New Graph-Based Text Summarization Method

Conference Paper · August 2015

DOI: 10.1109/PACRIM.2015.7334869

New Graph-Based Text Summarization Method

Saif alZahir and Qandeel Fatima
Image Processing and Graphics Lab,, CS Department
UNBC, PG, British Columbia, V2N 4Z9, Canada
e-mail: zahirs@unbc.ca and Fatima@unbc.ca

Martin Cenek
Computer Science and Engineering Department
UAA, Anchorage, Alaska, USA
mail: mcenek@uaa.alaska.edu

Abstract — **The exponential growth of text data on the World Wide Web as well as on databases off line created a critical need for efficient text summarizers that significantly reduce its size while maintaining its integrity. In this paper, we present a new multigraph-based text summarizer method. This method is unique in that it produces a multi-edge-irregular-graph that represents words occurrence in the sentences of the target text. This graph is then converted into a symmetric matrix from which we can produce the ranking of sentences and hence obtain the summarized text using a threshold. To test our method performance, we compared our results with those from the most popular publicly available text summarization software using a corpus of 1000 samples from 6 different applications: health, literature, politics, religion, science and sports. The simulation results show that the proposed method produced better or comparable summaries in all cases. The proposed method is fast and can be implement for real time summarization.**

Keywords— Text Summarization, extraction, abstraction, software testing, frequency of occurance, sentence ranking, aelevance.

I. INTRODUCTION

During the last five decades, researchers were encouraged to introduce a generic mechanism or algorithm that can reduce a huge amount of text data into meaningful shorter format. Such task is essential as nobody wants to spend most of his/her time reading books or lengthy report(s) if there is a precise the summaries of such documents are available. On the other hand, it is very difficult to manually summarize lengthy documents of text. In this period, several techniques were proposed. Even after more than half a century, text summarization is still a huge challenge for researchers and professionals.

In the business world, companies produce massive number of large reports, which are generated on annually, monthly, weekly, or even on a daily basis. Company owners and employees want these reports effectively summarized for quick reference. There are situations when we are surrounded by summarized information and we often take it for granted. For example, we cannot think of a newspaper without headlines or books and movies without reviews and trailers. Similarly, articles, such as this one, without abstracts and summarized results would be unacceptable.

Automatic text summarization [1] is defined as a summary that is generated by a machine to draw the most significant information in a shorter form and without any human assistance. A good Summary should keep the principal semantic content and help the user to quickly understand the large volumes of information. In 2002, Radev [8] formally defined a summary as: "A text that is produced from one or more texts, which conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". In simple words this definition can be divided into following points: (i) summaries can be produced from single or multiple documents; (ii) summaries should preserve the important part of the original text; (iii) reduces the original text by at least 50%.

Text Summarization is divided into two broad categories: (i) Abstraction and (ii) Extraction [3],[4]. Abstractive summaries are generated by interpreting the main concepts of a document and then stating those contents in a clear natural language [3],[4]. It aims to produce the main concept of the document or important material in a new way. Abstraction techniques are basically a substitute to the original document rather than the part of it. It uses linguistic methods to examine and interpret text. Abstractive summaries are very difficult to produce and did not arrive yet at a mature stage. Today's systems or computing devices cannot produce semantic representation, inference and natural language to such a level that is equivalent to humans. A person may generate two different summaries of the same article at different times. This behavior cannot be implemented in computer systems as of yet. Extractive summaries are generated via removing redundant sentences and/or words from the original text. Extracted text consists of important sentences or paragraphs from the original document [3]. These sentences are then concatenated into a shorter text to produce a meaningful and coherent summary. Extractive summaries are usually based on some statistical analysis of word/phrase frequency of occurrence, location words, etc., [3]. Extraction consists of two sub-methods – Fusion and Compression. As the name indicates, Fusion [1] is the coherent combination of extracted parts of the original text and compression [1] is to remove the unimportant sections of the text. It includes elimination of the redundant parts from the summary. Traditionally, extractive text summarization is divided into two steps [4]: (i) Pre-processing step and (ii) Processing step. Pre-processing step is cleaning the text from common words [3] like articles such asa, an, the, and prepositions such as on, of, over, at, etc. In this step sentences can defined by identifying the sentence boundaries. In some case one may need to apply stemming i.e., obtain the stem or radix of each word.

