Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 46 (2015) 285 – 291
Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Vertex Cover Algorithm Based Multi-Document Summarization Using Information Content of Sentences

Ansamma John[a,*], M Wilscy[b]

[a]Department of Computer Science & Engineerring, T.K.M.College of Engineering, Kollam, 691005, India
[b]Department of Computer Science, University of Kerala, Kariavattom, Trivandrum, 695581, India

Abstract

In recent times, the requirement for the generation of multi-document summary has gained a lot of attention among researchers. Multi-document summarization systems focus on generating compressed form of the documents which maintains the pertinent features of the original documents. Mostly, text summarization techniques use the sentence extraction technique where the salient sentences in the multiple documents are selected and presented as a summary. In our proposed system, we have developed a sentence extraction based multi-document summarization system using the principle of vertex cover algorithm which automatically selects relevant sentences that cover the predominant concepts of the input documents. This frame work represents the documents as a weighted undirected graph with sentences as the vertices and the similarity between the sentences as the edge weight between the corresponding vertices in the graph. The experimental result on the DUC 2002 data sets demonstrates the effectiveness of the proposed method in document summarization

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Multi-document Summary; Vertex cover; Redundancy; DUC 2002 Dataset (Document Understanding Conferences)

1. Introduction

The amount of information in World Wide Web is getting enlarged day by day, resulting in information overload. In other words, to utilize the information effectively is a challenging practical task. An urgent need for text

* Corresponding author. Tel.: +91-9447453101.
E-mail address: a_john13@yahoo.co.in

1877-0509
Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.022

summarization has materialized due to information overload[1]. Text summarization relates to the process of obtaining a textual document, obtaining content from it, and providing the necessary content to the user in a shortened form and in a receptive way to the requirement of user or application[2]. The technology eases the inconvenience of information overload because only a concise review has to be considered instead of a complete textual document[3]. From the early stages of text summarization, its main purpose was to assist the user in finding the information by condensing the vital information from a fundamental source and providing its shortened form. In this regard, text summarization is regarded as a mediator between the user and information included in several documents [4].

Text summarization techniques are classified as abstractive summarization and extractive summarization. Abstraction can be described as the process of reading and understanding the text to recognize its content that is then compiled to a concise form[5]. Abstractive summarization is based on Natural Language Processing (NLP) techniques for parsing, deducing and generating summary. In contrast, extractive summarization can be described as the method of verbatim extraction of textual elements like paragraphs, sentences, words etc. from the source text. Extractive summarization is noticed to be flexible and consumes less time as compared to abstractive summarization[6].

Multi-document summary possesses some notable merits over a single-document summary. It offers a domain summary of a topic based on a document set representing identical information in several documents, distinct information in separate documents, and association between sections of information across various documents. It can enable the user to look for more information on certain facets of interest, and look into the distinctive single-document summaries[7]. Most of the techniques employed in single-document summarization are also employed in multi-document summarization. There exist some notable difficulties[8]. (1) The degree of redundancy of information available in a group of topically-related documents is significantly larger than the redundancy of information within a single document, since each document incorporates important concepts and also the required shared background. So, anti-redundancy methods play a vital role. (2) The compression ratio (that is the summary size with regard to the size of the document set) will considerably be lesser for a vast collection of topically related documents than for single document summaries. When compression demands get intensified, summarization becomes challenging. (3) The co-reference problem in summarization possesses bigger challenges for multi-document than for single-document summarization [9, 10].

In this paper, we present a graph based automatic multi-document summarization system using vertex cover algorithm[11]. Vertex cover of a graph represents subset of its vertices which can cover all the edges of the graph. Extractive multi-document summarization problem finds the subset of sentences from the input document which are relevant and cover major concepts. So multi-document summarization problem can be transformed to a vertex cover problem. A weighted undirected graph is constructed for the entire document set where vertices of the graph represent sentences of the document. Edges are maintained between every pair of vertices if the sentences corresponding to the vertices have some similarity. Weight is assigned to each of the edges in the graph using the combined cosine similarity and Normalized Google distance measures[12]. Vertex cover algorithm is applied on this graph with some constraints to select the most appropriate and required numbers of vertices on which all edges of the graph are incident. Sentences corresponding to the selected vertices give the summary with maximum relevance, minimum redundancy and limited length. We have used DUC 2002 dataset[23] to evaluate the system and results are analyzed using ROUGE Tool kit[24].

The rest of the paper is organized as follows: The review of related researches is given in Section 2. The proposed automatic summarization system is presented in Section 3. The experimental results and analysis are given in Section 4. Finally, the conclusions are summarized in Section 5.

