ijecs open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11134-11138

Graph Based Approach For Multi Document Summarization

Mr. Vijay Sonawane, Prof. Rakesh Salam
Information Technology Department
Technocrats Institute of Technology, Anand nagar, Bhopal, India
Sonawanevijay4@gmail.com
rakeshsalam@rediffmail.com

ABSTRACT- ***Summarization is the process of decreasing large source document to shorten version of summary which will be easy to read. Document summarization is an emerging technique which is used for understanding the main purpose of any kind of documents. Summarization can be either single or multi document summarization. If summary is to be generated for single document then it is called as single document summarization. If summary is to be created for multiple relevant documents then it is called as multi document summarization. An Graph based approach for Multi Document Summarization is a graph based multi document summarization technique in which, set of documents is preprocessed, undirected graph will be constructed to calculate similarity between sentences, the word class is attached to each sentence, sentences are ranked according to word class and similarity of sentences and top ranked sentences are included in the summary.***

Keywords - Single Document, Multi Document, Summarization and Sentence Ranker.

1. INTRODUCTION

A **s**ummary can be defined as a text that is generated from one or more texts, that include a major part of the information in the original text(s), and that is no longer than half of the original text(s) [6].Text summarization is the process of distilling the most important information from a source (or sources) to produce a shorter version for a particular user (or users) and task (or tasks) [10]. Roughly summarization is the process of decreasing a large volume of information to a summary or abstract preserving only the most essential items.

Due to the rapid growth of the Internet and the emergence of low-cost, large-capacity storage devices, we are now exposed to a lot of online information in daily life [1]. This situation makes it difficult for us to find and gather which exact information we need. Automatic text summarization is a key technology to solve this difficulty [2], with the properly summarized information, we can quickly and easily understand what the major points of the original document are and find how relevant the original document is to our own needs. We need to get right information without having gone through the source document [12]. Therefore we need a summary of document so that we can get the main purpose of the whole document.

2. CLASSIFICATION OF TEXT SUMMARIZATION TECHNIQUES

Text Summarization is condensing the source text into a shorter version preserving its information overall meaning and content [6]. The text summarization techniques can be classified by using the way by which the summarization method is going to be performed over the text data.

The summary may be either generic or query specific [15]. In a generic summary, the important sentences are selected from the document and the sentences so extracted are arranged in the appropriate order. In a query specific summary generation, the sentences are silent scored based on the query given by the user. The extracted highest silent scored sentences and presented to the user as a summary. Following are the two broad level method classifications of text summarization techniques.

An extractive summarization method [15] consists of select important sentences, paragraphs etc. from the original source document and concatenating them into a shorter form. The importance of sentences is decided based on statistical and linguistic features of document sentences. Extraction summarization techniques merely copy the information deemed most important by the system to the summary (for example, key clauses, sentences or paragraphs). An abstractive summarization method consists of understanding the original source text and re-telling it in fewer words [18]. It uses linguistic methods to interpret and examine the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document. Abstraction text more strongly than extraction, but the programs that can do this are harder to develop as they require the use of natural language processing technology.

If summarization is generated for a single text document then it is called as the single document text summarization [9]. Single document text summarization techniques have the potential to simplify information consumption on mobile phones by presenting only the most relevant information contained in the document. If the summary is to be performed for multiple text documents then it is called as the multi document text summarization technique [16]. Multi-document summarization creates information reports that are both comprehensive and concise.

3. EXISTING TECHNIQUES OF TEXT SUMMARIZATION

3.1 RANDOM Based

The RANDOM based technique [10] is the simplest techniques as compare to all the above mentioned techniques as it randomly selects sentence from the document, depending upon the compression percentage and put them in the summary. In this technique, a random value between 0 and 1 is assigned for each sentence of the document. It also provided threshold value for compute the length of the sentence. We will assign a score of 0 to all sentences that do not satisfy this length cut off [12]. Finally we choose required sentences according to assigned highest score for extractive summary.

3.2 LEAD Based

LEAD based technique [12] is a technique in which first or first and last line of the paragraph are selected based upon the compression rate (CR) and it is very good for news articles as they have the major point set in the first lines of the articles. So, it can be tolerable that n% sentences are selected from beginning of the text e.g. selecting the first sentence of each document, then select the second sentence of each, etc. until the desired summary is constructed. This method is called LEAD based method for document summarization. In LEAD [10] based technique we will assign a score of 1/n to each sentence, where n is the sentence number in the corresponding document file. These describe that the first sentence of each document will have the same scores; the second sentence of each document will have the same scores, etc. It also provides a threshold value for calculate sentence's length.

