LAUTECH Journal of Engineering and Technology 12 (2) 2018: 96-106

APRIORI ALGORITHM APPROACH FOR AUTOMATIC TEXT PROCESSING AND GENERIC-BASED SUMMARIZATION SYSTEM

[1]Osang, F. B. and [2]Imeh U.

[1] Department of Computer Science, National Open University of Nigeria, Abuja, Nigeria
[2] Department of Computer Science, Akwa Ibom State University, PMB 1167, Mkpat Enin, Nigeria

*Corresponding author: imehumoren@ksu.edu.ng, +234 7057129566

ABSTRACT

Text Processing has always existed in various forms. It makes voluminous text easily digestible, offers brief and quick overview of the subject contents and may provide critical context analysis to the reader. With the growth of digital articles in forms of news, blogs, wikis etc., there is serious need for a text processor which can adequately summarized an article or documents for the reader. This redirected and takes away the effort needed to read, assimilate and create summaries manually. This research paper proposed a system which provides unique opportunity for developing a core set text summarization system using Apriori Algorithm techniques to perform Binary Associated Rule on Data Mining. The system makes available a means of storing the automatic Generic-based summaries for future references and requirements.

Keywords: Apriori Algorithm techniques, Text Processing, data summarization

1.0 Introduction

Camargo et al (2010) considered automatic data summarization as part of machine learning and data mining, stating that the main idea of summarization is to find a subset of data which contains the "information" of the entire set. Those techniques are widely used in industry today. Search engines are an example; others include summarization of news items, image collections and videos. Article summarization tries to create a representative summary or summary of the entire article, by finding the most informative sentences, while in image summarization, the system finds the most representative and important (i.e. salient) images. For surveillance videos, one might wish to extract the important events from the uneventful context (Camargo et al, 2010). Automatic summarization is the process of shortening a text article with software, in order to create a summary with the major points of the original article. Technologies that can make a coherent summary will consider variables such as length, writing style and syntax (Abderrafih, 2010).

There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or summary of the collection (whether articles, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs (Camargo et al, 2010).

An example of a summarization problem is article summarization, which attempts to automatically produce a summary from a given article. Sometimes one might be interested in generating a summary from a single source article, while others can use multiple source articles (for example, a cluster of articles on the same topic). This problem is called multi-article summarization. A related application is summarizing news articles, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary (Camargo et al, 2010).

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images. A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured. (Camargo et al, 2010).

At a very high level, summarization algorithms try to find subsets of objects (like set of sentences, or a set of images), which cover information of the entire set. This is also called the core-set. These algorithms model notions like diversity, coverage, information and representativeness of the summary. Query based summarization techniques, additionally model for relevance of the summary with the query. Some techniques and algorithms which naturally model summarization problems are TextRank and PageRank, Submodular set function, Determinant point process, maximal marginal relevance (Camargo et al, 2010).

1.2 Statement of the Problem

The challenges of manually reading and summarizing articles cannot be overemphasized. Most often articles are treated in their thousands, especially in the education circles where academic materials have to be read and scanned through severally in order to understand the context of the materials. Certain factors responsible for making the process of manual processing such a difficult ordeal are:

- Reading through a whole article and sorting out the essential points from it requires a lot of time and effort
- A lot of man power is required to efficiently read and separate important extracts out from an article and can lead to high expenditure by the organization or body handing the processing of the articles
- Employment of a few workers to handle hundreds of articles can lead to errors in processing of the work or a delay in the completion time of the work

2.0 Review of Related Literature

Early experimentation in the late 1950's and early 60's suggested that text summarization by computer was feasible though not straightforward (Luhn, 1959; Edmundson, 1968). The methods developed then were fairly unsophisticated, relying primarily on surface level phenomena such as sentence position and word frequency counts, and focused on producing extracts (passages selected from the text, reproduced verbatim) rather than abstracts (interpreted portions of the text, newly generated), (Hovy, et al. 2005). Automatic text summarization gained attraction as early as the 1950s. (Luhn et al., 1958) introduced a method to extract salient sentences from the text using features such as word and phrase frequency. They proposed to weight the sentences of a document as a function of high frequency words, ignoring very high frequency common words. (Edmundson et al. 1969) described a paradigm based on key phrases which in addition to standard frequency depending weights used the following three methods to determine the sentence weight:

- **Cue Method:** The relevance of a sentence is calculated based on the presence or absence of certain cue words in the cue dictionary.
- **Title Method:** The weight of a sentence is computed as the sum of all the content words appearing in the title and headings of a text.
- **Location Method:** This method assumes that sentences appearing in the beginning of document as well as the beginning of individual paragraphs have a higher probability of being relevant.