2. Review of related researches

Numerous extractive and abstractive summarization methods are proposed in the literature to summarize the multiple documents. Extractive summarization based on topic identification and sentence ranking is more popular than abstractive summarization. Some of the works presented in the extractive multi-document summarization are given as follows:

Dragomir R. Radev *et al.*[13] have presented a multi-document summarizer, MEAD, which created summaries by employing cluster centroids, a vector of statically important terms in the clusters, generated by topic detection and tracking system. It discussed two techniques, a centroid-based summarizer which ranks and extracts sentences based on the features such as centroid value, position information and first-sentence overlap value, and an evaluation scheme on the grounds of sentence utility and subsumption.

Goldstein *el al*[14] proposed a frame work using Maximal Marginal Relevance (MMR).This a query based summarization method which reduces redundancy in the summary while maintaining the relevance of the query during ranking process. Summarization by MMR, computes the cosine similarity between the given query and the input document and the cosine similarity between the currently selected sentence and partially constructed summary up to the current point. MMR computes relevance and novelty independently and their linear combination (the marginal relevance) is provided as a metric and maximizes this value to generate good summary.

Zhanying HE *et al*[15] have presented the text summarization problem as a data reconstruction problem and proposed the method Document Summarization based on Data Reconstruction (DSDR) which finds the summary sentences by minimizing the reconstruction error. DSDR first learns a reconstruction function for each sentence in the input documents and identifies an error formula for each sentence using that function. Optimal summary is generated by decreasing the reconstruction error by greedy method. DSDR selects the sentences that span the intrinsic subspace of the document sentence space so that it can cover (represent) the core information in the input documents.

3. Vertex cover based multi-document summarization system

In our work to extract the sentences for the summary, we used the concept of vertex cover problem in graph theory[11]. Vertex cover of a graph is a set of selected vertices such that every edge in the graph is incident to at least one of the selected vertices. Extractive multi-document summarization process selects finite set of relevant sentences that cover the entire concepts that is covered by all sentences in the input documents. Some conceptual similarities exist between multi-document summarization problem and the vertex cover problem. Multi-document summarization is performed by three steps: document transformation, graph modelling and sentence extraction by vertex cover algorithm.

We are developing a summarization system which generates the summary of related documents. Some important equivalent information may be available in more than one input documents participating in the summarization process. This can lead to the overlapping of information in the summary. Our multi-document summarization system addresses this problem also. This framework handles four important features of the summarization task such as relevancy, content coverage, diversity and length.

3.1. *The vertex cover problem*

Vertex cover problem is a popular problem in graph theory. Vertex cover of a graph is the subset of vertices of a graph such that each edge of the graph is incident to at least one vertex of the subset.

Formally vertex cover problem is defined as: Let $G = (V, E)$ be an undirected graph where V represents the set of vertices and E represents edges belonging to the graph. The goal is to find $S \subset V$ such that each edge $\{u, v\} \in E$: $u \in S \vee v \in S$. Vertex cover problem and multi-document summarization problem can be considered as a subset selection problem with some constraints. So we transformed multi-document summarization problem to a vertex cover problem by representing the input document as a graph. Sentences in the input documents represent vertices of the graph and an edge exists between every pair of vertices if the similarity between the corresponding sentences is greater than zero. So the graph constructed is a weighted graph and problem is transformed to a maximum weight vertex cover problem which selects the vertex set S with maximum information and covers all the edges. This task can be performed using three steps document transformation, graph modelling, and sentence extraction by vertex cover algorithm.

3.2. *Document transformation*

Let **D** be the web based input document set submitted for summarization. **D** is decomposed into a set of candidate sentences, $\mathbf{D} = \{s_1, s_{2,} s_3, \dots, s_n\}$ where s_i indicates the i^{th} sentence in the document set and *n* the total number of sentences in the document set. Candidate sentences are to be extracted from HTML or text documents, which contain information that is not interesting from summarization perspective. So the document set **D** is pre-processed using the techniques such as noise removal and sentence segmentation, stop word removal [16,17]and stemming using porter stemming algorithm [18,19].

Let $\mathbf{T} = \{t_1, t_2, t_3, \dots, t_m\}$ represents all the unique stemmed words obtained after the pre-processing of document set **D**, where m is the number of unique stemmed words. For further processing it is necessary to compute the similarity between the sentences in the document set **D**, which requires the appropriate representation of sentences. We used the vector space model (VSM), which is most popular representation for the text documents. In vector space model each sentence s_i is represented by a vector of dimension m as $\boldsymbol{s_i} = [tw_{i1}, tw_{i2}, tw_{i3}, \ldots, tw_{im}]$, where i =1,2,.. ,n and each element tw_{ij} indicates the weight of word t_j in the sentence s_i. Number of words actually present in each sentence will be very less compared to m, the size of the word set T, so most of the entries in the sentence vector may be zero.

