International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Extractive Text Summarization

Namita Mittal[A], Basant Agarwal[A*], Himanshu Mantri[A], Rahul Kumar Goyal[A] and Manoj Kumar Jain[A]

[A]Department of Computer Engineering, Malaviya National Institute of Technology, Jaipur India

Accepted 01 April 2014, Available online 15 April 2014, **Vol.4, No.2 (April 2014)**

Abstract

Text summarization helps in reducing the size of a text while preserving its information content. In this paper, a text summarization approach is proposed based on removal of redundant sentences. Initially, each sentence from original text (input) is scored based on how much redundant the sentence is and at what extent that sentence is able to cover other sentences by itself. This approach is best effective on the documents which are highly redundant and contain repetitive opinions about a topic. The summarization takes places in two stages wherein the input of a stage is the output of previous stage and after each stage the output summary is less redundant than the previous one.

Keywords: *Text summarization, redundancy removal, Extractive summary*.

1. Introduction

The main goal of text summarization is to convert the original input document in a shorter form such that there is no loss of important information and all the information contained in the original text is preserved. Also, the shorter version or the summary should satisfy the needs of the user (Gunes et al 2004).

Today on Internet, much more information is available on a particular topic that a user is searching for, it may be related to news information or biological information etc. (Chin-Yew et al. 2004). Since it is not possible to search out and read everything available information about a topic, some methods are required to condense the larger pool of information.

Sometimes the language of retrieved information about a topic may be harder to understand. For example, students who are learning French as a language course and French may not be their native language. In such situations, tools to simplify the language may be needed and would be highly useful. The summarization tool which would address both these problems would be able to cover larger number of population to be aware of a greater amount of information (Devasenal et al. 2012).

Text summarization is a crucially important issue in today’s world due to vastly available information. In today’s fast-growing information world, due to immense availability of information, text summarization has become important tool for interpreting text. It is very difficult for normal human beings to manually summarize large pool of documents. Huge amount of information is retrieved from the Internet than required. Hence a dual problem arises. First, the relevant documents have to be retrieved out of the large pool of documents and second, the retrieved documents are to be absorbed to grasp large amount of information (Pal et al. 2002).

In this paper, a graph based method is proposed that eliminates the redundant sentences to get the summary of input document.

This paper is organized as follows. Section 2 describes the related work. Proposed approach is presented in section 3. Experiments and results are discussed in section 4. Finally, section 5 stats the conclusion.

2. Related Work

Text summarization has attracted the researchers due to its wide range of applications. Ganesan et al. (2010) present a novel graph-based summarization framework (Opinosis) that generates concise abstractive summaries of highly redundant opinions. One another graph mining approach for text classification is defined in the approach given by the authors Arey et al. (2005). A Hough transform based technique for word and line segmentation from digitized images and business card reader system and license plate recognition system is presented in (saha et al. 2010).

3. Proposed Approach

The proposed approach is more efficient on documents which are highly redundant. For example, if summary of large number of user reviews about a product or service is required, proposed method may work efficiently. The approach is to extract sentences which are redundant enough to cover up few other sentences in them and scoring higher to make a place in final summary. For example sentence as shown in Figure 1, My phone calls drop frequently with the iPhone great device, but the call drops too frequently.

*Corresponding author: **Basant Agarwal**

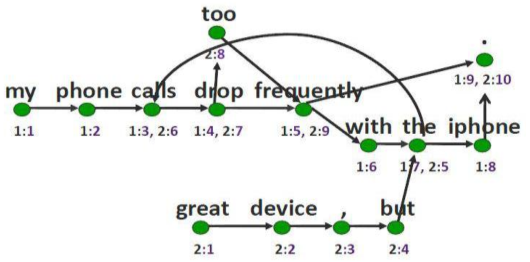

Figure 1 Example Sentence

1. Each unique word corresponds to a node
2. Since we only have one node per unique word unit, each node keeps track of all sentences that it is a part of using a sentence identifier (SID) along with its position of occurrence in that sentence (PID).
3. Each node will thus carry a Positional Reference Information (PRI) which is a list of {SID:PID} pairs representing the node’s membership in a sentence.

The proposed approach is divided in three stages as shown in Figure 2.

3.1 Pre-processing

3.1.1 Remove stop-words: Since stop-words e.g. ‘is’, ‘are’ etc. do not contribute much in the meaning of a text, they are removed from the original text.

3.1.2 Stem original text: Stemming is finding the root word of any text. Stemming is applied to the original text which causes ‘started’ to ‘start’ and ‘explores’ to ‘explore’, thus eliminating redundancy or words in other forms.

3.1.3 Input text: The original text is input word-by-word and every word is inserted into its corresponding path.

3.1.4 Update SID-PID Table: SID is sentence identifier and PID is position identifier of a word. Each unique word is associated its positional information in the original text. Each word is connected with its SID: PID pair table.

