Published in final edited form as:
Proc IEEE WIC ACM Int Conf Web Intell Intell Agent Technol. 2009 October 10; 1(2009): 267–272.
doi:10.1109/WI-IAT.2009.46.

Summarizing Documents by Measuring the Importance of a Subset of Vertices within a Graph

Shouyuan Chen,
Dept. Computer Science and Technology, Beijing 10084, China, chenshouyuan@gmail.com

Minlie Huang, and
Dept. Computer Science and Technology, National Laboratory for Information Science, and Technology, Beijing 100084, China, aihuang@tsinghua.edu.cn

Zhiyong Lu
National Center for Biotechnology Information, National Library of Medcine, Bethesda, 20894, USA, luzh@ncbi.nlm.nih.gov

Abstract

This paper presents a novel method of generating extractive summaries for multiple documents. Given a cluster of documents, we firstly construct a graph where each vertex represents a sentence and edges are created according to the asymmetric relationship between sentences. Then we develop a method to measure the importance of a subset of vertices by adding a super-vertex into the original graph. The importance of such a super-vertex is quantified as super-centrality, a quantitative measure for the importance of a subset of vertices within the whole graph. Finally, we propose a heuristic algorithm to find the best summary. Our method is evaluated with extensive experiments. The comparative results show that the proposed method outperforms other methods on several datasets.

Keywords

document summarization; centrality; graph; diversity

I. INTRODUCTION

With the rapid growth of information on the Internet, people need to get a concise version of a large set of articles in a short time. As the most well-known event, NIST has organized the DUC competition on document summarization since 2000. This paper follows the line of extractive summarization whose task is to choose a set of sentences from the original documents.

A good summary is expected to preserve the topic information contained in the documents as much as possible, and at the same time to contain as little redundancy as possible, known as information richness and diversity, respectively. The requirement raises a fundamental problem: how important will a selected summary be to represent the whole documents? Despite a significant body of work in summarization, this issue has yet been systematically studied in the past. Existing systems such as MEAD [1], and many graph-based methods [2–5], proposed means of finding a set of sentences to form summaries, but none of them could answer the question.

Within the graph-based modeling framework, the previous problem can be transferred to that how important a subset of vertices will be to the whole graph? Centrality is a good

metric to determine the relative importance of a vertex within a graph. The centrality of a vertex could answer some questions such as how important a person is within a social network. Or in summarization, it can answer how salient a sentence is for a document set. Although there have been various graph-based summarization methods, however, to the best of our knowledge, there is still no work that could measure the importance of a subset of vertices within a graph.

In this paper, we propose a novel method to the problem. Our method can measure the importance of a subset of vertices within a graph. The importance is quantified by the super-centrality of a super-vertex added into the original graph. The super-centrality measures how important a subset of vertices will be for the whole graph. Furthermore, the most important subset can be found with the aid of the heuristic algorithm as described in this paper. When our method is adapted to the summarization task, it is shown to achieve better results than those from comparable methods.

This paper is organized as follows: we survey related work in the next section; in Section III we introduce the method of measuring the importance of a subset of vertices. In Section IV, extensive experiments are described to justify our method. Finally we make our conclusion in Section V.

II. RELATED WORK

There are several well known *centrality* measures in graph theory, including degree centrality, betweenness, closeness, and eigenvector centrality [6–7]. The famous PageRank algorithm can be regarded as an application of eigenvector centrality. Our method can measure the importance of a subset of vertices, which will largely extend the use of *centrality*.

A key step of summarization using graph-based algorithms is to select the top ranking vertices. Most methods have to employ a redundancy removal module because they did not treat the selected vertices as a whole. The most well-known redundancy removal methods are maximum marginal relevance (MMR) [8], cross-sentence informational subsumption [9], and diversity penalty. The central idea of these methods is to penalize redundancy by lowering a sentence's rank if it is similar to sentences that are already selected. In [10], an absorbing random walk was proposed to improve diversity based on an initial ranking list. However, all these methods treat sentence ranking and diversity ranking separately, sometimes with heuristic procedures and parameters. Different from previous methods, we do not use any separate processes to remove redundancy. Instead, our method can exclude redundancy while identifying a subset of vertices for summarization.

