3	1
S_{16}	2	1	1
S_{17}	2	2	2
S_{18}	6	2	2

4. Naive Bayesian Classifier

A Bayesian classifier classifies each sentence to be in summary or out of summary classes based on its feature vector and the training data.

4.1 Sentences Representation

Each sentence is represented by a set of discriminative features. Such features are sentence –to sentence cohesion, position in the paragraph, sentence length, and number of infinitives in the sentence, title similarity, keyword similarity and proper noun occurrences [18,19]. Also, for each sentence the probability that will be included in summary can be computed as follows:

$$P(s \in S|V_1\,, V_2, \dots , V_n) = \frac{\mathrm{P}(\mathrm{V}_1,\mathrm{V}_2,\dots\mathrm{V}_\mathrm{n}|\mathrm{s}\in S)\mathrm{P}(\mathrm{s}\in S)}{\mathrm{P}(\mathrm{V}_1,\mathrm{V}_2,\dots\mathrm{V}_\mathrm{n})} \qquad (2)$$

Where **s** is the sentence, S is the Summary class, V is the features vector and n is the number of features [20]. Assuming that features are statistically independent:

$$P(s \in S|V_1, V_2, \dots V_n\,) = \frac{\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P}(\mathrm{V}_\mathrm{i}|\mathrm{s}\in S)\mathrm{P}(\mathrm{s}\in S)}{\prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}(\mathrm{V}_\mathrm{i})} \qquad (3)$$

The sentence is classified into summary class if the following condition is fulfilled:

$\prod_{i=1}^{n} \mathrm{P}(\mathrm{V}_\mathrm{i}|\mathrm{s} \in S)\mathrm{P}(\mathrm{s} \in S) > \prod_{i=1}^{n} \mathrm{P}(\mathrm{V}_\mathrm{i}|\mathrm{s} \in NS)\mathrm{P}(\mathrm{s} \in NS)$ (4), where NS is the non- summary class [21,22].

5 Dataset and Metrics

5.1 Test Collection

The test collection for both proposed algorithms (CBA and BC) is delivered by the Linguistic Data Consortium (LDC) at the University City of PENN USA. The LDC provides two Arabic collections, the Arabic GIGAWORD and the Arabic NEWSWIRE-a corpus [23]. The source documents contain meta-data and tags and are represented as UTF–8 files. The dataset contains 100 documents divided into 5 reference sets; each contains 20 related documents discussing the same topic [24].

5.2 Evaluation

Summary quality and consistency assessment is very difficult, because there is no objective summary. There are two types of summary measures: Form and Content measures. Form measures concern with assessment of text grammar, organization and coherence. Content measures concern with assessment of the percentage of information presented in the machine summary (precision) as well as the percentage of important information omitted from machine summary (recall). Also, there are automatic evaluation measures such as ROUGE. The assessment of both algorithms (CBA, BC) results was conducted manually and automatically. The manual assessment was based on the text overall responsiveness and the automatic assessment used ROUGE method. For the manual assessment, the human assessors were given the following instructions: Each summary is to be assigned an integer grade from 1 to 5 based on the overall responsiveness of the summary. A text should be assigned 5,if it covers the important aspects of the related documents including language fluency and readability. A text should be assign a 1, if it is either insensible, unreadable or contains very limited information from the related documents. The Length Aware Grading Measure (LAGM) was used to normalize the summaries which are out of limit. The (LAGM) is defined as$LAGM = \mathrm{g}(1 - \frac{\max(\max(\mathrm{lmin} - |s|, |\mathrm{s}| - \mathrm{lmax}), 0)}{\mathrm{lmin}})$ where g is a grade, l_{min} is the lower word limit count, l_{max} is the upper world limit count and |s| is the number of words in the summary. The automatic assessment was based on human created model summary. The summary model produced by the fluent speaker of Arabic language. The RUGE model variations were used [25,26].

It is a reasonable for an algorithm to behave similarly in the existence of bugs to the way it would behave without bugs. Thus, CBA and BC were tested and evaluated for robustness and the result was computed and recorded.

