ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 7 Issue 2 February 2018, Page No. 23526-23530
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i2.02

CONDENZA:A System for Extracting Abstract from a Given Source Document

Mgbeafulike .I .J[1] & Ejiofor C. I.[2]

Department of Computer Science Chukwuemeka Odumegwu Ojukwu University[1,2]

Abstract
Despite the increasingly availability of documents in electronic form and the availability of desktop publishing software, abstracts continue to be produced manually. The purpose of CONDENZA is to develop a system for abstract extraction from a given source document. CONDENZA describes a system on automatic methods of obtaining abstracts. The rationale of abstracts is to facilitate quick and accurate identification of the topic of published papers. The idea is to save a prospective reader time and effort in finding useful information in a given article or report. The system generates a shorter version of a given sentence while attempting to preserve its meaning. This task is carried out using summarization techniques. CONDENZA implements a method that combines apriori algorithm for keyword frequency detection with clustering based approach for grouping similar sentences together. The result from the system shows that our approach helps in summarizing the text documents efficiently by avoiding redundancy among the words in the document and ensures highest relevance to the input text. The guiding factors of our results are the ratio of input to output sentences after summarization.

Keywords— summarization, text, document, apriori algorithm, clustering algorithm, search, feature extraction, Sentence Score, Word count, cluster, Frequency.

1.0 Introduction

The automatic extraction of abstracts from a give source text document has been a neglected area of information science. Despite the increasingly availability of documents in electronic form and the availability of desktop publishing software, abstracts continue to be produced manually. It is therefore the purpose of this study to develop a system for abstract extraction from a give source document. An abstract is a brief overview or summary of the central subject matter of a given document. It is typically a very condensed summary of a study that highlights the major points and concisely describes the content and scope of the study. The challenges of manually reading and summarizing documents cannot be overemphasized. Most often documents 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: (i) Reading through a whole document and sorting out the essential points from it requires a lot of time and effort. (ii) A lot of man power is required to efficiently read and separate important extracts out from a document and can lead to high expenditure by the organization or body handing the processing of the documents. Automatic text summarization is the process of shortening a text document with software, in order to create a summary with the major points of the original document. Text summarization is also a technique where a computer automatically creates an abstract or summary of one or more texts. According to Babar [1] a summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using a summary is that it reduces the reading time Technologies that can make a coherent summary take into account variables such as length, writing style and syntax Abderrafih [2]. 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

abstract of the collection (whether documents, 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. [3]. An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. This paper presents the current technologies and techniques as well as prevailing challenges in automatic text summarization; consequently, we propose a model for improving text summarization by using a method that combines apriori algorithm for keyword frequency detection with clustering based approach for grouping similar sentences together.

2.0 Related Work

Text summarization has been an area of interest for many years. The need for an automatic text summarizer has increased much due to the abundance of electronic documents. Mani et al. [4] defines text summarization as the process of distilling the most important information from single or multiple documents to produce a condensed version for particular user(s) and task(s). Shen et al. [5] differentiates the two approaches to text summarization as abstraction based and extraction based. Abstraction based approach understands the overall meaning of the document and generate a new text whereas the extraction based approach simply selects a subset of existing sentences in the original text to form the summary. Liang et al, [6] developed a BE (basic element)-based multi-document summarizer with query interpretation. The idea is to assign scores to BEs according to some algorithms, assign scores to sentences based on the scores of the BEs contained in the sentences, and then apply standard filtering and redundancy removal techniques before generating summaries. The experimental results show that this approach was very effective. Khresmoi text summarizer developed at Dublin City University (DCU), Ireland for the Khresmoi project. The aim of the summarizer is to provide a summarized view of medical documents for use in the Khresmoi system interface. To achieve this, the summarizer selects the most meaningful/interesting segments in a text, for inclusion in the summary, by using features to describe segments and weight the importance of segments in documents Kelly et al. [7]. Baxendale [8] presented experimental data on how the leading sentences of a document are more important than the ones at the end in terms of its informative content or significance. Hence the position of a sentence in a document forms an important selection criterion. Luhn H. P. [9] presented the idea that frequently occuring terms signify the overall content of the document. S. Brin et al. [10] used the Pagerank based score to rank the sentences which gives more importance to sentences that refer to others as well as are referred by others.