After some decades, with the growing popularity of the internet and the immense amount of documents online, then the need to have concise summaries arose. During these intervening decades, progress in Natural Language Processing (NLP), coupled with great increases of computer memory and speed, made possible more sophisticated techniques, with very encouraging results. In the late 1990's, some relatively small research investments in the US (not more than 10 projects, including commercial efforts at Microsoft, Lexis-Nexis, Oracle, SRA, and Text Wise, and university efforts at CMU, NMSU, UPenn, and USC/ISI) over three or four years have produced several systems that exhibit potential marketability, as well as several innovations that promise continued improvement. (Hovy, 2005).

However, to produce a summary automatically is very challenging. Issues such as redundancy, temporal dimension, co-reference or sentence ordering, to name a few, have to be taken into consideration especially when summarizing a set of documents (multi-document summarization), thus making this field even more difficult (Goldstein et al. 2000). Moreover, research attempting to overcome the lack of coherence that summaries often present has been fuelled in the last years, resulting in combined approaches that identify relevant content and merge it into new fragments of information (Barzilay and McKeown 2005, Zajic et al. 2008).

In other contexts, Text Summarization techniques and approaches have been used for solving specific tasks. For instance, Balahur and Montoyo (2008) used opinion mining techniques for extracting opinion features from customer reviews and then summarizing them.

Sauper and Barzilay, (2009) proposed an automatic method to generate Wikipedia articles, where specific topic templates, as well as the information to select are learnt using machine learning algorithms. The templates are obtained by means of recurrent patterns for each type of document and domain. For extracting the relevant content, candidate fragments are ranked

according to how representative they are with respect to each topic of the template.

Other approaches that also rely on the use of templates to organize and structure the information previously identified, are based on information extraction systems. In (Kumar et al., 2009), reports of events are generated from the information of different domains (biomedical, sports, etc.) that is stored in databases. In such research, human-written abstracts are used, on the one hand, to determine the information to include in a summary, and on the other hand, to generate templates. Then, the patterns to fill these templates in are identified in the source texts.

Natural Language Generation (NLG) has been also applied for adding new vocabulary and language structures in summaries. In (Yu et al., 2007) very short summaries are produced from large collections of numerical data. The data is presented in the form of tables, and new text is generated for describing the facts that such data represent. (Belz, 2008) also suggests a Text summarization approach based on NLG, in order to generate weather, forecast reports automatically.

Another interesting approach is to use citations from articles. In (Kan et al., 2002) it was shown that from bibliographic entries it was possible to produce an indicative summary. The main idea behind this assumption is that such entries contain informative as well as indicative information, for example, details about the resource or metadata, such as author or purpose of the paper. In their research, a big annotated corpus (2000 annotated entries) is developed for such purposes. Following the idea of generating summaries from this input information, in (Qazvinian and Radev, 2008) citations are analysed to produce a single-document summary from scientific articles. The final objective is to generate summaries about a specific topic.

3.0 Methodology

i) Useful Concepts

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

ii) Support

The support supp(X) of an item set X is defined as the proportion of transactions in the data set which contain the item set.

supp(X)= no. of transactions which contain the item set X / total no. of transactions

In the example database, the item set {milk,bread,butter} has a support of 4 /15 = 0.26 since it occurs in 26% of all transactions.