6 Experiment Results

In the resulting summary, on the one hand, all sentences were ranked based on similarity with respect to the centroid. The summary is produced by choosing sentences which are closed to the centroid until the desired bound is reached. A sentence very similar to the centroid appears within the resulting summary before the one is less similar to the centroid based on the algorithm output shown in Figure 9. This method gives a coherent summary in terms of processing a single cluster which is centered on specific theme. On the other hand, the Bayesian classifier was implemented, whereas the total size of the corpus was partitioned into training set 80% and testing set 20%, where the acceptable summary size was between 240 and 250 words.

```
int[] summarizatinNodes(decimal[,] cosineMatrix, int[] degree, int cenNode, int size)
     {
       decimal[] summSensNodeTempValue = new decimal[size];
       int[] summSensNodeTempIndex = new int[size];
       summSensNode = new int[summSensNo];
       for (int i = 0; i < size; i++)
        {
          if (cosineMatrix[cenNode, i] > 0)
            {
             summSensNodeTempValue[i] = cosineMatrix[cenNode, i];
             summSensNodeTempIndex[i] = i;
            }
          else
             summSensNodeTempIndex[i] = -1;
        }
       for (int i = 0; i < summSensNo; i++)
        {
          for (int j = i + 1; j < size; j++)
            {
              if (summSensNodeTempValue[i] < summSensNodeTempValue[j])
                {
                  tempValue = summSensNodeTempValue[i];
                 summSensNodeTempValue[i] = summSensNodeTempValue[j];
                 summSensNodeTempValue[j] = tempValue;
                 tempIndex = summSensNodeTempIndex[i];
                 summSensNodeTempIndex[i] = summSensNodeTempIndex[j];
                 summSensNodeTempIndex[j] = tempIndex;
                }
            }
        }
       for (int i = 0; i < summSensNo; i++)
         summSensNode[i] = summSensNodeTempIndex[i];
      return summSensNode;
     }
```

Figure 9. Summarization Algorithm

In this experiment, the above algorithms are trained on the dataset then the testing process of the resulted summary is conducted along with human summary for same documents. The CBA program starts reading 20 documents that represent reference $S_{1,}$ .The output machine summary of S_1 is compared with the human summary for the same 20 documents (S_1) using precession(P)and recall (R) measures. Both measures are computed by counting the common terms in both machine and human summaries and recording them as N_c for reference $S_{1\ .}$At the same time, the number of terms in machine summary for reference S1 is counted and recorded as N_m. Then the precession P is computed as $P=N_c/N_m$, In a similar way, the number of terms in human summary is computed and recorded as N_h then the Recall is computed as $R=N_c/N_h$ for reference S1[27]. The same process is repeated for other references (S_2 to S_5). Table 3 summaries the results. Also, the algorithm is tested for only one of the reliability factors, it is robustness. The freest bugs' versions of the algorithm code are executed and their recall and precision are recorded as shown in Table 3.

Then by intentionally injecting random bugs into the algorithm source code for generating CBA1, CBA2 from CBA, Those programs were executed and the results were shown in Table 4.

Table 3. Summary of CBA Precision and Recall for the data set

Summary id	Precision	Recall	F-Measures
s_1	0.8400	0.6287	0.7191
s_2	0.5769	0.5836	0.580
s_3	0.5476	0.5587	0.5531
s_4	0.4889	0.9712	0.9063
s_5	0.7782	0.9146	0.8409
Average	0.6463	0.7313	0.7199

Table 4. Precision and Recall of CBA faulty versions

Algorithms version	precession	recall
CBA1	0.573	0.411
CBA2	0.431	0.715
average	0.502	0.563