Weight of each word in the sentence vector must indicate the correlation between the corresponding words and sentence. Therefore if the j^{th} word is absent in a pre-processed i^{th} sentence then the corresponding weight tw_{ij} is assigned zero otherwise a weight is computed using word frequency (*tf*) and inverse sentence frequency method (*isf*). This weight is a statistical measure used to evaluate how important a word is to a sentence in a collection of sentences. In *tf-isf* pattern *tf* indicates the frequency of occurrence of the word in the sentence and *isf* is a measure of how important a word in a sentences. If we consider *tf* alone for weight computation, frequently occurring words can highly influence the summarization process which is not desirable. The *tf-isf* balances the local *tf* and global *isf* in the sentence[20]. The weight tw_{ij} of j^{th} word in i^{th} sentence is computed as

$$tw_{ij} = tf_{ij} \times log\left(n/n_j\right) \tag{1}$$

where tf_{ij} indicates how many times word t_j occurs in sentence $\boldsymbol{s_i}$, n_j specifies the number of sentences in which the word t_j occurs, n denotes the total number of sentences in the input document set $\boldsymbol{D}$.

3.3 *Graph modelling*

An undirected weighted graph is constructed for the pre-processed input document set with n vertices, where n is the number of sentences in the input document set. An edge E_{ik} is maintained between every pair of vertices V_i and V_k if there is a similarity between the sentence $\boldsymbol{s_i}$ and $\boldsymbol{s_k}$. Weight of an edge E_{ik} is the similarity between the sentence $\boldsymbol{s_i}$ and $\boldsymbol{s_k}$. Similarity between the sentences is calculated using cosine similarity measure and Normalized Google Distance (*NGD*) based similarity measure[12]. We tested the performance of our system by computing the edge weight between the vertices using three methods such as cosine similarity; Normalized Google Distance based similarity and combination of both the methods. It is then observed that combination of both the methods gives better performance and it is used in this work.

Cosine similarity between the sentence vector $\boldsymbol{s_i}$ and $\boldsymbol{s_k}$ is calculated as

$$cossim(s_i, s_k) = \frac{\sum_{j=1}^{m} tw_{ij} tw_{kj}}{\sqrt{\sum_{j=1}^{m} tw_{ij}^2 * \sum_{j=1}^{m} tw_{kj}^2}}, \quad i, k = 1,2,..,n \tag{2}$$

where tw_{ij}, tw_{kj} indicates the weight of j^{th} word in the $\boldsymbol{s_i}$ and $\boldsymbol{s_k}$ sentence vectors respectively. *NGD* based similarity between the sentences vector $\boldsymbol{s_i}$ and $\boldsymbol{s_k}$ is calculated as in the following equation.

$$NGDsim(s_i, s_k) = \frac{\sum_{t_j \in si} \sum_{t_l \in sk} NGDsim(t_j, t_l)}{|s_i|.|s_k|} \tag{3}$$

where

$$NGDsim(t_k, t_l) = exp\left(-NGD(t_j, t_l)\right) \tag{4}$$

is the *NGD* based similarity between the j^{th} word t_j and l^{th} word t_l.

$$NGD(t_j, t_l) = \frac{max\{log(f_j), log(f_l)\} - log(f_{jl})}{logn - min\{log(f_j), log(f_l)\}} \tag{5}$$

where f_j is the number of sentences containing the word t_j , f_{jl} i_ the number of sentences containing both the words t_j and t_l. WE_{ik}, weight of an edge E_{ik} is calculated as [21]

$$WE_{ik} = 0.5 \times cossim(s_i, s_k) + 0.5 \times NGDsim(s_i, s_k) \quad (6)$$

Once the weights are computed summary sentences are extracted using vertex cover algorithm.

3.4 *Summary sentence selection using vertex cover algorithm*

Once an undirected weighted graph is constructed for the entire input document set **D**, maximum weighted vertex cover algorithm is applied on the graph, to detect the important sentences that cover the entire concepts of the input documents. Associated with every vertex two variables are maintained 1) summary_flag - which indicates the presence of a vertex in the summary. Initially all are assigned 0, indicating that no vertex is included in the summary. 2) availability variable - determines whether the vertex is to be considered for further iteration. It is maintained to avoid redundant information from the summary. Sentences to be selected for the summary are identified by the algorithm given below.