To be even more explicit we can point out that 4 is the number of transactions from the database which contain the item set {milk,bread,butter} while 15 represents the total number of transactions.

iii) Confidence

The confidence of a rule is defined:

$$conf(X \rightarrow Y) = supp(X \cup Y)/sup$$

1.0

For the rule {milk,bread}=>{butter} we have the following confidence:

supp({milk,bread,butter}) / supp({milk,bread}) = 0.26 / 0.4 = 0.65

This means that for 65% of the transactions containing milk and bread the rule is correct. Confidence can be interpreted as an estimate of the probability **P(Y | X)**, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

iv) Lift

The **lift** of a rule is defined as:

$$lift(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(Y) * supp(X)}$$

1.1

The rule {milk,bread}=>{butter} has the following lift:

supp({milk,bread,butter}) /supp({butter}) x supp({milk,bread})= 0.26/0.46 x 0.4= 1.4

v) Conviction

The **conviction** of a rule is defined as:

$$conv(X \rightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \rightarrow Y)}$$

1.2

The rule {milk,bread}=>{butter} has the following conviction:

1 – supp({butter})/ 1- conf({milk,bread}=>{butter}) = 1-0.46/1-0.65 = 1.54

The conviction of the rule X=>Y can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions.

In this example, the conviction value of 1.54 shows that the rule {milk,bread}=>{butter} would be incorrect 54% more often (1.54 times as often) if the association between X and Y was purely random chance.

4.1 General Process

Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent item sets in a database.
2. Second, these frequent item sets and the minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step needs more attention. Finding all frequent item sets in a database is difficult since it involves searching all possible item sets (item combinations). The set of possible item sets is the power set over ***I*** and has size $2n - 1$ (excluding the empty set which is not a valid item set). Although the size of the power set grows exponentially in the number of items n in ***I***, efficient search is possible using the **downward-closure property** of support (also called anti-monotonicity) which guarantees that for a frequent item set, all its subsets are also frequent and thus for an infrequent item set, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent item sets.

4.2 Text Data Mining Algorithm

According to Rashmi (2017) data mining, also known as Knowledge Discovery in Databases (KDD), to find anomalies, correlations, patterns, and trends to predict outcomes.

Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent item-sets and relevant association rules. It is devised to operate on a database containing a lot of transactions, for instance, items brought by customers in a store.

It is very important for effective Market Basket Analysis and it helps the customers in purchasing their items with more ease which increases the sales of the markets. It has also been used in the field of healthcare for the detection of adverse drug reactions. It produces association rules that indicates what all combinations of medications and patient characteristics lead to ADRs (Rashmi, 2017).

Vithlani (2012) explains that the Apriori classic algorithm used in data mining for learning association rules stating that learning association rules basically means finding the items that are purchased together more frequently than others. An example of learning associations' rules-based applications is the Google auto-complete, where after you type in a word it searches frequently associated words that user type after that particular word.

4.3 Application of Apriori Algorithm

i. **Application of the Apriori algorithm for adverse drug reaction detection**: The objective is to use the Apriori association analysis algorithm for the detection of adverse drug reactions (ADR) in health care data. The Apriori algorithm is used to perform association analysis on the characteristics of patients, the drugs they are taking, their primary diagnosis, co-morbid conditions, and the ADRs or adverse events (AE) they experience. This analysis produces association rules that indicate what combinations of medications and patient characteristics lead to ADRs.

ii. **Application of Apriori Algorithm in Oracle Bone Inscription Explication**: Oracle Bone Inscription (OBI) is one of the oldest writing in the world, but of all 6000 words found till now there are only about 1500 words that can be explicated explicitly. So explication for OBI is a key and open problem in this field. Exploring the correlation between the OBI words by Association Rules algorithm can aid in the research of explication for OBI. Firstly, the OBI data extracted from the OBI corpus are pre-processed; with these processed data as input for Apriori algorithm we get the frequent item set. And combined by the interestingness measurement the strong association rules between OBI words are produced. Experimental results on the OBI corpus demonstrate that this proposed method is feasible and effective in finding semantic correlation for OBI.

4.3.1 Apriori Algorithm Pseudocode

procedure **Apriori** (T, *minSupport*) { //T is the database and *minSupport* is the minimum support L1= {frequent items};

for (k= 2; L_{k-1} !=∅; k++) **{**

C_k= candidates generated from L_{k-1}

//that iscartesian product L_{k-1} x L_{k-1} and eliminating any k-1 size item set that is not frequent

for each transaction **t** in database **do{**

#increment the count of all candidates in C_k that are contained in t

L_k = candidates in Ck with *minSupport*

}//end for each

}//end for
return $U_k L_k$;
}

As is common in association rule mining, given a set of item sets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the item sets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.
Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets
among the candidates.
Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all $2^{|S|} - 1$ of its proper subsets.
Another example of Apriori Algorithm is explained further. Consider the Table 3.2 showing the sentences in a document