Summary produced by multiple documents is known as multi-document text summarization. Single document summarization includes one document as input to the summarizer. For monolingual summaries, the input and the

978-1-4673-7788-1/15/$31.00 ©2015IEEE

output summary is in the same language but for multilingual summaries, input or original text is in different language as compared to the target language of final output summary.

II. RELATED WORK

Computer based Text Summarization Techniques are scarce and can be divided into six categories. The first category is graph-based approach [4] [8] [9] [3] [10] [11] [1]. Graph based methods are used for both single document and multi-document summarization. In this category, each sentence is treated as a node. Two nodes or specifically two sentences are connected with an edge, if the two sentences have some similarity. Calculation of this similarity depends on many features. Like two nodes can be connected, if the sentences have the same words. Cosine similarity between sentences is a widely used measure. There are other methods that calculate the similarity between the sentences. In the graph-based approach, one may encounter sub-graphs, which may or may not be connected to the other sub-graphs. Sub-graphs show the number of topics covered in the document(s). From this, we can identify the most important sentences of the document(s). Nodes in the graph represent the sentences of the text. The nodes or sentences which have more edges connected to other nodes are the important ones. Graph based approach depends on the sentence centrality and centroid [4]. These are other measures that are used to calculate similarity between the nodes. The other approaches to find similarity measures between the nodes are, discounting, cumulative sum method [8], and Position Weight [9].

The second category is the machine-learning approach [3] [1] [7]. This approach was introduced in 1990s. The main methods, which use this approach, are naive-bayes methods, rich features and decision trees, hidden markov models, log linear models and neural networks. Researchers used naive-bayes methods [1] to calculate the naive-Bayes classifier. This classifier categories each sentence as worthy of extraction or not.

The third category is clustering based approach [11] [7]. This approach is suitable for both single document and multi-document summarization. As a first step, it creates clusters of documents, which are to be summarized and then find the relationships that exist among them. Similar documents and passages are clustered together so that the related information remains in the clusters. Then each cluster is indexed, which depends on the theme of the cluster. After clustering, sentences are ranked within each cluster then their saliency scores are calculated. In the last step high score sentences from each cluster are extracted to form the summary.

The fourth category is Lexical Chaining Approach [12] [13]. Lexical chains are basically defined as semantically related words spread over the entire document. It is a chain of words in which a word gets inclusion in a chain if it is cohesively and coherently related to the other words already present in the chain. Each chain of words represents the semantically related cluster of words. Words from the document are grouped together into meaningful clusters to identify various themes within a document. Then these clusters are arranged systematically to form a binary tree structure. Lexical chains start building up when the first word of the document starts. Then it checks for the second word that whether it is semantically related to the first one or not. If the words are related then the second word is also added in the first chain else second chain starts and so on.

The fifth category is Frequent Term Approach [14]. This method checks for the terms, which are frequent and semantically similar. There are some methods like term frequency – inverse document frequency, tf-idf, for the calculation of frequent terms in the document. For the calculation of semantic similarity, it checks the length of the path linking the terms, position of the terms, measures the difference of information content and the similarity between the terms. After this, summarizer filters the sentences having most frequent, semantically related terms and extracts them for summary.

The sixth category is Information Retrieval Approach [11] [15]. This approach is an enhancement of the two graphical methods LexRank (threshold) and LexRank (continuous), proposed in 2012. In this method the main feature is logical closeness i.e., how two sentences are logically related to each other rather than just the topical closeness. In addition, sentences must be coherent in sense. Finally more related sentences are picked up in a chain to produce the logical summary. This technique is very similar to graph based approach and lexical chaining. In fact it's a hybrid of two categories.