Algorithm - Sentence selection using vertex cover

Let n be the number of sentences or vertices in the document set, S_i be a vector in the vector space model corresponding to the i^{th} stemmed sentences without stop words, D be a vector corresponding to the entire document set. E be edges available in the modelled graph $W_{i,j}$ be combined similarity between sentences / vertices S_i and S_j.

Input: Set of documents to be summarized.
Output: Sentences selected from input documents which are to be include in the summary
  Summary_length ← 0, Summary ← { }
  For i ← 1 to n
      Summary_flag(i) ← 0
      Availability (i) ← 1
      Sentence_score (i) ← combined similarity between S_i and D obtained using equation (6)
  End for
  Accept required_size
  While required_size > summary_length
    Selected_vertex ← vertex with maximum Sentence_score
    Summary_length ← Summary_length + number of words in the selected vertex
    Summary = Summary U $S_{seceted\_vertex}$
    E= E- $E_{selected\_vertex,}$ where $E_{selected\_vertex,}$ are edges incident on *selected_vertx*
    Summary_flag (selected_vertex) ← 1
    For i ←1 to n
      If (availability (i) = 1) and (combined Similarity (S_i, $S_{seceted\_vertex}$) >0.5)
          Availability (i) ← 0
    End for
    For i ←1 to n
      If (availability (i) = 1) then
          Sentence_score(i) = ∑ WE_{ik} where WE_{ik} is the weight of an edge E_{ik} Є E incident on vertex i, calculated using equation (6)
      End for
  End while

By the above algorithm a summary is generated with high relevance, with less redundancy and of limited length. Step 8 guarantees that sentences selected are relevant, step 10 minimizes the redundancy in the summary and step 4 and 6 limits the summary size to the required value.

4. Experimental results

Proposed system is tested using subset of Document Understanding Conferences (DUC) 2002 data set. DUC 2002 dataset of multiple documents written about the same topics and corresponding 200 words extractive summaries are used for the evaluation of our system..

In our experiment, all the documents in the document set are combined to get a single document after the removal of the HTML tags and separation of sentences from each document. Stop word removal and word stemming is performed as the pre-processing step to transform these sentences into a format suitable for the processing. Proposed algorithm is performed on these pre-processed sentences to extract the summary.

For the performance evaluation we used the ROUGUE-1.5.5 (Recall-Oriented Understudy for Gisting Evaluation) package[22]. ROUGE is accepted by DUC as the official evaluation metric for document summarization. It provides measures which automatically identifies the quality of the system generated summary with summary given in DUC data set or the human generated summary. ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU are the measures available to evaluate the quality of summarization[22].

$$ROUGUE-N = \frac{\sum_{S \in Sref} \sum_{N-gram \in S} Maxcount(N-gram)}{\sum_{S \in Sref} \sum_{N-gram \in S} Count(N-gram)} \tag{7}$$

Where the *Sref* specifies the reference summaries, *S is* a sentence in the reference summary, *N* indicates the length of N-gram, *Maxcout (N-gram)* is the maximum number of N-grams occurring in the generated summaries and reference summaries. An N-gram is a contiguous sequence of N items from a given sequence of text. The items can be syllables, letters, words or base pairs according to the application. N-gram overlap with N=1 is identical to cosine similarity. If N>1, N-gram overlap is more accurate than cosine similarity and it considers the ordering of words in a sentence.

ROUGUE-L and ROUGUE-W are the two extension of ROUG-N provided in the ROUGE tool kit[22]. ROUGUE-L computes the length of the longest common subsequence (LCS) between candidate (system generated) summary and the reference summary and ROUGUE-W consider the weighted LCS and are calculated as[22].

$$R_{LCS} = \frac{\sum_{i=1}^{n} LCS_U(rs_i, C)}{m}, P_{LCS} = \frac{\sum_{i=1}^{u} LCS_U(rs_i, C)}{n}, F_{LCS} = \frac{(1+B^2) R_{LCS} P_{LCS}}{R_{LCS} + B^2 P_{LCS}} \tag{8}$$

where R_{LCS} , P_{LCS}, F_{LCS} are recall, precision and F-measure in ROUGUE-L , *C* is the candidate summary which contains *n* words and rs_i , $i=1,2,...,u$ are sentences in a reference summary containing *m* words. $LCS_U(rs_i, C)$ is the LCS score of the union of longest common subsequence between reference sentence rs_i and candidate summary C.