3.3 MEAD Based

MEAD [12] is a centroid-based extractive summarizer which scores sentences based on sentence-level and inter-sentence features which indicates the quality of the sentence as a summary sentence. It then selects the top-ranked sentences for producing the output summary. MEAD produce centroid for all of the sentences and then select those sentence which are centroid (vector) closed to the sentences. MEAD [12] based extractive summaries sentences are score according to certain sentence features - Centroid, Position, and Length. In this technique the score of a sentence is calculated using the following formula as follows [10].

$$\text{Score (Si)} = \begin{cases} \sum (Wc * Ci + Wp * Pi) & \text{If Length (Si) > Threshold} \\ 0 & \text{If Length (Si) < Threshold} \end{cases}$$

Here,

Wc = The weight for the Centroid feature.
Wp = The weight for the Position feature.
Ci = The calculated Centroid value for ith sentence.
Pi = The calculated Position value for ith sentence.
Si = The ith sentence of the document.
i = Sentence number within the cluster
n = Number of sentences in a single or multiple text documents.

The highest score value of sentence is taken in the extract file. Thus the MEAD based summary is created. The default weights for Centroid and Position are both 1. The default Length cutoff is 9.

Stemming

In information retrieval, stemming is the process for convert inflected (or sometimes derived) words to their root form, generally a written word form. In documents a word can be seen in different formats, such as present vs. past tense, plural vs. singular etc [23]. Most of the time these words have the same meaning and but treat them different is unnecessary. In order to use these words as the same concept, stemmers are used.

While performing further calculations the stemmer efficiency is important. Most of times stemmers can do over-stemming such that two words are given the same stem, while it should not be. For example, the words “experiment " and “experience “are two different words, which should not be stemmed into the same root. But stemmers can find out their root as "experi". Another problem of stemming is related to under-stemming such that two words should have been stemmed into the same word, but have not been. For example, “ran” and “run “can be found as two different stems, instead of one.

- *Stemming Examples*

A stemmer for English, for example, should identify the string suffix "applicant"(and possibly "applicator ") as based on the root "applica ", and "stemmer ", "stemming ", as based on "stem". A stemming algorithm reduces inflected the words "assists", "assisting", and "assisted" to the root word, "assist".

Stopword Filtering

Stop words are words which are remove out prior to, or after, processing of natural language data (text) [23]. It is controlled by user input and not automated. There is not standard list of stop words which all tools use, if even used. Input documents usually include words that do not add information but are necessary for syntactical formation, such as words like "this", "was", etc. Since these words are less useful and less informative, they introduce noise into the input matrix (document representation). In order to get these kinds of words, a stop word removal step is used.

Stop word removal is done using predefined, human-create list of words. The words in the list are not used while generating the input matrix. Since a predefined list is used, this approach is language dependent. Instead of using these kinds of lists, a frequency threshold can be used. If a word is seen more/less frequently than predefined threshold, that word can be considered as stop word. But the decision of threshold is another issue to be considered.

- *Stop Words Examples*

Following are the examples of stop words as follows - about, the, this, after, again, is, all, am, are, the, was, were, back, backed, can, do, does, done, down etc.

4. PROPOSED METHODOLOGY

Graph Based Multi Document Summarization Multi Document Summarization is graph based multi document summarization algorithm. The Algorithm consists of the steps mentioned in Fig.1.The input passed to the system is a set of text documents. Firstly, the input set of related documents is pre-processed. Classes are attached to each sentence of the document and sentence length is calculated. The undirected graph will be constructed for each text document with sentences as vertices and similarities as edges. Thereafter, the sentences are ranked according to their absolute class, summed class and salient scores. The select top-ranking sentences to form the summary for each document and semantic checking are also used to filter out redundant information. Next, the single summary of each document will be assembled into only one document. Finally, the above described process is applied to this combined document to form the desire extractive summary.

4.1 Preprocessing

Before attaching a class to a sentence, the input set of related documents will be required to preprocess. Initially, the input documents are parsed to select all sentences. Those sentences, which are too short or almost, contain no information [12], then they are eliminated. Here all stop words are removed from each document and words are converted to their respective root form. Stemming is applied to reducing inflected words to their root form.