Table 4.1 Document item set

Sentence	Value
Sentence 1	This research is about cats and dogs
Sentence 2	We will talk about dogs and cats
Sentence 3	Lions are big cats
Sentence 4	Wolfs are ancestors of dogs

4.3.2 Using the Apriori Algorithm we will be find the most frequent words pairs in the sentences. The sequence of the algorithm can be defined as follows:

i) Get the items (words) to be sorted.
ii) Set an arbitrary value s that will indicate maximum frequency size (In this example s=2).
iii) Start Pass 1 through the items.
iv) After the Pass 1, is completed, check the count for each item.
v) If the count of item is more than or equal to s i.e. ***Count*** *(item* ***i****) >= s, then the item* ***i*** *is frequent. Save this for next pass.*
vi) After Pass 2 ends, check for the count of each pair of items.
vii) If more than or equal to s, the pair is considered to be frequent, i.e. Count (item ***i****, item j) >= s.*

Using the steps of the algorithm defined above, we have the following result after making the first pass shown in the Table 3.3.

Table 4.2 Word/item frequency table

Item (word)	Frequency
Are	2
Ancestors	1
And	2
About	2
Big	1
Book	1
Cats	3
Dogs	3
Lions	1
Is	1
Of	1
This	1

Talk	1
We	1
Will	1
Wolfs	1

Now we get the most frequent words that is above the value of ***s*** which in this example is 2, and show them in the Table 3.4.

Table 4.3 Frequent Word/item frequency table

Item (word)	**Frequency**
Cat	3
Dog	3
Are	2
And	2
About	2

Then we combine these frequent word terms into pairs as shown in the table below and show them in the Table 4.4

Item / Word Pairs
Cat Dog
Cat Are
Cat And
Cat About
Dog Are
Dog And
Dog About
Are And
Are About
And About

Table 4.5 Frequent Word/item Pairs table

At this point we count the frequency of the frequent word pairs in the original document sentences in Table 3.5.

Table 4.6 Frequent Word/item pair frequency table

Item (word)	**Frequency**
Cats Dog	2
Cats Are	1
Cats And	2
Cat About	1
Dog Are	1
Dog And	2
Dog About	1
Are And	0
Are About	0
And About	1

We then take the most frequently occurring pairs (i.e. we will take frequencies equal to or greater than *s* which is = 2) as shown in the Table 4.7.

Table 4.7 Paired Frequent Word/item pair frequency table

Item (word)	Frequency
Cats Dog	2
Cats And	2
Dog And	2

Thus, the sentences having the most combination of the frequent pairs of words are shown in Table 4.8:

Table 4.8 Most Frequent Word/item pair frequency table result

Sentence	Value	Paired Words	Paired Word Frequency
Sentence 1	This book is about cats and dogs	(Cat Dog) (Cats And) Dogs And)	3
Sentence 2	We will talk about dogs and cats	(Cat Dog) (Cats And) Dogs And)	3
Sentence 3	Lions are big cats	-	0
Sentence 4	Wolfs are ancestors of dogs	-	0

From the result of the Table 4.8 it can be seen that the sentences with the most paired word frequencies are the sentence 1 and sentence 2.
Thus, these two sentences will be used as the sentences that will make up the abstract as they contain most of the words that describe the concept of the document. Thus, our abstract will become the combination of sentence 1 and sentence 2: "This book is about cats and dogs. We will talk about dogs and cats"