III. MEASURING THE IMPORTANCE OF A SUBSET OF VERTICES WITHIN A GRAPH

A. Constructing of a graph

Given a set of sentences $S = \{s_1, s_2, \ldots, s_n\}$, we construct a basic graph $G_0 (V,E)$ in two steps:

1. For each sentence s_i, we create a vertex $v_i \in V(G_0)$;
2. For each pair of vertices v_i and v_j, we create a directed edge $(v_i, v_j) \in E (G_0)$ from v_i to v_j, and each edge is associated with an asymmetric weight as follows:

$$w(v_i \rightarrow v_j) \triangleq \min\{\frac{\overrightarrow{s_i} \bullet \overrightarrow{s_j}}{\|\overrightarrow{s_i}\|}, 1\} \quad (1)$$

where $\overrightarrow{s_i}$ and $\overrightarrow{s_j}$ is a vector representation of v_i and v_j, respectively. The weight can be viewed as the ratio of the information shared by v_i and v_j to the information contained in v_i.

Random walk provides a theoretical framework of our proposed algorithm. Thus, it is proper to translate the basic graph G_0 *(V,E)* to a stochastic graph G_1*(V,E)*. In this paper, the transformation is a normalization of edge weights with an addictive Laplacian smooth that guarantees a well-formed (both irreducible and aperiodic) graph.

B. Computing the centrality for single vertices

The centrality of vertex *u* within graph G_1*(V,E)* can be computed as follows:

$$\pi_u(G_1) = \sum_{x \in V_u(G)} p(u \mid x; G_1)\pi_x(G_1) \quad (2)$$

where $V_u(G_1) = \{x | x \in V(G_1), (x,u) \in E(G_1)\}$, and all centralities sum up to 1.

$$\sum_{x \in V(G)} \pi_x(G_1) = 1 \quad (3)$$

This recursive equation can be calculated via a power iteration method on graph G_1. Notice that graph G_1 is both irreducible and aperiodic. Hence, after applying the knowledge of random process on a graph, we have:

$$\lim_{n \to \infty} p_{uv}^{(n)} = \pi_v(G_1) \quad (4)$$

where $p_{uv}^{(n)}$ represents the probability of traveling from any *u* to *v* in *n* steps.

C. Adding a super-vertex

Many tasks require selecting a subset of vertices within a graph. In summarization, we need to select several sentences to represent the whole document set. In a social network, we need to find a list of people that represent active groups. In addition to computing the centrality of a single vertex, it is a key issue to compute the centrality of a set of vertices in those tasks. In this regard, we propose to approach the problem by adding a super-vertex into the original graph.

Definition 1. (Super-vertex) *Given a basic graph* $G_0(V, E)$ *with a set of vertices* $V = \{v_1, v_2, \ldots, v_n\}$*, we construct an extended graph* $G_0'(V', E')$ *and its corresponding stochastic graph* $G_1'(V', E')$*, by adding a super-vertex C where* $V' = V \cup \{C\}$*. The super-vertex C = Subset(V) consists of several vertices from* $V(G_0)$ *and it has edges:*

$$\forall u \in V(G_0)$$
$$w(C \to u)=1, w(u \to C)=\max_{v_i \in Subset(V)} w(u \to v_i) \quad (5)$$

The transition probability matrix of $G_1^{'}$ can be computed with by normalization technique. We denote graph $G_1^{'}(V', E')$ as $G_1^{'}(C)$ with respect to the newly added super-vertex.

The super-vertex is a mixture of original vertices. Thus the information contained by a super-vertex can be viewed as a summation of the information contained by all the vertices represented by the super-vertex. On the other hand, the super-vertex has an edge connecting to any other vertex with a uniform weight of one. Ideally, the original graph and its centrality distribution should not be remarkably affected when a super-vertex is added.

It is worth noting that the *max(.)* function inherently avoids redundancy when we select a subset of vertices to represent the whole graph. This will be discussed further in Section III-D.

At the end of this section, let us summarize some notations used so far:

1. $G_0(V,E)$: the basic graph;
2. $G_1(V,E)$: the corresponding stochastic graph for G_0;
3. $G'_0(V',E')$ ($G'_0(C)$ for short): the extended graph of $G_0(V,E)$ with an added super-vertex C;
4. $G'_1(V,E)$ ($G'_1(C)$ for short): the corresponding stochastic graph for $G'_0(V',E')$.