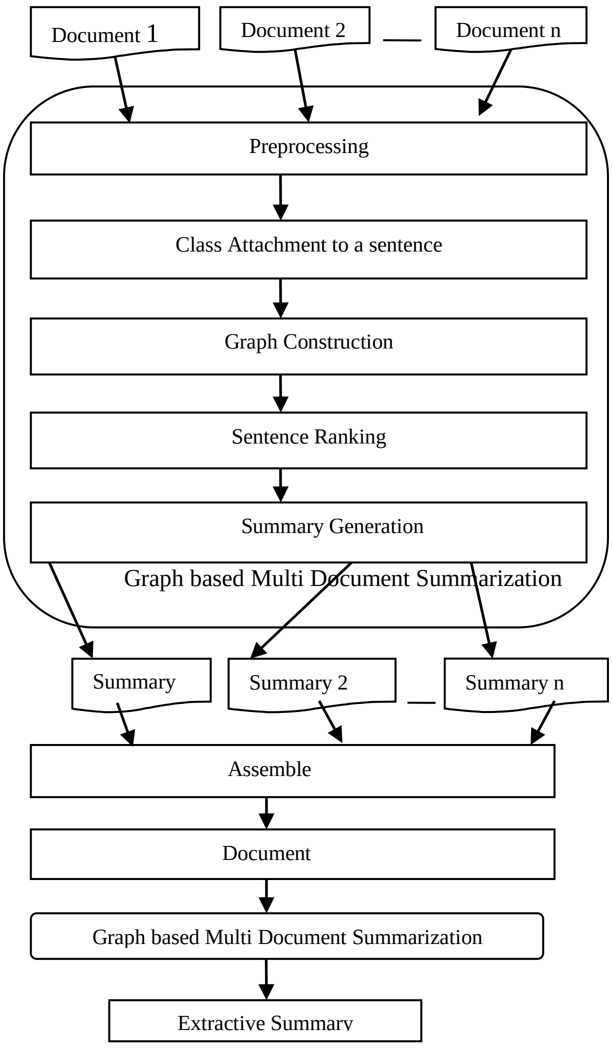

Fig.1: Main Process Graph Based Multi Document Summarization

For example, “finding” is converted to “find” [23]. In GBMDS, text file of stop words is maintained. If a sentence contains stop word present in a file then it is removed.

4.2 Class Attachment to the Sentence

Before constructing the graph, class is attached to each sentence of the documents. Here the database of word class is maintained. The sentences words attach to word class using predefined word class [23]. According to the database the absolute and summed class is attached to each sentence and calculated length of each sentence [7]. Length of each sentence is calculated as a number of characters present in a sentence. If sentence contains n characters then length of that sentence is n.

4.3 Graph Construction

The graph G = (V x E) which represents each sentence presenting in the document becomes a node and the edges of the graph represent similarity between the sentences.

$$\text{Similarity}(S_i, S_{i+1}) = \frac{\text{sum}(A_i - A_{i+1}, B_i - B_{i+1} \ldots Z_i - Z_{i+1})}{\text{Number of Characters}}$$

Where,
i = i^{th} sentence of the document.
A_i = Count indicating the number of times A has occurred in i^{th} sentence.

A_{i+1} = Count indicating the number of times A has occurred in i+1th sentence.
B_i = Count indicating the number of times B has occurred in ith sentence.
B_{i+1} = Count indicating the number of times B has occurred in i+1th sentence.
Up to
Z_i = Count indicating the number of times Z has occurred in ith sentence.
Z_{i+1} = Count indicating the number of times Z has occurred in i+1th sentence.

Using this formula to calculate the similarity between the sentences, this means calculate the graph value from each sentence of source document.

4.4 Sentence Ranking

Once the document graph is constructed, the sentences in a source document will be ranked based on the absolute class, similarity between sentences and length of sentence [13]. The sentence is given high rank if its absolute class is higher than the remaining sentences of absolute class. If an absolute class between two sentences are given same value then the sentence is ranked based on the length of sentences. i.e. The sentence which has highest length will be given to next higher rank or else on the basis of similarity between sentences [12].

4.5 Summary Generation

In this step, final summary is generated by using selecting top ranking of sentence. Here, top rank of each sentence is refined according to the summed class. Summed class is used for arrangement of summary in proper sequence [10]. Simply, high ranking scores with sentences may be selected as the final ones in the summary. The sentences score is calculated based on relevant value and in-formative value.

5. CONCLUSION

A summary can be defined as a text that is generated from one or more texts, that include an important part of the information in the original text(s), and that is no bigger than half of the original text(s). Graph based approach for multi document summarization technique. In this technique, sentences are preprocessed, class is attached to each sentence, sentence length is calculated, undirected graph will be constructed, and each sentence is given rank based on class and then top ranked sentences has selected in summary, therefore its more efficient than other technique.