III. SUMMARIZERS PERFORMANCE

In this research, we have evaluated all publicly available summarizers on a 1000 text samples corpus. The corpus of passages of text is from different applications: Health, Literature, Politics, Religion, Science and Sports. We have eliminated several summerizers as they did not meet the minimum requirements of producing cohernt summaries. The remaining online summarizers we used are as follows:

1. SMRRY.
2. Tool4noobs.
3. FreeSummarizer.
4. AutoSummarizer.
5. SplitBrain.
6. Text Compactor.
7. Shvoong.
8. HelpfulPapers.
9. Article Summarizer Online.
10. MS Word Summarization.

Two of the summarizers, HelpfulPapers [20] and FreeSummarizer [21] do not have any online information or contacts. As for FreeSummarizer, we found that it is more suitable for news and long text. SplitBrain [17] and Shvoong [19] didn't provide any description of their product or any related information. For the remaining seven, we found information either on the website or by contacting through email. Tool4noobs [16] generates a summary for the given text by ranking each sentence considering the relevance. Text Compactor [18] is mainly designed for busy Students, Teachers or Professionals. SMMRY's [22] task is to provide an efficient manner of understanding the text, which is done primarily by reducing the text to only the most important sentences. AutoSummarizer [23] used several algorithms to produce better summaries. According to the developer, it uses K-means for clustering the sentences and a Bayer's naïve bayes classifier to calculate the probabilities between words and sentences. Naïve bayes method is one of the machine learning approach. In this method naive-Bayes classifiers are calculated to categories each sentence as worthy of extraction or not. AutoSummarizer used hybrid approach for the algorithms. It combines machine learning approach [3] [1] [7] and clustering based approach [11] [7] to produce meaningful summaries. Article Summarizer [24] underlines the main ideas, provide a brief overview, reflect the writing style and rephrase original text. Article Summarizer mainly deals with long passages and articles. It is helpful to figuring out what the main message of a text is, as it can help break it down. Finally, MS Word Summarizer [26] is very popular as compared to other summarizers. It has an AutoSummarize tool in Microsoft Office Word 2007. AutoSummarize identifies the key points in a document. It works best on well-structured documents, such as reports, articles, and scientific papers. AutoSummarize determines the key points by analyzing the document and assigning a score to each sentence. Sentences that contain frequently used words in the document are given a higher score. You can choose a percentage of the highest-scoring sentences to display in the summary. You can select whether to highlight key points in a document, insert an executive summary or abstract at the top of a document, create a new document and put the summary there, or hide everything except the summary.

Evaluation of the summaries generated by a summarizer is the most important step. Generally summaries are evaluated via two measure: (i) Intrinsic; and (ii) Extrinsic Measures [3]. Intrinsic measure is used to calculate the quality of a summary by humans or in other words, intrinsic measure depends on the human evaluation. Extrinsic measure is used to determine the quality of a summary by following a task based performance measure. It is a predefined procedure or standard which is used to measure the quality of summaries. In this research we use intrinsic and extrinsic measures to show our method performance.

IV. The Proposed Method

The proposed method is multi-graph based. The number of edges in the graph between two sentences (i.e., two nodes) is equal to the number of same words in both sentences. According to our assumption, a word may occur in a sentence more than once as in the example of Figure 1, such occurrence will be added in the symmetric matrix as shown in Tables 1 and 2. The total number of edges is stored in a symmetric matrix that represents the text being summarized. Then, we sum the values of rows (or columns – symmetric matrix) of the matrix to generate what we call a sum vector, which is then used for ranking the sentences as shown in table 1. This approach is used to replace other graph-based methods measures: term frequency, tf, and inverse document frequency, idf, which are used for more than half a century by researchers until this date. Our new approach can be summarized as follows:

1. Generate the symmetric matrix with the exact number of edges between the nodes (i.e., the sentences)
2. Algebraically sum the rows of the matrix to produce rand vector
3. Sort the ranking vector to get the sentence rank
4. Apply the cut-off mechanism using the required threshold to produce the summary.