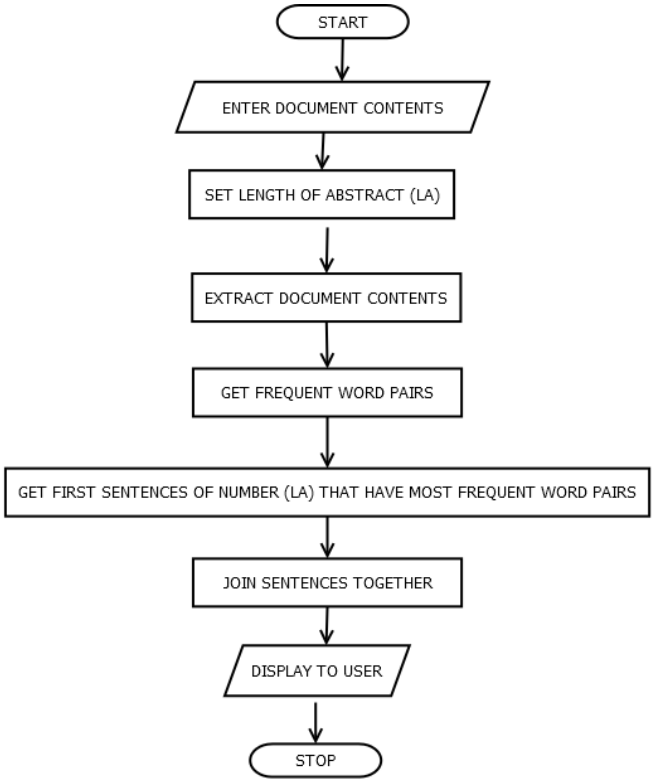

Fig. 4.1 Flowchart for Automatic Text Processing and Generic-Based Summarization System

5.0 SYSTEM IMPLEMENTATION and EVALUATION

The Document Abstract Extraction System modules are illustrated in figure 5.1a,

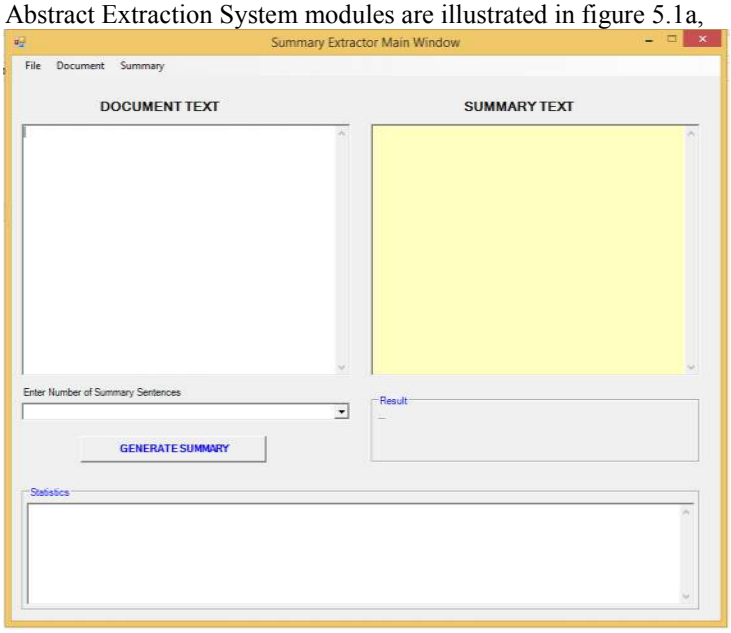

Fig 5.1a Main Program Module

The results of the evaluation of the document abstract extraction system are shown in this section of the chapter.

A) **Document Entry:** The figure below shows the document entry into the document entry textbox as shown in the figure 5.2.