D. Measuring the importance of a subset of vertices by super-centrality

Given a super-vertex $C=Subset(V)$ on graph G_1',, we use the following equation to compute its centrality:

$$\pi_C(G_1')=\sum_{u \in V(G_1^{'})-\{C\}} p(C \mid u;G_1')\pi_u(G_1') \quad (6)$$

Note that $\pi_C(G_1^{'})$ is the probability of arriving at super-vertex C during a random walk on graph G_1'. This is the same meaning as the centrality of individual vertices. For simplicity, we term the centrality of a super-vertex *super-centrality*. The super-centrality measures how important a super-vertex is for the graph, and it measures a subset of vertices as a whole.

1) Exact Algorithm—The first approach to computing the super-centrality is an exact algorithm, as used to computing the centrality of single vertices. This approach simply adds the super-vertex, then calculates the corresponding stationary distribution of this super-vertex which is the super-centrality by power iteration method.

An obvious optimization is to set the initial distribute as the stationary distribution of the last computation of the super-vertex. As the overall structure of the graph varies slightly, the algorithm will converge in a few steps. The algorithm is an exact algorithm meaning that it will compute the precise centrality of the super-vertex C. However, the time complexity is still high $O(n^2)$.

2) Approximation Algorithm—To be more efficient in computing the super-centrality, an approximating algorithm with a much better time complexity is developed. It is based on the assumption that the addition of super-vertex *C* would only slightly change the distribution of other vertices. And thus we do not need a power iteration here, instead the super-centrality can be calculated by a simple ratio: $\pi_C(G_1^{'})=\frac{\pi_C^{(0)}(G_1^{'})}{1+\pi_C^{(0)}(G_1^{'})}$.

This algorithm is very fast because there is no iteration process with graph $G_1^{'}$. And its precision is high in practice, see experiments in Section IV-B.

3) Heuristic algorithm—The approximation algorithm is still inefficient for large scale corpus. Assuming that we have *N* vertices on a graph, if we need to select *k* vertices to represent the original graph, the worst algorithm has to try $\binom{n}{k}$ times. Obviously, there is a combinatorial explosion. Thus, a more efficient algorithm is desired.

In this paper, we propose a heuristic approach. The idea behind is straightforward: for each step, we extract a vertex that achieves the greatest increase of super-centrality $\pi_C(G_1^{'})$; then append it into the summarization set *S*, until the word limit is exceeded.

Our algorithm has inherently avoided redundancy when selecting a subset of representative vertices. In each step of iteration, we select the vertex that can maximize $\Delta\pi_c(G_1^{'})$. If a new sentence s_i is redundant to one item of the current set $S = \{s_1, s_2, \ldots, s_{i-1}\}$, the two weights $w(v \rightarrow s_i)$ and $w(v \rightarrow S)=\max_{x\in S}\{w(v \rightarrow x)\}$ will be almost the same (see the definition of edge weight in Formula (1)). Thus such an s_i has no contribution to function $\Delta\pi_S(s_i)$and it is not likely to be selected. This is why we use the *max(.)* function in formula (2). This is also the reason that our method can inherently avoid redundancy (or improve diversity) to select a subset of vertices to represent the whole graph.

IV. EXPERIMAL RESULTS

A. Data preparation and evaluation metric

The official datasets from DUC 2006 and TAC 2008 were used in our experiments. The task of DUC 2006 required participants generate a 250-word summary for each topic, while TAC 2008 demanded for a short 100-word summary.

We evaluated our method by comparing the generated summaries to the hand-written reference summaries under the ROUGE-1 measure [11]. *ROUGE* measures the quality of a summary by counting the overlapping units such as the *n*-gram, word sequences and word pairs between the generated summary and the reference summary. We use ROUGE-1 as the evaluation metric.

B. Comparing the exact and approximating algorithms of computing super-centrality

In this experiment, we will prove that our approximating algorithm of computing super-centrality is a good approximation of the exact algorithm. The experiment is conducted to calculate the relative error between the exact algorithm and approximate version on 50 random selected topics.

Experiments show that the average relative error of approximation algorithm is no more than 0.1% against the exact one. Thus, the approximating algorithm seems to be practically useful because of its precision to the exact algorithm and computational efficiency. As a result, we used the approximating algorithm hereafter to compute the super-centrality of a super-vertex hereafter.