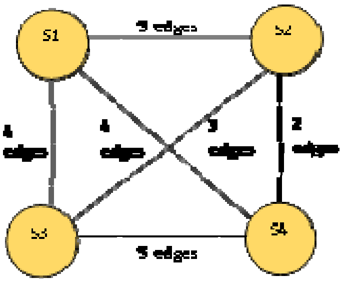

Fig. 1: Frequency of word occurrence analysis example

Table 1: Matrix generated by our method

	S1	S2	S3	S4		Sum	Sentence Rank
S1	0	5	4	4		13	1
S2	5	0	3	2		10	4
S3	4	3	0	5		12	2
S4	4	2	5	0		11	3

Table 2: Number of edges for our example

word in Sentence 1	**Match(s) in Sentence 2**
S1W1	
S1W2	S2W2
S1W3	
S1W4	
S1W5	
S1W6	
S1W7	S2W6
S1W8	
S1W9	S2W1
S1W10	
S1W11	
S1W12	S2W11
S1W13	
S1W14	
S1W5	
S1W15	
S1W16	
S1W17	
S1W12	S2W11
S1W18	
Total edges	**5**

Table 1 shows comparisons between the sentences and themselves as 0. For example, the comparison S1-S1, S2-S2, S3-S3, and S4-S4 are 0 in the matrix. Actually, the comparison(s) between S1 to S1 is 100 percent but we are assuming it as 0 in the matrix as it has no impact on the summarization process.

Our method is different from other method in that it does not use the cosine equation to find the similarity between the sentences. For the calculation of cosine equation, researchers have been using tf-idf but we are not calculating these factors for each word within a sentence. The other difference is that we are not identifying any relation within a sentence because within a sentence we don't have to find edges. Those edges may contribute towards redundant information in the summary. We are not tracking each word for the calculation of matrix. We are focusing on the significant words only.

The proposed method is efficient, simple and fast. Almost all methods have pre-processing schemes. For, our algorithm we tried to test its performance with and without preprocessing. We found that its performance has increased significantly when the preprocessing scheme was employed. Pre-processing reduces the size of the matrix by a considerable amount, which increases the performance and accuracy of the algorithm. Pre-processing mainly includes removal of articles, prepositions and meaningless words (like a sentences starting with a bracket or any special character). Advanced stage of pre-processing includes tagging (nouns, pronouns, adjectives etc.), semantics, stemming were not used in this research. Such advanced preprocessing will be considered for further research in the future.

V. Results And Analysis

To test the performance of the proposed method and compare it with the results of those publically available summarizers, we have selected a set of more than 1,000 text passages. The samples in the set are divided into three main data sets. In the first data set, we have 24 documents of 10 pages of text. The second set contains 6 documents each of which has 1 page of text. The third set contains 30 documents of one passage of text each.

We have evaluated the the results of the data sets of the sample and report a small segment of the results in Tables 3 and 4. The reason for that is that including the summaries of a 10 page text sample is too long to include in the paper. Tables 3, and 4 show the results of two samples of passages from the third data set.

The selected sample on which we observed the outputs from all these online summarizers is one of the passages from the corpus that we collected. Sample: "Globalization is commonly discussed in terms of how it enforces language and culture monopoly in transnational social structures and practices. It is also linked to deterritorialization of contacts among individuals, groups and institutions. With the exhaustion of the nation-state, the core-periphery metaphor is used to designate new distributions of power. This view obscures, however, the complex, polyphonic and heterogeneous nature of many peripheral contexts. It is argued that the regional element needs revisiting its capacities to serve as an interface between the global and the local (often national). The paper construes regionalism as a valid dimension of language studies in a foreign language macro culture of the Central and Eastern European countries. Some discussion follows the ongoing marketization of universities and technologization of language and translation teaching for the pressing needs of global and local markets. A counter-balanced engagement is proposed. Alongside some flashes of the region's academic cooperation in the past, an argument is made for the development of Critical Discourse Studies, with a checklist of topics being suggested and profiled on social and linguistic issues sensitive for this region." [28]