Table1 ROUGUE F-Score evaluation

Methods	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W	ROUGE-SU4
Vertex Cover	0.30409	0.07059	0.259	0.269	0.08736
MMR	0.3032	0.0695	0.2030	0.2574	0.08459
DSDR	0.29692	0.06324	0.235	0.23529	0.0790

The performance of our method on DUC 2002 dataset is compared with Document Summarization based on Data Reconstruction (DSDR) and Maximum Marginal relevance (MMR) applied on the centroid methods. The ROUGUE score of these methods are given in Table 1. It is observed that our method generates better score than other two methods. The summary generated contains relevant sentences without redundancy.

5. Conclusion

We have developed an automatic graph based, multi-document summarization system which is applicable for both the single and multi-document summarization. Summary generated guarantees minimum redundancy, required length and covers main concepts of the document. Sentences are extracted based on cosine similarity and Normalized Google Distance based measure. Experiment is performed using DUC 2002 data set and the associated extracted summaries. Performance of the system is compared with centroid method where redundancy is controlled by MMR and DSDR methods using ROUGE Tool kit. Our method gives better score than other two methods. This system can be further improved by identifying new technique to calculate the edge weight in the graph which improves the relevance of the sentences selected for the summary.

References

1. Barzilay R., Elhadad N and McKeown. K. R. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*. 1980. vol.17. p. 35–55 .
2. Inderjeet Mani. Automatic summarization, *Kluwer Academic Publishers*; 2004.
3. Jen-Yuan Yeh , Hao-Ren K and Wei-Pang Yang. iSpreadRank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications*, vol. 35, no. 3, p. 1451-1462, October 2008.
4. Yatsko V, Shilov S and Vishniakov T. A Semi-automatic Text Summarization System. *In proceedings of the 10th International Conference on Speech and Computer* ; 2005. p. 283-288.
5. M Kutlu, C Cigir, I Cicekli. Generic text summarization for Turkish. *The computer Journal;* 2010. p. 1315-1323
6. Kaustubh Patil and Pavel Brazdil. SUMGRAPH: Text Summarization Using Centrality In The Pathfinder Network. *International Journal on Computer Science and Information Systems*; 2007. vol.2. no.1. p. 18-32.
7. Shiyan OU, Christopher SG Khoo, Dion H Goh. Design and development of a concept-based multidocument summarization system for research abstracts. *Journal of Information Science*; June 2008. vol. 34. no. 3. p. 308-326.
8. Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali. Swarm Based Features Selection for Text Summarization. *IJCSNS International Journal of Computer Science and Network Security*; January 2009. vol. 9. no.1.
9. Jade Goldstein, Vibhu Mittal, Jaime Carbonell, Mark Kantrowitzt. Multi-Document Summarization by Sentence Extraction. *NAACL-ANLP 2000 Workshop on Automatic summarization*. Seattle, Washington; 2000. vol. 4. p. 40 - 48.
10. Breck Baldwin , Thomas S Morton. Dynamic coreference-based summarization. *In Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*. Granaare DA, Spain; June 1998.
11. Thomas H, Kormen,Charles, E Leiserson, Ronald L Rivest. *Introduction to algorithms*. Prentice Hall India; p. 949-950
12. Cilibrasi, Rudi L, Paul MB Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*; 2007. vol 19. no.3. p. 370-383.
13. Dragomir R. Radev A, Hongyan Jing, Magorzata Sty and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Manageme;* 2004. vol. 40. no.6. p. 919–938.
14. Carbonell, Jaime, Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM*; 1998.
15. Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. Document Summarization Based on Data Reconstruction. *In AAAI*; 2012.
16. Florian Boudin, Juan Manuel Torres Moreno. NEO-CORTEX: A Performant User-Oriented Multi-Document Summarization System. *Lecture Notes in Computer Science*. Springer; 2007. Vol 4394. p.551-562.
17. Pant, G, Srinivasan, P, Menczer, F. Crawling the Web, Web Dynamics: Adapting to Change in Content, Size, Topology and Use. *Springer- Verilog*. New York; November 2004. p. 153-178.
18. Lovins, J B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*; 1968. vol 11. p. 22-31.
19. C J Van Rijsbergen, S E Robertson, M F Porter. New models in probabilistic information retrieval. London: British Library. *British Library Research and Development Report;* 1980. no. 5587
20. Salton, Gerard, Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management;* 1988. p. 513-523.
21. Alguliev, Rasim M., et al. MCMR: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications.* 2011 p. 14514-14522.
22. Lin C Y. ROUGE: A package for automatic evaluation summaries. *In Proceedings of the workshop on text summarization branches out*.
23. DUC. <http://duc.nist.gov>
24. ROUGE Toolkit. <http://berouge.com/default.aspx>