3.0 Materials and Methods

This paper proposes a new approach to single document summarization. The summarization combines Apriori Algorithm for keywords frequency selection and clustering based approach for sentence grouping similar sentences together. This ensures good reporting and avoids redundancy in the extraction process. The algorithm finds the frequency of the most occurring sets of words in the sentences that make up the write-up of the document. The text in the document will be broken down into sentence units. Then, the sentences will be checked for the combination of the most frequently occurring group of words. The first *n* sentences with the most occurrences of the common word groups will be used as the sentences that will make up the abstract, where *n* is the number of sentences that will make up the abstract. 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:

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

The clustering algorithm is as shown below

Initialize $\mathbf{m}_i$, i = 1,…,k, for example, to k random $\mathbf{x}^t$
Repeat
For all $\mathbf{x}^t$ in X
$b_i^t \leftarrow 1$ if $\| \mathbf{x}^t - \mathbf{m}_i \| = \min_j \| \mathbf{x}^t - \mathbf{m}_j \|$
$b_i^t \leftarrow 0$ otherwise
For all $\mathbf{m}_i$, i = 1,…,k
$\mathbf{m}_i \leftarrow$ sum over t $(b_i^t \mathbf{x}^t)$ / sum over t (b_i^t)
Until $\mathbf{m}_i$ converge
The algorithm for the abstraction is as shown below;
1. Start
2. Select Document
3. Extract Sentences
4. Set Abstract length *L*
5. Set an arbitrary value s that will indicate maximum frequency size
6. Get frequency of word combinations in sentence
7. Get *s* number of sentences having the highest word combination frequencies
8. Join sentences to produce the abstract document
9. Display the abstract document
10. End

3.0 Proposed System

The proposed system is a single document summarization based on extractive techniques and will be implemented on text documents. The proposed system consists of five (5) phases which includes (i) preprocessing of input text, (ii) Document Analysis (iii) Filtering and Synthesis (iv) Abstract Generation (v) Abstract Document. The proposed work is a sentence extraction based single document summarization which creates a generic abstract of a given text document. This work uses a combination of Apriori algorithm and Clustering based methods to improve the quality of summary.

3.1 System Architecture

The system architecture is modeled in the diagram below.

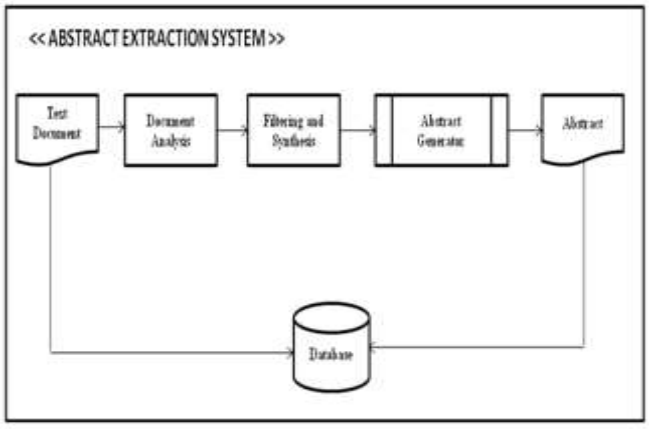

Figure 3.1 System Architecture

The system takes the text document as input. This input is always stored in the database. The document analysis phase carries out a preprocessing function i.e. the removal of stop-word and tokenization; stop-words are extremely common words (e.g. a, the, for). For this part, a stoplist which is a list of stop-words is used. Tokenization involves breaking the input document into sentences. The filtering and synthesis phase is where the summary text is produced from the summary representation. The abstract generator phase is where the representation of the document is turned into one of a summary of the document. The abstract is the generated summarized document which is the output of the system. This is also stored in the database.