In the same way, the BC shown in Figure 10 is implemented, trained and tested on the same data set. The output results recorded in Table 5. Also, the algorithm is tested for the robustness and the output result is shown in Table 6.

```
Compute the prior probability of each class P($C_i$) (In- Summary or Out of-Summary)
For (i=1 to 2) do
    Compute P(V /$C_i$ );
Maximize P(V/Ci)P(Ci)
if P((V/in-summary=”yes”)P(in-summary=”yes”) >(V/in-summary=”no”)P(in-summary=”no”))
    The sentence S in the summary
else
    The sentence S out of summary
```

Figure 10. Bayesian Classifier

Table 5. BC Precision and Recall for the data set

Summary id	Precision	Recall	F-Measures
s_1	0.698	0.543	0.611
s_2	0.677	0.534	0.597
s_3	0.583	0.658	0.620
s_4	0.589	0.673	0.628
s_5	0.708	0.614	0.658
average	0.651	0.604	0.623

Table 6. Precision and recall of bc faulty versions

Algorithms version	Precession	Recall
BC1	0.210	0.262
BC2	0.373	0.420
average	0.292	0.341

7. Discussions

By implementing the evaluation measures indicated earlier, the total runs of the CBA was 5 times .Each run processes 20 documents related to specific theme. The result is shown in Table 3, Table 4, Table 7 and Table 8, respectively:

Table 7. Human overall and Human LAG responsive scores

Summary Id	Human overall	Human (LAG)
s_1	3.6500	3.6500
s_2	3.7000	3.5458
s_3	4.4500	4.4129
s_4	3.7500	3.7500
s_5	3.9000	3.9000

Table 8. Summary of ROUGE scores for the CBA on the data set.

Summary Id.	Rouge1	Rouge2	Rouge3	Overall
s_1	0.780	0.610	0.461	0.610
s_2	0.670	0.500	0.452	0.541
s_3	0.823	0.653	0.268	0.581
s_4	0.593	0.434	0.346	0.458
s_5	0.549	0.457	0.348	0.451

Table 7 shows the human grading as well as the length aware grading measure (LAG) for 5 different summaries produced by running the programs 5 times on the specified themes. The result in table 7 indicates that the CBA performs very well. Where the average of the Human grade is 3.89 at the same time; the average of LAG grade is 3.852. The CBA performs better than ID8 implemented in [15] as appears in Figure 11.

Table 3 illustrates the precision and the recall results. We observe that CBA performs very well. Where the average of the precision is 0.6463 and the average of the recall is 0.7313. The CBA performs better than both the ID8 and the algorithm implemented in [14] as appears in Figure 13. Table 8 illustrates the ROUGE results scores for CBA on the same data set which is provided by LDC [27]. Where ROUGE1, ROUGE2, ROUGE3 averages are 0.683, 0.5308 and 0.375. The results show that CBA performs very well. The CBA outperforms the centroid algorithm implemented in [27] as reflected in Figure 13.

At the same time, the result obtained by the Bayesian classifier was shown in Table 5 and Table 6.

