ESUM: An Efficient System for Query-Specific Multi-document Summarization

C. Ravindranath Chowdary and P. Sreenivasa Kumar

Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai 600 036, India
{chowdary,psk}@cse.iitm.ac.in

Abstract. In this paper, we address the problem of generating a query-specific extractive summary in a an efficient manner for a given set of documents. In many of the current solutions, the entire collection of documents is modeled as a single graph which is used for summary generation. Unlike these approaches, in this paper, we model each individual document as a graph and generate a query-specific summary for it. These individual summaries are then intelligently combined to produce the final summary. This approach greatly reduces the computational complexity.

Keywords: Efficient summarization, Coherent and Non-redundant summaries.

1 Introduction

Text summarization has picked up its pace in the recent years. In most of the summarizers, a document is modeled as a graph and a node will get high score if it is connected to the nodes with high score. Extractive, centrality based approaches are discussed in [1,2,3]. Degree centrality is discussed in [1] and eigenvector centrality is discussed in [2,3]. Eigenvector centrality of a node is calculated by taking into consideration both the degree of the node and the degree of the nodes connecting to it. Query specific summary generation by computing node scores iteratively till they converge is discussed in [4]. So, the node scores are computed recursively till the values converge. Generating information without repetition is addressed in [5]. These systems do not explicitly address the issue of efficiency of the system in terms of computational complexity, coherence and non-redundancy of the summary generated. All these issues are addressed in our approach. To improve the efficiency of generating multi-document query-specific summaries, we propose a distributed approach where summaries are computed on individual documents and the best of these summaries is augmented with sentences from other summaries.

2 The ESUM System

2.1 Terminology

To summarize a document, we model it as a graph. Each sentence in the document is considered as a node and an edge is present between any two nodes if the

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 724–728, 2009.

similarity between the two nodes is above a threshold. *Similarity* is calculated as given below:

$$sim(\overrightarrow{n_i}, \overrightarrow{n_j}) = \frac{\overrightarrow{n_i}.\overrightarrow{n_j}}{|\overrightarrow{n_i}||\overrightarrow{n_j}|} \quad (1)$$

where $\overrightarrow{n_i}$ and $\overrightarrow{n_j}$ are term vectors for the nodes n_i and n_j respectively. The weight of each term in $\overrightarrow{n_i}$ is calculated as $tf * isf$. tf is term frequency and isf is inverse sentential frequency. The quality of a summary is measured in terms of many features- few of them are coherence, completeness, non-redundancy. A summary is said to be coherent if there is a logical connectivity between sentences. A summary is complete if all the query terms are present in it. A summary is said to be non-redundant if there is a minimum or no repetition of information.

2.2 Description of Our Model

We use a method which is similar to the one proposed in [4] for calculating the score of a node with respect to a query term. Initially each node is assigned a score of one and then Equation 2 is iterated till the scores of the nodes converge. The node scores for each node w.r.t each query term $q_i \in Q$ where $Q = \{q_1, q_2, ..., q_t\}$ are computed using the following equation.

$$w_{q_i}(s) = d\frac{sim(s, q_i)}{\sum_{m \in N} sim(m, q_i)} \; + \; (1-d) \sum_{v \in adj(s)} \frac{sim(s, v)}{\sum_{u \in adj(v)} sim(u, v)} w_{q_i}(v) \quad (2)$$

where $w_{q_i}(s)$ is node score of node s with respect to query term q_i, d is bias factor and N is the set of all the nodes in the document. First part of equation computes relevancy of nodes to the query and the second part considers neighbours' node scores. The bias factor d gives trade-off between these two parts and is determined empirically. For a given query Q, node scores for each node w.r.t each query term are calculated. So, a node will have a high score if: 1) it has information relevant to the query and 2) it has neighbouring nodes sharing query relevant information.

Contextual Path(CPath). For each query term, a tree is explored from each node of the document graph(DG). The exploration of the tree will continue till certain depth or till the node containing query word is reached, which ever is earlier. The tree so formed is called Contextual Path($CPath$). The definition of CPath is as follows:

Definition 1. ***Contextual Path(CPath):*** *A $CPath_i = (N_i, E_i, r, q_i)$ is defined as a quadruple where N_i and E_i are set of nodes and edges respectively. q_i is i^{th} term in the query. It is rooted at r with at least one of the nodes having the query term q_i. Number of children for each node is one except for r. All the neighbours (top k similar nodes) of r are included in $CPath$. But $CPath$ is empty if there is no node with query term q_i within depth d.*

A $CPath$ is constructed for each query term of Q. $CPath$s formed from each node in DG are assigned a score that reflects the degree of coherence and information

richness in the tree. *CPathScore* rooted at node r for a query term q is calculated as given in Equation 3.

$$CPathScore_{q_i} = \beta w_{q_i}(r) \; + \sum_{\substack{(u,v)\in CPath_{q_i} \\ u \; is \; parent \; of \; v}} [\frac{\alpha w(e_{u,v}) + \beta w_{q_i}(v)}{(level(v)+1)^2}] \qquad (3)$$

Where $\alpha = \frac{a}{b} * 1.5$, here a is average of top three node weights among the neighbours of u excluding parent of u and b is maximum edge weight among nodes incident on u. $w(e_{u,v})$ is the score of edge (u, v) and $w_{q_i}(v)$ is node score of v with respect to the query term q_i. $level(v)$ is the level of v in the $CPath$. α and β values determine the importance given to edge weights(coherence) and node weights(relevance) respectively. Equation 3 is used to calculate the CPath score. It is the linear sum of node scores and edge scores of the CPath. This measure ensures the highest scored CPath is compact and highly coherent.

