Hierarchical Graph Summarization: Leveraging Hybrid Information through Visible and Invisible Linkage

Rui Yan[1], Zi Yuan[2], Xiaojun Wan[3], Yan Zhang[1,⋆], and Xiaoming Li[1]

[1] School of Electronics Engineering and Computer Science, Peking University, China
[2] School of Computer Science and Engineering, Beihang University, China
[3] Institute of Computer Science and Technology, Peking University, China
{r.yan,wanxiaojun,lxm}@pku.edu.cn,
ziyuan@cse.buaa.edu.cn, zhy@cis.pku.edu.cn

Abstract. Graph-based ranking algorithm has been recently exploited for summarization by using sentence-to-sentence relationships. Given a document set with linkage information to summarize, different sentences belong to different documents or clusters (either *visible* cluster via anchor texts or *invisible* cluster by semantics), which enables a hierarchical structure. It is challenging and interesting to investigate the impacts and weights of source documents/clusters: sentence from important ones are deemed more salient than the others. This paper aims to integrate three types of hierarchical linkage into traditional graph-based methods by proposing Hierarchical Graph Summarization (HGS). We utilize a hierarchical language model to measure the sentence relationships in HGS. We develop experimental systems to compare 5 rival algorithms on 4 instinctively different datasets which amount to 5197 documents. Performance comparisons between different system-generated summaries and manually created ones by human editors demonstrate the effectiveness of our approach in ROUGE metrics.

Keywords: Summarization, Hierarchical Graph, Visible and Invisible Linkage.

1 Introduction

In the era of information explosion, people need new information to update their knowledge whilst information on Web is updating extremely fast. Multi-document summarization has been proposed to address such dilemma by producing a summary delivering the majority of information content from a document corpus, and the short summary is necessarily helpful to facilitate users to quickly understand the large number of documents. Automated multi-document summarization has drawn much attention in recent years. In the communities of information retrieval and natural language processing, a series of conferences on automatic text summarization have advanced the summarization techniques and produced a couple of experimental online systems.

Graph-based ranking algorithms have been recently exploited for summarization by making use of sentence-to-sentence relationships and played an important role with the exponential document growth on the Web. In general, traditional graph summarization

⋆ Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 97–108, 2012.

utilizes plain linkage among sentences without considering higher-level information beyond the sentence-level information, which is insufficient. Given a document set with linkage information to summarize, different sentences belong to different documents and clusters, either clustered by **visible** linkage (e.g., anchor texts) or **invisible** linkage (e.g., semantic cohesion), which enables a hierarchical text structure. It is challenging and interesting to investigate the impacts and weights of source documents/clusters: different documents and clusters usually have different importance for users to understand the document set. Sentence from important documents/clusters are deemed more salient than the trivial ones. In brief, simultaneous consideration of three-layer hierarchical linkage has not been investigated under a unified framework.

In order to address above insufficiency, we aim to model these three levels of hierarchical linkage, i.e., sentence-to-sentence, sentence-to-document and document-to-cluster relationships, into traditional graph-based summarization, and we name this approach as Hierarchical Graph Summarization (HGS). We propose a hierarchical language model to measure the sentence relationships for the ranking process in HGS. Document/cluster-level information through visible and invisible linkage is used for smoothing: neighboring text information is proved to be useful [11]. We will first investigate the presence of visible and invisible linkage for clustering.

Visible Linkage. A web document is connected to other web documents by explicit links via anchor texts, which are denoted as visible linkage.

Invisible Linkage. A web document is connected to other web documents through implicit semantic coherence, denoted as invisible linkage.

The contributions of this paper are as follows:

• The **1st contribution** is to utilize the instinctively explicit linkage among web documents, which is a natural understanding of enormous web data organization. We distinguish such visible linkage from invisible linkage by semantic cohesion and utilize both information into clustering.

• The **2nd contribution** is to incorporate a three-level hierarchical linkage structure into a unified language smoothing model, which is used to measure sentence relationships by utilizing both document-level and cluster-level information simultaneously.

We start by reviewing previous work in Section 2. In Section 3 we describe the basic graph summarization and describe our proposed HGS in Section 4. We conduct empirical evaluations in Section 5, including performance comparisons and result discussion. Finally we draw conclusions in Section 6.

2 Related Work

Multi-document summarization (MDS) has drawn much attention in recent years. In general, MDS can either be extractive or abstractive. The former assigns salient scores to semantic units (e.g. sentences, paragraphs) of documents indicating the importance and then extracts top ranked ones, while the latter demands information fusion(e.g. sentence compression and reformulation). Here we focus on extractive summarization.

To date, various extraction-based methods have been proposed for generic multi-document summarization. MEAD [3] is an implementation of the centroid-based

method that scores sentences based on features such as cluster centroids, position, and TF.IDF, etc. NeATS [6] adds new features such as topic signature and term clustering to select important content. Themes (or topics, clusters) in documents have been discovered and used for sentence selection [10,14,13].