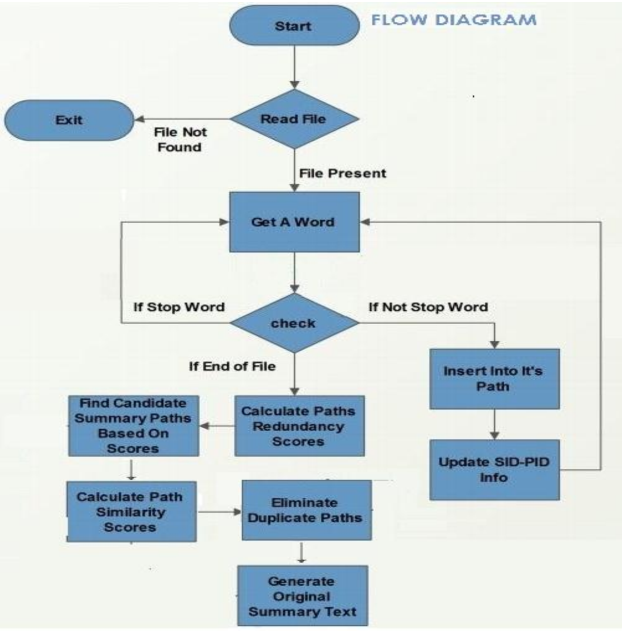
FLOW DIAGRAM
Start
Exit
File Not Found
Read File
File Present
Get A Word
If Stop Word
check
If Not Stop Word
If End of File
Insert Into It's Path
Find Candidate Summary Paths Based On Scores
Calculate Paths Redundancy Scores
Update SID-PID Info
Calculate Path Similarity Scores
Eliminate Duplicate Paths
Generate Original Summary Text

Figure 1 Overview of proposed method

3.2 Path Scoring

After all the input is fed in, the **Path Redundancy Scores** are calculated for all the valid sentences.

To calculate path redundancy score of a sentence, the SID-PID table of every possible pair of words is checked to see in how many sentences both the words are occurring simultaneously. This can be accomplished by matching their SIDs.

The next step is to find candidate summary paths on the basis of path redundancy scores.

3.3 Similar Sentences Removal

The output of stage-2 is fed as input to stage-3. To further remove the redundancy, similarity scores are calculated for all paths. We have used Jaccard similarity measure for the same purpose.

$$J = \frac{A \ \cap B}{A \ \cup B}$$

where A and B are two sentences, A∩B is the number of intersecting words and AUB is the number of words in the union of two sentences (Chin-Yew et al. 2004).

Once the similarity scores are calculated, the sentences which are similar almost similar to each other, only once of them is retained to be included in the final summary and rest are eliminated. Finally, the final summary is generated.

4. Experiments and results

We have used ROUGE-Recall Oriented Understudy for Gisting Evaluation toolkit to test the quality and completeness of the summaries produced by out summarizer. The summaries produced are brief, reasonably in-shape and they communicate important information of the original document.

Table 1 Results obtained for some of documents

Recall	Precision	F-Score
0.75	0.51064	0.6076
0.86486	1	0.92753
0.60938	0.68421	0.64463
0.60938	0.68421	0.64463
0.65686	0.44966	0.53386
0.97087	0.65789	0.78431
0.21719	0.99463	0.35653
0.76608	0.68947	0.72576
0.44444	0.51128	0.47552
0.69939	0.56716	0.62637
0.2988	0.99538	0.45963
0.76136	0.56303	0.64734
0.86207	0.67568	0.75758
0.68519	0.55639	0.61411
0.5679	0.49462	0.52873
0.76768	0.63866	0.69725
0.73404	0.6699	0.7005

Our test results reveal that more than 60% of the generated sentences match with the original input text. ROUGE determines the quality of a summary by comparing it with a standard summary of the same original text. The method counts the number of overlapping n-gram unit, word sequences and word pairs between both the summaries. F-score values are given for some of the documents in Table 1.

5. Conclusion

In this paper, we have described about general overview of automatic text summarization. Research in this area is continuously going on and day by day new developments are being made. The summaries produced by our summarizer are reasonably well-shaped and brief and readable. They convey the information content of the original text to a great extent. Almost 60% of the sentences extracted in the final summary match with the original sentences.

References

Gunes E., Radev DR. (2004). Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artif. Int. Res*., 22(1):457–479.

Devasenal CL, Hemalatha M. (2012), Automatic Text Categorization and Summarization using Rule Reduction, *In International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30, 31,*

Lal P., Ruger S. (2002). Extract-based summarization with simplification. *In Proceedings of the ACL 2002 Automatic Summarization / DUC 2002 Workshop, 2002.*

Chin-Yew L.. (2004). Rouge: a package for automatic evaluation of summaries. *In Proceedings of the ACL Workshop on Text Summarization Branches Out, Barcelona, Spain*, pp 74-81

Ganesan K., Zhai CX, Han J.. (2010). Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics* (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 340-348.

Aery M., Chakravarthy S. (2005), lnfoSift: Adapting Graph Mining Techniques for Text Classification, *American Association for Artificial Intelligence*.

Saha S., Basu S., Nasipuri M.,. Basu D., (2010). A Hough Transform based Technique for Text Segmentation, *Journal of Computing*, vol. 2, Issue 2, February 2010, pp. 134-141.