Definition 2. ***Summary Graph(SGraph).*** *For each node r in DG, if there are t query terms, we construct a summary graph $SGraph = (N^{'}, E^{'}, Q)$ where $N^{'} = \cup_{i=1}^{t} N_i$, $E^{'} = \cup_{i=1}^{t} E_i$ where N_i and E_i are the sets of nodes and edges of $CPath_i$ rooted at r respectively and $Q = \{q_1, q_2, ..., q_t\}$*

For each node r in DG, if there are t query terms $Q = \{q_1, q_2, ..., q_t\}$, score of the *SGraph SG* is calculated using Equation 4.

$$SGraphScore = \frac{1}{\sqrt{size(SG)}} \sum_{q\in Q} CPathScore_q \qquad (4)$$

Here, $CPathScore_q$ is the score of $CPath_q$ rooted at r. The summary graph is constructed for each node in DG and the highest scored one among them is selected as the candidate summary for the DG. Let SG_1, SG_2, ...SG_n be the candidate summaries of n DGs respectively. We include the highest scored summary say SG_i among the n summaries into final summary. Now, we recalculate the score of each node in the remaining $n-1$ candidate summary graphs using the Equation 5 and include the highest scored node into the final summary. The above step is repeated till the user specified summary size is reached.

$$\underset{i}{Max}\{(\lambda \sum_{1\le k\le t} w_{q_k}(n_i)) - (1-\lambda)\underset{j}{Max}\{sim(n_i, s_j)\}\} \qquad (5)$$

In the Equation 5, n_i is a node in *RemainingNodes* and s_j is a node in *final summary*. This equation gives us the maximum scored node from *RemainingNodes* after subtracting similarity score from the node in *final summary* with which it has maximum similarity. This method of calculating the score assures us that the selected node is both important and the information it contributes to the *final summary* is less redundant. The equation is inspired by MMR-Reranking method which is discussed in [5]. For a set of documents which are related to a topic and for the given query, we generate a summary which is non-redundant, coherent and query specific. Non-redundancy is ensured by the way we are selecting the nodes to be added into the *final summary*, i.e., the use of Equation 5. Query specificity is ensured by the way in which we assign scores to the nodes.

3 Experimental Results

We have evaluated our system on DUC 2005 corpus[1]. The values of variables are as follows - bias factor d is fixed to 0.85 in Equation 2(based on [4]), λ is fixed to 0.6 in Equation 5(based on [5]), the values of other variables are fixed based on the experimentation. The system was developed in Java. $Fanout$ indicates number of children explored from each node in $CPath$ construction. The values for β and $Fanout$ are set to 1 and 3 respectively. Table 1 shows the comparison between our system and the best performing systems of DUC 2005 in terms of macro average. 25 out of 50(DUC has 50 document clusters) summaries generated by our system outperformed system-15 in terms of ROUGE scores. $SIGIR08$ [6] is the latest summarizer and ESUM outperformed it. This clearly demonstrates that the quality of summaries generated by the ESUM system is comparable to the best of DUC 2005 systems and the latest summarizer [6]. Further, on the time complexity count the ESUM system is much better compared to other systems. The typical integrated graph based algorithm has complexity $O((\sum l_i)^2)$. Because ESUM constructs graphs only for individual documents, the time complexity here is $O(\sum l_i^2)$. l_i denotes the size of the i^{th} document. Evidently, ESUM approach is computationally superior and does not compromise on the quality of results generated. MEAD [7] is a publicly available summarizer that follows integrated graph approach. On average for a cluster with 25 documents, ESUM performs more than 80 times faster compared to MEAD system. On the same platform, ESUM summarizes in 20 seconds and MEAD in 29 minutes. Since our approach is distributed, as the number of input documents increase, ESUM scales near linearly whereas other systems suffer dramatic increase in running time because of their non-distributive nature.

Table 1. Results on DUC 2005(*macro average*)

Systems	R-1	R-2	R-W	R-SU4
ESUM	0.37167	0.07140	0.08751	0.12768
SIGIR08	0.35006	0.06043	0.12266	0.12298
System-15	0.37515	0.07251	0.09867	0.13163
System-17	0.36977	0.07174	0.09767	0.12972

4 Conclusions

The paper proposed a solution to the problem of query-specific multi-document extractive summarization. The proposed method generates summaries very efficiently and the generated summaries are coherent to read and do not have redundant information. The key and important feature of the solution is to generate summaries for individual documents first and augment them later to produce the final summary. This distributed nature of the method has given significant

[1] http://www-nlpir.nist.gov/projects/duc/data.html

performance gains without compromising on the quality of the summary generated. Since in terms of computational complexity the proposed system is well ahead of other systems, the solution is an efficient summary generating system.

References

1. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Inf. Process. Manage. 33(2), 193–207 (1997)
2. Erkan, G., Radev, D.R.: LexPageRank: Prestige in multi-document text summarization. In: Proceedings of EMNLP, Barcelona, Spain, July 2004, pp. 365–371. ACL (2004)
3. Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Barcelona, Spain, p. 20. ACL (2004)
4. Otterbacher, J., Erkan, G., Radev, D.R.: Using random walks for question-focused sentence retrieval. In: HLT 2005: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, ACL, pp. 915–922. ACL (2005)
5. Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR, Melbourne, Australia, pp. 335–336. ACM, New York (1998)
6. Wang, D., Li, T., Zhu, S., Ding, C.: Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In: SIGIR 2008: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, pp. 307–314. ACM, New York (2008)
7. Radev, D.R., Jing, H., Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In: NAACL-ANLP 2000 Workshop on Automatic summarization, Seattle, Washington, pp. 21–30. ACL (2000)