3.2. System Implementation

The proposed system for Extracting Abstract from a Given Source Document was implemented using the following tools: VB.NET which was used to develop the user interface of the software to provide the interaction layer to enable the users of the system to interact with it. It will also be used to implement the processing logic of the model that will be used to extract information from the documents. MS-Access system was used to create the database for storing the information on the documents that are to be extracted for the abstract information as well as store the produced abstracts,
The modules for the proposed Document Abstract Extraction System are presented below.

Abstract Extraction Module: This module is used for the extraction of the document abstract.

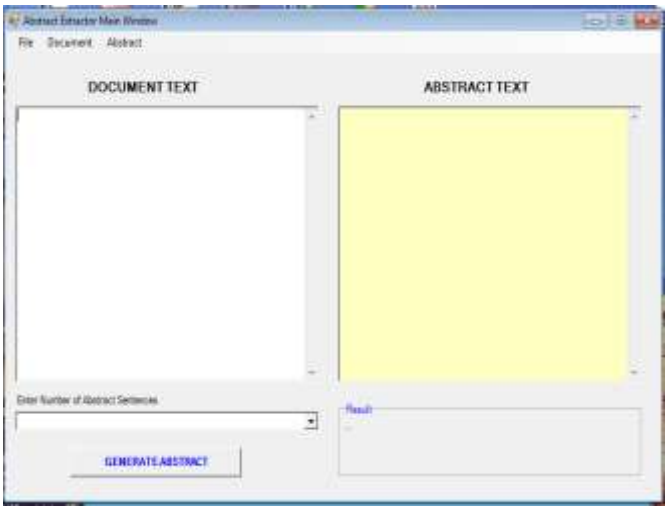

Figure 3.2 Text Input Interface

4.0. Result and Discussion

The tests carried out showed that the documents were able to be extracted successfully and also the number of sentences that the abstract was to have could also be controlled by the researcher thus fulfilling and main objectives of this computer science research. A sample output is as shown below;

Figure 4.1 Abstract Output Interface

The performance of the system is measured with the number of input words in the source document, number of words in the output summary file and its reduced words. Evaluation showed that there is at least 57% decrease of the output summary from the original input text.

5.0 Conclusion and Future Works

The project was aimed at providing a document extraction software system for the summarization of contents in a text document. This was achieved through the use of apriori algorithm and cluster algorithm that was used to weigh the best combination of words and sentences that contains most of the vital concepts of the documents. The result of this application will be enhanced by using spectral clustering and may side by side feature extraction to provide high level clustering.

The future work is expected to continue in this direction;

i. The system can be implemented for multiple documents.
ii. The System may also consider using Genetic algorithm for faster algorithm implementation.

REFERENCES

1. Samrat Babar (2013) "Text Summarization: An Overview ". ReseachGate.[online]. Available from: https://www.researchgate.net/publication/257947528_Text_SummarizationAn_Overview (October 2013)

2. 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.

3. Camargo E. Jorge 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.

4. I. Mani and M.T. Maybury.Advances in Automatic Text Summarization.The MIT Press, 1999.

5. D. Shen, J. T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In IJCAI, pp. 2862-2867, 2007.

6. 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>

7. 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. Available From <http://www.khresmoi.eu/assets/Deliverables/WP4/KhresmoiD44.pdf> (28 November 2017)

8. Baxendale, P. (1958). Man-made index for technical litterature - an experiment. IBM J. Res. Dev., 2 (4), 354 -361.

9. Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2), 159–165.

10. Brin S. and Page L., The anatomy of a large-scale hypertextual web search engine in WWW. Elsevier Science Publishers B. V. Amsterdam, The Netherlands, 1998.