6. REFERENCES

1. Giuseppe Di Fabrizio, Ahmet Aker, "STARLET: Multi-Document summarization of Pro Product and Service Reviews with balanced rating Distributions", 2011 1th IEEE International Conference on Data Mining Workshops, DOI 10.1109/ ICDMW.2011.158, 2011 IEEE.
2. Daan Van Bristsom, Antoon Bronselaer, Guy De Tr'e "Automatically Generating Multi Document Summarization", 2011 11th International Conference on Intelligent System Design and Application.
3. J. Feng, M. Johnston, and S. Bangalore, "Speech and multi-modal interaction in mobile search," Signal Processing Magazine, IEEE, vol. 28, no. 4, July 2011.
4. G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target Extraction through double propagation," Comput. Linguist, vol. 37, 2011.
5. N. Gupta, G. Di Fabbrizio, and P. Haffner, "Capturing the stars: predicting ratings for service and product reviews," in Proceedings of the NAACL HLT 2010 Workshop on Semantic Search, ser. SS '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 36–43.
6. A. Aker, T. Cohn, and R. Gaizuaskas, "Multi document summarization using a* search and discriminative training", in proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics, 2010, pp. 482-491.
7. Naomi Daniel, Dragomir Radev, Timothy Allison "Sub-event based multi-document summarization" Proceedings of the HLT-NAACL 03 on Text summarization workshop - Volume 5
8. Jayabharathy, Kanmani, Buvana "An Analytical Framework for Multi-Document Summarization" International Journal of Computer Science Issues (IJCSI);May2011, Vol. 8 Issue 3, p308
9. Antoon Bronselaer, Saskia Debergh, Dirk Van Hyfte, Guy De Tr'e, " Estimation of topic cardinality in document collections," in Proceeding of the 10th SIAM 2010 conference on data mining.
10. Antoon Bronselaer, Guy De Tr'e, "Aspects of object merging", in Proceeding of the NAFIPS Conference, Toronto ,Canada, 2010.
11. G. Carenini and L. Rizoli, "A multimedia interface for facilitating comparisons of opinions," in IUI '09: Proceedings of the 13th international Conference on Intelligent user inter-faces. ACM, 2009, pp. 325–334.
12. Mohsin Ali, Monotosh Kumar Ghosh, "Multi-document Text Summarization:
SimWithFirst Based Features and Sentence Co-selection Based Evaluation", 2009 International conference on Future Computer and Communication. Department of Computer Science and Engineering, Khulna University, Bangladesh.
13. W.Duan, B.Gu, A.B.Whinston," Do Online Reviews Matter?- An Empirical Investigation of Panel Data," Journal Decision Support System, vol.45,2008.
14. B.Pang, L.Lee , "Opinion mining and sentiment analysis," Foundation and Treands in Information Retrieval, vol.2 2008.
15. A.Esuli, "Automatic generation of lexical resources for opinion mining: models, algorithms and application," SIGIR forum vol.42, 2008.
16. D. Park, J. Lee, and I. Han, "The effect of on-line consumer reviews on Consumer purchasing intention:

The moderating role of involvement," Int. J. Electron.Commerce, vol. 11, pp. 125–148, July 2007.

17. G.Carenini, R.Ng, and A.Pauls, "Multi-document summarization of evaluative text," in 11th Meeting of the European Chapter of the Association for Computational Linguistics, 2006.

18. Mokoto Hirohata, Yousuke Shinnaka "Sentence Extraction Based Presentation Summarization Techniques and Evaluation Metrics" 2005 IEEE ICASSP.

19. Chin Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries".

20. Chin Yew Lin. "ROUGE Working Notes" 2004 IEEE.

21. Sparck Jones, K. Automatic summarizing: factors and directions. Advances in Automatic Text Summarization.MIT Press

22. P,Naveen Kumar, A.P.Shiva Kumar "Concept Frequency: A Feature set Based Text Compression Model" 2012 International Journal of Advanced Research Computer Science and Software Engineering.

23. Rafeeq Al-Hashemi, "Text Summarization Extraction System (TSES) Using Extracted Keywords " International Arab Journal of e-Technology, Vol. 1, No. 4, June 2010

24. E Balagurusamy, "Programming with a Java", Fourth Edition, Tata McGraw Hill Publication.

25. Herbert Schildt, "The Complete Reference Java", Seventh Edition, Tata McGraw Hill Publication.

26. G. Booch, James Rumbaugh, "Object Oriented Modeling and Design", Second Edition, Prentice Hall

27. R. Pressman, "Software Engineering: A practitioner's Approach", Seventh Edition, McGraw International Edition, 2010,