C. In comparison to the state-of-the-art methods

We compare our method with several state-of-the-art methods. We implemented two graph based summarization algorithms:

1. LexRank [3]. It aims to extract salient sentences. This algorithm is based on the eigenvector centrality. We select the top sentences within the length limit as the final summaries.
2. GRASSHOPPER [10]. This algorithm is to tune the extracted vertices to be absorbing state in a random walk.

The implement algorithms, including ours, LexRank and GRASSHOPPER are built on same graph structures, in which edge weights are computed by LSI with parameter $k = 200$ [12]. The difference only exists in computing the centralities and in the process of selecting the final sentences.

Figure 1 show that our method achieved the highest ROUGE-1 Recall scores among all the tested systems (Precision and F-Score has a similar rank). Note that we did not enable the redundancy removal module for the LexRank algorithm in these experiments. Instead, we demonstrate this separately in the subsequent section.

We also compared our results with those submitted to DUC 2006 and TAC 2008, respectively. Our system ranked 7^{th} place among 71 submits of TAC2008 and 6^{th} among 35 submits of DUC2006. Further, our results are comparable to those of other best systems.

D. Avoiding redundancy

An innovative aspect of our algorithm lies in its ability to remove redundancy while selecting representative sentences. Thus in this part, we justify this innovation by comparing our method with the most common MMR method (other diversity penalty procedures have almost the same processes) for redundancy removal.

MMR was employed in this experiment as a post-processing system for the LexRank approach. The penalty factor λ was tuned with a step of 0.1. From Figure 2, we can see that (1) our method outperformed the LexRank method with a MMR module; (2) the performance of the MMR algorithm depended heavily on the penalty factor; and (3) the best factor relied largely on the dataset used (0.9 for the TAC 2008 dataset while 0.8 for the DUC 2006 dataset); Thus the best factor selected for one dataset might not be optimal for another one.

V. CONCLUSION AND FUTURE WORK

This paper presents a novel method of summarizing multi-documents by measuring the importance of a subset of vertices within a graph. After introducing a super-vertex into the graph, an exact and approximating algorithm is exploited to compute the importance of a super-vertex. The most important super-vertex can be found with the aid of a heuristic algorithm. The method is justified by experimental results on the DUC 2006 dataset and TAC 2008 dataset when compared with other state-of-the-art approaches.

Our future work is to apply the method to other problems such as finding representative persons in a social network.

Acknowledgments

The work is supported in part by NSFC project No. 60803075, Chinese 973 Project No. 2007CB311003. ZL is supported by the Intramural Program of the National Institutes of Health. We also want to thank Prof. Xiaoyan Zhu for her kind support.

REFERENCE

1. Radev DR, Jing HY, Stys M, Tam D. Centroid-based summarization of multiple documents. Information Processing and Management 2004;40:919–938.
2. Mani I, Bloedorn E. Summarizing Similarities and Differences among Related Documents. Information Retrieval 1999;1(1):35–67.
3. Erkan G, Radev D. LexRank: Graph-based Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research 2004;Vol 22
4. Mihalcea, R.; Tarau, P. A language independent algorithm for single and multiple document summarization. Proceedings of IJCNLP. 2005;
5. Mihalcea, R.; Tarau, P. TextRank: Bringing order into texts. EMNLP'04; 2004.
6. Brandes, U.; Erlebach, T. Network analysis: methodological foundations. Springer, lecture notes on computer science; 2005. 2005
7. Newman M. The Mathematics of Networks. The New Palgrave Encyclopedia of Economics. 2007
8. Carbonell, J.; Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. SIGIR'98; 1998.
9. Radev, D. A common theory of information fusion from multiple text sources, step one: Cross-document structure. Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue; 2000.
10. Zhu X, Goldberg A, Gael JV, Andrzejewski D. Improving Diversity in Ranking using Absorbing Random Walks. HLT-NAACL 2007:97–104.
11. Lin, CY.; Hovy, E. HLT-NAACL. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics; p. 71-78.
12. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of the American Society for Information Science 1999 January 7;Vol. 41(No. 6): 391–407.

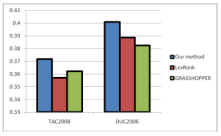

Figure 1.
ROUGE-1 Recall results on the TAC 2008 and DUC2006 dataset.

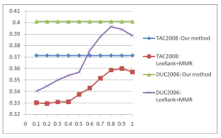

Figure 2.
Our method vs. LexRank with MMR.