In order to meet the page requirement, we have selected the first 5 summarizers as well as the proposed method to evaluate the passage using Rouge. Rouge is an automatic summary evaluation metric, which is used to find quality of a generated summary by comparing it with gold standard summaries (generated by experts or human judges). We are using Rouge 2.0 for the evaluation of summaries as shown in Table 3 and 4. ROUGE 2.0 is a Java Package for the evaluation of summarization tasks building on the Perl Implementation of ROUGE with some updated and improved measures [29]. The output file from Rouge generates three variables: Avg_Recall, Avg_Precision and AvgF_score. These three variables are calculated by using the intersection of gold standard summaries with the summaries generated by the software. If A represents the gold standard summaries and B represents the generated summaries by the software, then we can say that:

$$\text{Precision} = (A \cap B)/A \quad (1)$$
$$\text{Recall} = (A \cap B)/A \quad (2)$$
$$F = (2PR)/(P+R) \quad (3)$$

Analyzing the outputs from Rouge, the Avg_Recall is higher when we get longer generated summaries and the Avg_Precision usually gets smaller value in this case. Ave._Precision is defined as the actual precision of the generated summaries with respect to the gold standard summaries. When gold standard summaries are smaller in length, then the generated summaries which are not very long generates better Avg_Precision and Avg F_Score.

As for our results, Tables 3 and 4 show that our method is efficient and produced excellent results as compared to other summarizers. Our scores in Table 3 exceeds all other summarizers results for the Ave_Precision and Ave._F_Score and comparable to others for the Ave._Call. In Table 4, our results outperformed all other methods in all categories.

As for the first and second data sets, the output summaries generated by all summarizers were not good. Most of the summarizers failed to generate summaries or they produced meaningless text. For example, software no. 6, text compactor couldn't summarize long text documents and displayed the error as "source text too long". Software no. 1 (SMMRY), 9 (Article Summarizer Online) and 10 (MS word Summarizer) also showed same kind of errors during the simulation. These summarizers are unable to summarize large documents. The other software, which are able to generate summaries for long text documents, they have generated very inefficient summaries. On the other hand, our results were easy to understand and reduced the text size significantly. Those summaries are too long in length to include in this paper.

Table 3. Performance Comparison: Sample 1

	Summarizer	Avg_ Recall	Avg_ Precision	Avg F_Score
1.	SMMRY	0.57975	0.06604	0.11857
2.	Tool4noobs	0.56052	0.06490	0.11634
3.	freeSummarizer	0.43278	0.07246	0.12414
4.	autosummarizer	0.61439	0.07143	0.12798
5.	Split brain	0.61056	0.07353	0.13125
6.	**Our Method**	**0.50179**	**0.08871**	**0.15077**

Table 4. Performance Comparison: Sample 2

	Summarizer	Avg_ Recall	Avg_ Precision	Avg F_Score
1.	SMMRY	0.61073	0.05063	0.09351
2.	Tool4noobs	0.51503	0.06182	0.11039
3.	freeSummarizer	0.60535	0.06154	0.11172
4.	autosummarizer	0.59535	0.04968	0.09171
5.	Split brain	0.54470	0.06154	0.11058
6.	**Our Method**	**0.65184**	**0.07544**	**0.13523**

In addition to the analysis above, we observed the following:

- Software no. 1 ignores brackets and their content. It also removes the words like: “nevertheless”.
- Software no. 8 ignores the text before 1st question mark or it ignores the first question.
- Software no. 9 considers each question as one sentence and ignores word like: “And so forth”, (if they appear at start of the sentence)
- Most used summarizers cannot recognize a separate sentence if there is no space in between the sentences. They consider 2 or 3 sentences as one sentence.
- Many summarizers couldn't recognize questions or question marks within the passage. Due to this, generated summaries are very large in length and do not have a good impact on the readers.
- Some summarizers include a part of the sentence because of semicolons. This is good for few examples but for most of the cases, this feature produced meaningless summaries.
- Some summarizers are unable to deal with the closing brackets or quotes. In other words they are unable to deal with punctuation properly.
- If a full stop comes in within a sentence like, Dr. Saif alZahir, many summarizers consider this sentence as two sentences.