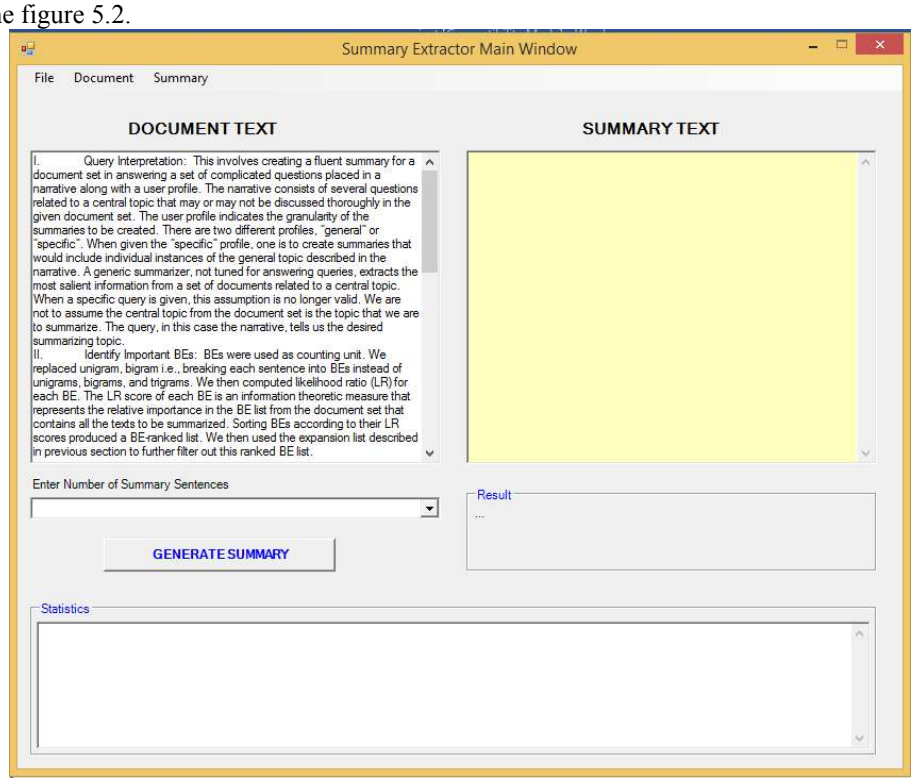

Figure 5.2 Document Entry

Document Extraction (Summary Sentence set to 3): The abstract extraction was done using the value of 3 for the number of sentences in the abstract. The result is shown in the Abstract Textbox portion of the Main Module screen.

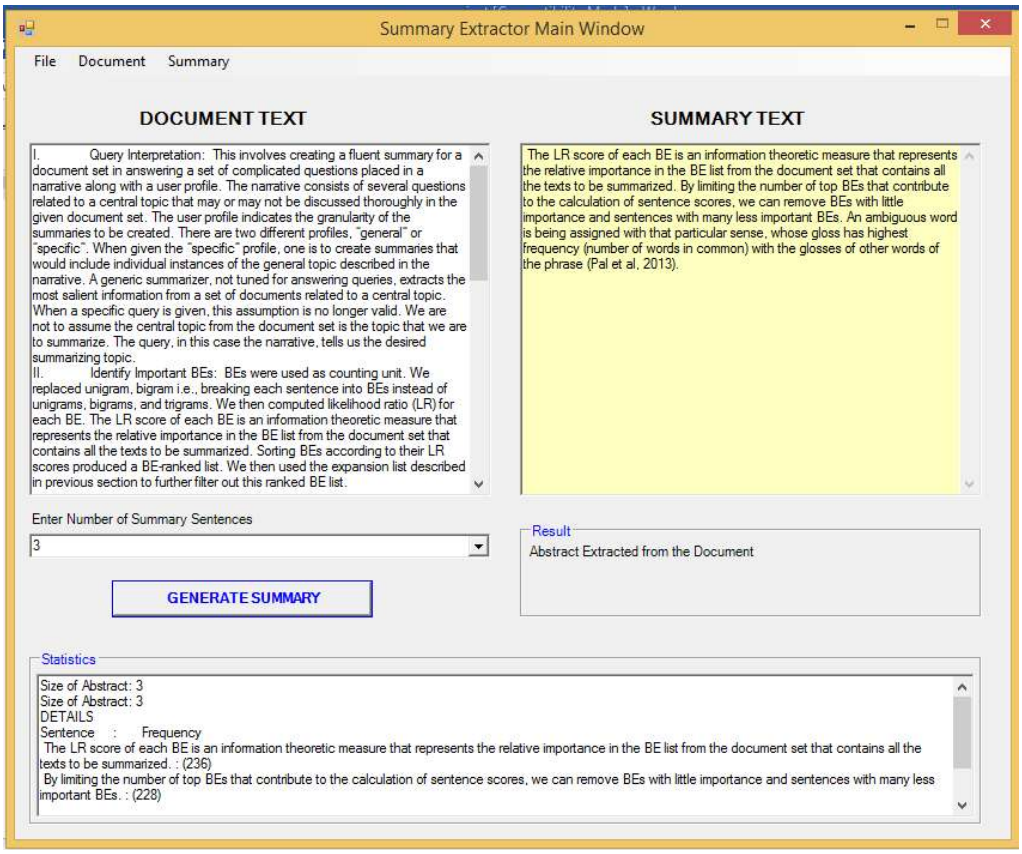

Fig 5.3 Document Extraction (Summary Sentence set to 3):

Summary Analysis: The summary analysis shows the size of the generated summary, its details including the sentence and their frequency delimited with colons. The image of the summary analysis of a document with the summary sentence set to 3 is shown in Figure 5.4.