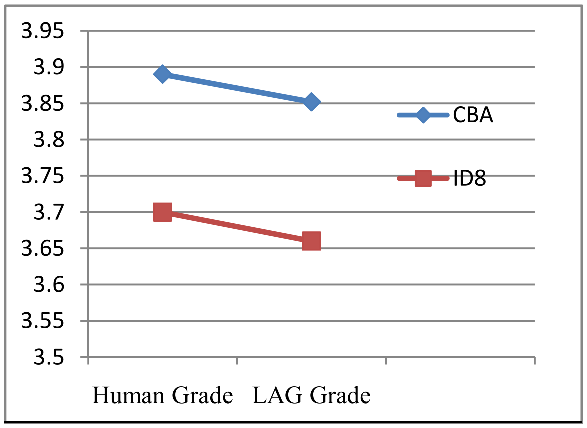

Figure 11. Manual assessment for CBA and ID8

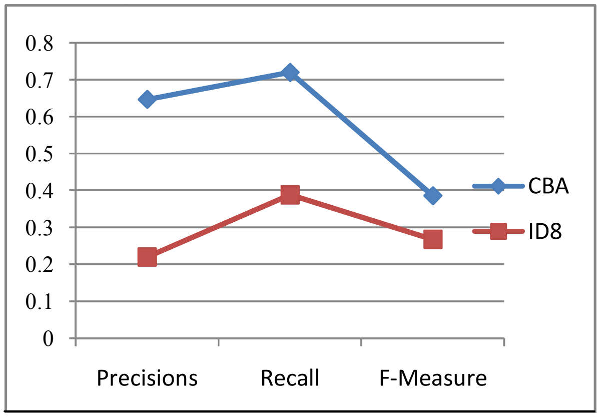

Figure 12. CBA performance along with ID8

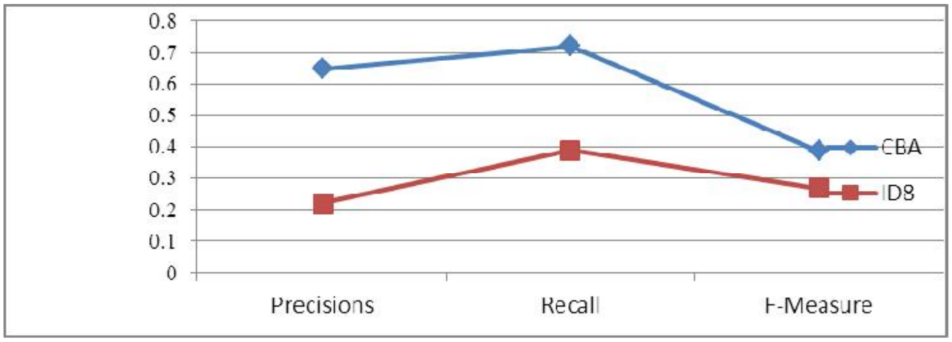

Figure 13. CBA and Centroid performance

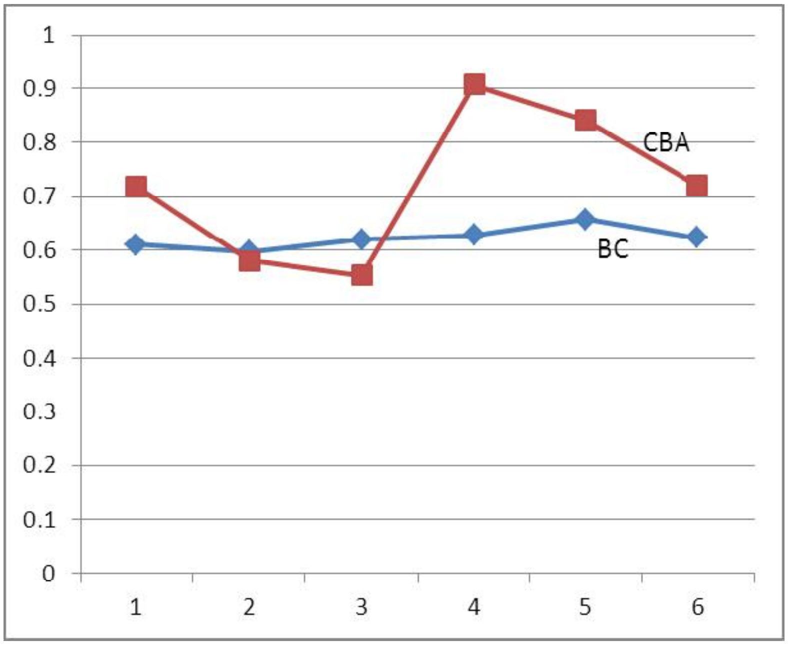

Figure 14. Centroid-Based (CBA) and Bayesian Classifier (BC) F-Measures

If the result which is tabulated in Table 5 is compared to the Centroid-Based algorithm result shown in Table 3, the new result will be obtained and depicted as shown in Figure 14.

The difference between the computed averages in Table 4 and Table 6 is 0.0969; the investigation of the level of significance is considered, then the average of differences is computed. Where the computed value of **t** is **t**=14.8 which is greater than $\mathbf{t}_{\alpha/2,\ 5=\ (}\mathbf{t}_{0.025,\ 5)}$ =2.571, thus t> $_{(}\mathbf{t}_{0.025,\ 5),}$ therefore, the averages difference is significance and CBA outperforms the BC [28].