VI. CONCLUSION

In this paper, we have introduced a new graph-based text summarizer that can efficiently reduce the size of a text while maintaining the integrity. The proposed method is simple to implement and does not rely on the calculations of the cosine similarity between sentences to rank the them in the summary. Such a process is a complex and not exact due to semantic assumptions. When comparing our method with the output results of the other real time summarizers on 1,000 text samples, we found that our results are better or comparable to those online summarizers. It is expected that our results will improve if an efficient preprocessing routine is used prior to the implementation of the proposed method. Such practice is common in Text summarization. Finally, the proposed method can be used for real time applications.

REFERENCES

[1] Dipanjan Das and Andre F.T. Martins, “A Survey on Automatic Text Summarization”, 21st November 2007.

[2] Karel Jezek and Josef Steinberger, “Automatic Text Summarization (The state of the art 2007 and new challenges)”, 2008.

[3] Vishal Gupta and Gurpreet Singh Lehal, “A Survey of Text Summarization Extractive Techniques”, Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.

[4] Gunes Erkan and Dragomir R. Radev, “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization”, Journal of Artificial Intelligence Research 22, pp. 457-479, 2004.

[5] H. P. Luhn, “The Automatic Creation of Literature Abstracts”, IBM Journal, April 1958.

[6] H. P. Edmundson, “New Methods in Automatic Extracting”, Journal of the Association for Computing Machinery, Vol. 16, No. 2, April 1969.

[7] Elena Lloret, “Text Summarization: An Overview”, TEXT-MESS (TIN 2006-15265-C06-01), 2006.

[8] Shanmugasundaram Hariharan and Rengaramanujam Srinivasan, “Studies on Graph based Approaches for Single and Multi Document Summarizations”, International Journal of Computer Theory and Engineering, Vol. 1, No. 5, Dec, 2009.

[9] Shanmugasundaram Hariharan, Thirunavukarasu Ramkumar and Rengaramanujam Srinivasan, “Enhanced Graph Based Approach for Multi-Document Summarization”, the International Arab Journal of Information Technology, Vol. 10, No. 4, July 2013.

[10] Xiaojun Wan, “An Exploration of Document Impact on Graph-Based Multi-Document Summarization”, Association for Computational Linguistics, pp. 755-762, October 2008.

[11] Md. Majharul Haque, Suraiya Pervin and Zerina Begum, “Literature Review of Automatic Multiple Documents Text Summarization”, International Journal of Innovative and Applied Studies, Vol. 3, No. 1, pp. 121-129, May 2013.

[12] Claudia Sofia Oliveira Santos, “ALEXIA – Acquisition of Lexical Chains for Text Summarization”, February 2006.

[13] Maheedhar Kolla, “Automatic Text Summarization Using Lexical Chains: Algorithms and Experiments”, 2004.

[14] Naresh Kumar Nagwani and Shrish Verma, “A frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm”, International Journal of Computer Applications, Vol. 17, No. 2, March 2011.

[15] Pritam Singh Negi, M. M. S. Rauthan and H. S. Dhami, “Text Summarization for Information Retrieval using Pattern Recognition Techniques”, International Journal of Computer Applications, Vol. 21, No. 10, May 2011.

[16] http://www.tools4noobs.com/summarize/

[17] http://www.splitbrain.org/services/ots

[18] http://textcompactor.com/

[19] http://www.shvoong.com/summarizer/

[20] http://helpfulpapers.com/summarizer-tool/

[21] http://freesummarizer.com/

[22] http://smmry.com/

[23] http://autosummarizer.com/index.php

[24] http://www.summarizing.biz/best-summarizing-strategies/article-summarizer-online/

[25] http://www.summarizer.info/

[26] http://office.microsoft.com/en-ca/word-help/automatically-summarize-a-document-HA010255206.aspx

[27] https://essential-mining.com/summarizer/online/?ui.lang=en

[28] http://www.lu.lv/fileadmin/user_upload/lu_portal/apgads/PDF/BJELLC-II-iekslapas.pdf

[29] http://kavita-ganesan.com/content/rouge-2.0