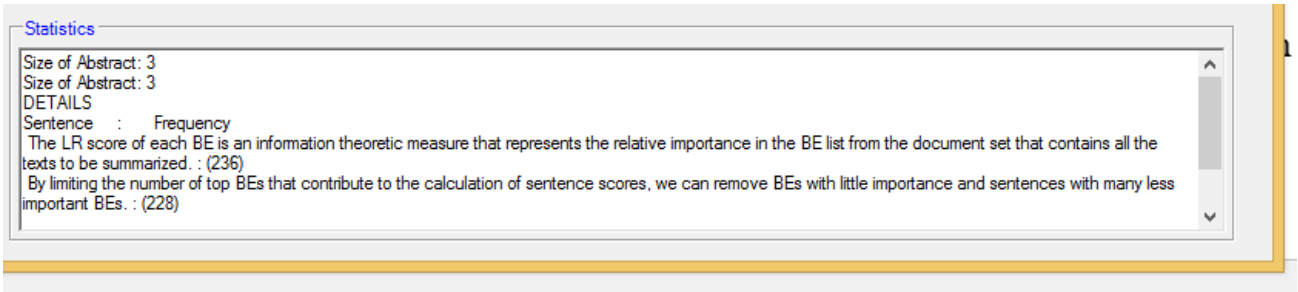

Fig. 5.4 Summary text analysis

Document Extraction (Abstract Sentence set to 5): The summary extraction was done using the value of 5 for the number of sentences in the summary. The result is shown in the Summary Textbox portion of the Main Module screen.

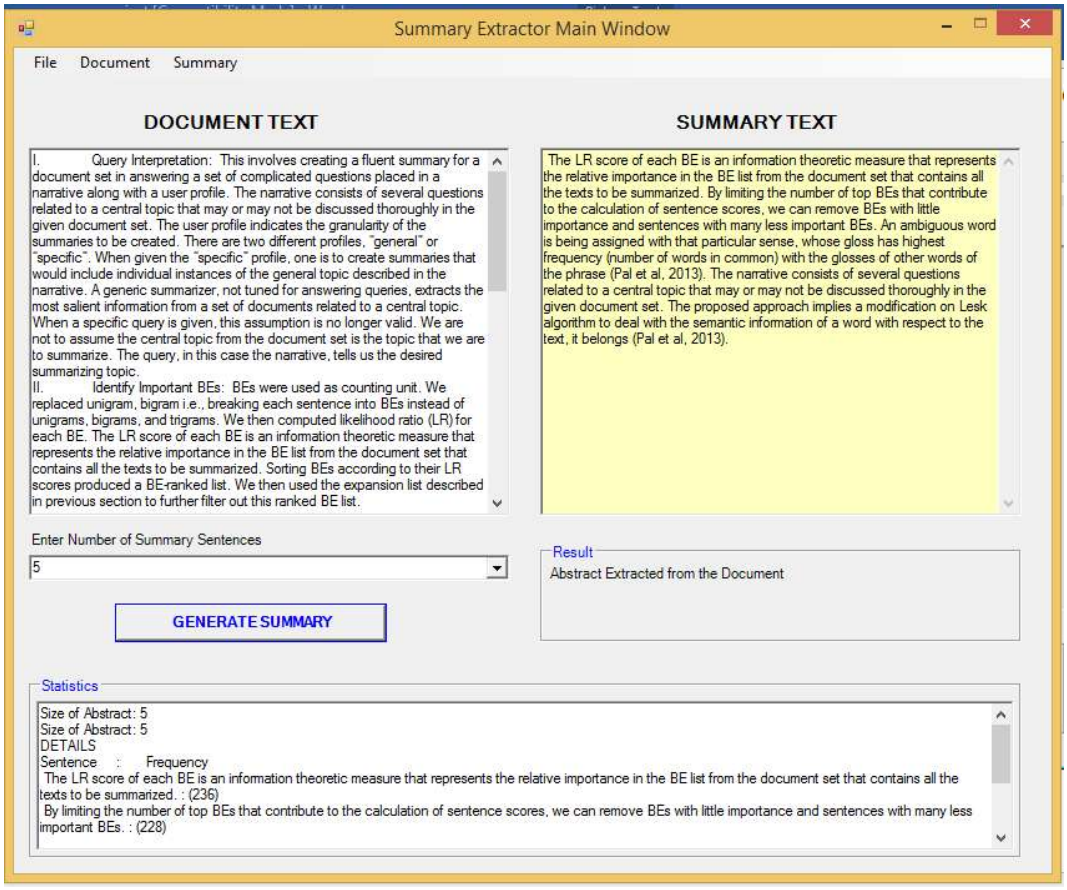

Fig. 5.5 Document Extraction (Summary Sentence set to 5):