On the one hand, the precession and recall averages of CBA versions (CBA1, CBA2) are 0.502 and 0.563 respectively. On the other hand, the precession and the recall averages of BC versions (BC1, BC2) are 0.2915 and 0.341 respectively. The average of the precision and recall between the results of CBA and its versions is 0.1443 and 0.168 respectively. At the time the average of the precision and recall between the results of BC and its versions is 0.359 and 0.263 respectively. The CBA result shown a low deviation average that means the CBA gives similar result either contains bugs or not compared to BC. Therefore, CBA is more robust than BC.

8 Conclusions

In this paper, Centroid-Based Algorithm (CBA) and Bayesian Classifier (BC) were used for Arabic Text Summarization. A software program that includes both GBA and BC algorithms is designed, implemented and tested. A real-world dataset was used for testing and validating the software summarizer performance, the result of the experiment was very promising. In this experiment, the CBA records high scores of F-Measures compared to the BC as indicated in

Figure 14, it outperforms the BC. Moreover, the CBA outperforms other summarizers used for Arabic text summarization, so far, such as ID8 and Centroid. On the one hand, the CBA in this experiment improved the responsiveness scores averages .It raised both the Human (Overall) and Human (LAG) from 3.70 and 3.66 respectively as reported in [15] up to 3.89 and 3.852 respectively as obtained in this task. The CBA raised the F-score from 0.26786 as reported in [15] up to 0.71988 as obtained in this experiment. On the other hand, the CBA raised ROUGE1 from 0.4443 as reported in [14] up to 0.683 as obtained in this experiment. That means the CBA outperforms ID8, Centroid and BC. The CBA results shown that, it has the ability to compress or reduce the text into 25% of its original size without losing the main concept behind the original text. This property enables the algorithm to be more distinguishable than other algorithm used for the same purpose. The CBA technique is robust compared to BC as indicated in section 6 and 7. It outperforms all those techniques which are used in Arabic text extractive summarization so far. In fact, human's text summarization based on the text understanding by humans themselves, unfortunately, none of the CBA and BC algorithms associates with text understanding so that is why there is a limitation in the algorithms performance. Therefore, the future work should deal with building semantic techniques such as ontologies. Building different ontologies based on some semantic rules that may include semantic concepts. Such concepts may combine both abstractive extractive Arabic text summarization .Implementing ontology in this aspect could be more efficient technique to obtain positive results. The comparative study among proposed ontology will take place in order to select the best algorithm with high performance.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Haboush A, Al-zoubi M, Momani A, Tarazi M. Arabic Text Summarization Model Using Clustering Techniques. In the World of Computer Science and Information Technology Journal (WCSIT). 2012;2(3):62–67.

[2] Violeta S. A Collocation-Driven Approach to Text Summarization". In the TALN 2011 Montpellier, 27juin – 1erjuillet 2011.

[3] Greenbacker CF, McCoy KF, Carberry S, McDonald D. Semantic Modeling of Multimodal Documents for Abstractive Summarization. In the Proceedings of the Workshop on Automatic Text Summarization Collocated with Canadian Conference on Artificial Intelligence, 2011, Canada.

[4] Nagwani N, Verma S. A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm. In the International Journal of Computer Applications (0975 – 8887) Volume 17– No.2, March 2011.

[5] Yasin H, Yasin M, Yasin F. Automated Multiple Related Documents Summarization viaJaccard's Coefficient. In the International Journal of Computer Applications (0975 – 8887), Volume 13– No.3, January 2011, Pakistan.

[6] Gülçin Ö, Alpaslan F, Çiçekli İ. Text summarization using latent semantic analysis ". Master thesis, Middle East Technical University, February 2011.

[7] Zamin N, Ghani A. Summarizing Malay Text Documents". In the World Applied Sciences Journal12 (Special Issue on Computer Application & knowledge management): 39-46, 2011, Malaysia.

[8] Saggion H. Using SUMMA for Language Independent Summarization at TAC 2011. In the proceeding of the TAC 2011 Workshop November, 2011, National Institute of Standards and Technology Gaithersburg, Maryland USA.

[9] Delort J, Alfonseca E. Description of the Google update summarizer at TAC-2011. In the proceeding of the TAC 2011 Workshop November, 2011, National Institute of Standards and Technology Gaithersburg, Maryland USA.

[10] Kogilavani A, Balasubramani P. Clustering and feature specific sentence extraction based summarization of multiple documents. International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010.

[11] Saggion H, Torres-Moreno J, da Cunha I, SanJuan E, Vel´ azquez-Morales P. Multilingual Summarization Evaluation without Human Models. In the Coling 2010: Poster Volume, pages 1059–1067, Beijing, August 2010.

[12] Erkan G, Radev R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. In the Journal of Artificial Intelligence Research. 2004;22:457-479.

[13] El-Haj, Mahmoud, Kruschwitz. Chris Fox "University of Essex at the TAC 2011 Multilingual Summarization Pilot.

[14] Thakkar K, Shrawankar U. Test Model for Text Categorization and Text Summarization. In the International Journal on Computer Science and Engineering (IJCSE). 2011;3(4), India.

[15] Vijayapal Reddy P, Vishnu vardhan B, Govardhan A. Analysis of BMW Model for Title Word Selection on Indic Script. In the International Journal of Computer Applications (0975 – 8887) Volume 18– No.8, March 2011.

[16] Haboush A, Momani A, Al-Zoubi M, Tarazi M. Arabic Text Summarization Model Using Clustering Techniques", World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 3, 62 – 67, 2012.

[17] Zamen N, Ghani A. Summarizing Malat Text Documents. World applied Science Journal 12 Computer Application and Management, 30-46,2011,SSN 1818-4952.

[18] Kogilavani A, Balasubramani P. Clustering and feature specific sentence extraction based Summarization of multiple documents. International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010, DOI: 10.5121/ijcsit.2010.2409 99.

[19] Perumal P, Nedunchezhian R. Performance Evaluation of Three Model-Based Documents Clustering Algorithms "European Journal of Scientific Research ISSN 1450-216X Vol.52 No.4 (2011), pp.618-628 © Euro Journals Publishing, Inc. 2011. Available: http://www.eurojournals.com/ejsr.htm.

[20] Thakkar K, Shrawankar U. Test Model for Text Categorization and Text Summarization ", International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397 Vol. 3 No. 4 Apr 2011.

[21] Yasin H, Yasin M, Yasin F. Automated Multiple Related Documents Summarization via Jaccard's Coefficient "International Journal of Computer Applications (0975 – 8887) Volume 13– No.3, January 2011.

[22] Nagwani N, Verma S. A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm. International Journal of Computer Applications (0975 – 8887) Volume 17– No.2, March 2011.

[23] Haboush A, Momani A, Al-Zoubi M, Tarazi M. Arabic Text Summarization Model Using Clustering Techniques. World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 3, 62 – 67, 2012.

[24] Reddy P, Mahendra R, vardhan B, Govardhan A. Analysis of BMW Model for Title Word Selection on Indic Script ",International Journal of Computer Applications (0975 – 8887) Volume 18– No.8, March 2011.

[25] He R, et al. Cascaded Regression Analysis Based Temporal Multi-document Summarization. Informatics. 2010; 34:119–124.

[26] LDC. Web site, https://www.ldc.upenn.edu/language-resources

[27] Murray R. Spiegel. PhD. Statistics Theories and Problems. McGraw-Hill International Book Company, Singapore; 1980.

[28] Alexander G, David DL, David M ADIGAN. ENKIN Large-Scale Bayesian Logistic Regression for Text Categorization Techno metrics. 2007;49(3):291-304.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=487&id=6&aid=4230