6.0 Conclusion

The paper aimed at providing a document extraction software system for the summarization of contents in any documentation. This was achieved via the use of a data mining algorithm used to weigh the best combination of words and sentences that contains most of the vital concepts of the documents. The system uses the combination of Visual Basic.NET and the Access relational database system.

References

Alok Ranjan Pal, Projjwal Kumar Maiti, Diganta Saha (2013). "*An Approach to Automatic Text Summarization Using Simplified Lesk Algorithm and Wordnet* ". International Journal of Control Theory and Computer Modeling (IJCTCM) Vol.3, No.4/5

Goldstein J, Mittal V, Carbonell J, Kantrowitz M (2000). "Multi-document summarization by sentence extraction". In: NAACL-ANLP 2000 workshop on automatic Summarization. pp. 40–48

Hovy E, Lin CY, Zhou L, Fukumoto J (2006. "Automated summarization evaluation with basic elements". In: Proceedings of the 5th International Conference on Language Resources And Evaluation (LREC)

http://www.prajval.in/edudetail/74/302/<p>What-are-the-limitations-of-apriori-approach-for-mining (20 sept 2018)

Jain Rashmi (2017). "A beginner's tutorial on the apriori algorithm in data mining with R implementation". HackerEarth. [online]. Available from <http://blog.hackerearth.com/beginners-tutorial-apriori-algorithm-data-mining-r-implementation> (24 March 2017)

Jorge E. Camargo and Fabio A. González (2010). "*A Multi-class Kernel Alignment Method for Image Collection Summarization.* In Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP '09),". Springer-Verlag, Berlin, Heidelberg, 545-552. doi:10.1007/978-3-642-10268-4_64

Lehmam, Abderrafih (2010). "*Essential summarizer: innovative automatic text summarization*

software in twenty languages" ACM Digital Library. Published in Proceeding RIAO'10 Adaptivity, Personalization and Fusion of Heterogeneous Information, CID Paris, France

Liang Zhou, Chin-Yew Lin (2016). "*A BE-based Multi-Document Summarizer with Query Interpretation*". Information Sciences Institute University of Southern California. Available From https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/duc_2005.pdf

Liadh Kelly, Johannes Leveling, Shane McQuillan, Sascha Kriewel, Lorraine Goeuriot, Gareth Jones (2013) "*Report on summarization techniques* ". [online]. Information and Communication Technologies (ICT) Theme of the 7th Framework Programme for Research and Technological Development.

Lloret, Palomar (2011) "Text summarization in progress: a literature review" Springer Science+Business Media B.V. 2011, DOI 10.1007/s10462-011-9216-z

Lloret, Palomar et. al (2013) "COMPENDIUM: A text summarization system for generating abstracts of research papers", Elsevier B. V.

M. Wells (2009) "3 *Advantages of Automatic Text Summarization*". Ezine Articles. [online] Available from <http://ezinearticles.com/?3-Advantages-of-Automatic-Text-Summarization&id=3465270> (22 December 2009)

Nikhil Vithlani (2012) "*Apriori algorithm for Data Mining – made simple*". BlogSpot.

S.A. Babar, Pallavi D. Patil (2014) *"Improving Performance of Text Summarization"*. International Conference on Information and Communication Technologies (ICICT 2014), Elsevier 2015

S.A. Thorat, S.A. Babar (2014) *"Improving Text Summarization using Fuzzy Logic & Latent Semantic Analysis"*. International Journal of Innovative Research in Advanced Engineering (IJIRAE) Volume 1 Issue 4

Vishal Gupta, Gurpreet Singh Lehal (2010). "*A survey of Text summarization techniques"*, Journal of Emerging Technologies in Web Intelligence VOL 2 NO 3; August 2010.