Most recently, the graph-based ranking methods have been proposed to rank sentences/passages based on "votes" or "recommendations" between each other. TextRank [9] and LexPageRank [2] use algorithms similar to PageRank and HITS to compute sentence importance. Cluster information such as document-level information has been incorporated in the graph model to better evaluate sentences [12].

Generally, summarization considers content characteristics such as coverage, diversity [1,17,5], and all these characteristics require a calculation of sentence linkage measurement. To the best of our knowledge, currently, neither the instinctively visible linkage of anchor texts from web document organizations is utilized for summarization, nor the three-layer hierarchical linkage has been investigated simultaneously in a unified language model to measure sentence relationships. HGS approach can naturally and simultaneously take into account these two advantages in graph-based summarization.

3 Basic Graph Summarization

The basic graph summarization is essentially a way of deciding the importance of a vertex within a linkage graph based on global information recursively drawn from the entire graph, using the Markov Random Walk Model (MRW). The basic idea is that of "voting" or "recommendation" between the vertices, where each vertex is a sentence. A link between two vertices is considered as a vote cast from one vertex to the other vertex. The score associated with a vertex is determined by the votes that are cast for it, and the score of the vertices casting these votes.

Formally, given a document set D, let $G = (V, E)$ be a graph to reflect the relationships between sentences in the document set, as shown in Figure 1 (Part A). V is the set of vertices and each vertex s_i in V is a sentence in the document set. E is the set of edges, which is a subset of $V \times V$. Each edge e_{ij} in E is associated with an affinity weight $f(s_i \rightarrow s_j)$ between sentences s_i and s_j ($i \neq j$). Sentence s is generated from the language model Θ_s. The affinity weight is measured by Kullback-Leibler divergence of s_i and s_j, contained in a decreasing logistic function $\mathcal{L}(x) = \frac{1}{1+e^x}$ to map the distance into interval [0,1] as proposed in [16,17]. That is, $f(s_i \rightarrow s_j) = \mathcal{L}(D_{KL}(s_j||s_i))$, where $D_{KL}(s_j||s_i)$ is:

$$D_{KL}(s_j||s_i) = \sum_{w \in W} p(w|\Theta_{s_j}) \log \frac{p(w|\Theta_{s_j})}{p(w|\Theta_{s_i})} \tag{1}$$

W is the set of words in our vocabulary and w denotes a word. The language model of Θ_s will be discussed in details later. If Θ_{s_i} and Θ_{s_j} are very close, the KL-divergence would be small and $f(s_i \rightarrow s_j)$ would be high, which intuitively makes sense.

Given $f(s_i \rightarrow s_j)$, the transition probability from s_i to s_j is then defined by normalizing the corresponding affinity weight as follows.

$$p(s_i \rightarrow s_j) = \begin{cases} \frac{f(s_i \rightarrow s_j)}{\sum_{k=1}^{|V|} f(s_i \rightarrow s_k)}, & if \sum f \neq 0 \\ 0, & otherwise \end{cases} \tag{2}$$

Note that $p(s_i \rightarrow s_j)$ is asymmetric and it measures the affinity from s_i to s_j. We let $f(s_i \rightarrow s_i) = 0$ to avoid self transition. We use the row-normalized matrix $M = [M_{ij}]_{|V|\times|V|}$ where $M_{ij} = p(s_i \rightarrow s_j)$ to describe G with each entry corresponding to the transition probability and all zero elements are replaced by a smoothing factor empirically set to $1/|V|$.

Based on the matrix *M*, the saliency score $\Psi(s_i)$ for sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as in the PageRank algorithm as follows:

$$\Psi(s_i) = \mu \cdot \sum_{all\ j \neq i} \Psi(s_j) \cdot M_{ji} + \frac{1-\mu}{|V|} \tag{3}$$

For implementation, the initial scores of all sentences are set to 1 and the iteration algorithm in Equation (3) is adopted to compute the new scores of the sentences. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any sentences falls below a given threshold (0.0001 in this study). μ is the damping factor usually set to 0.85, as in the PageRank algorithm. We then apply the Maximum Marginal Relevance (MMR) mechanism for redundancy removal, similar to the method used in [11].

We see that according to the KL-divergence scoring method, our main tasks are to estimate Θ_s. Since s can be regarded as a short document, we can use any standard method to estimate Θ_s. Here, we use Dirichlet prior smoothing [18] to estimate Θ_s as follows:

$$p(w|\Theta_s) = \frac{c(w,s) + \mu_s \cdot p(w|B)}{|s| + \mu_s} = \frac{c(w,s)}{|s| + \mu_s} + \frac{\mu_s}{|s| + \mu_s} \cdot p(w|B) \tag{4}$$

where $|s|$ is the length of s, $c(w,s)$ is the count of word w in s, $p(w|B)$ is a background model used as smoothing factor. Generally $p(w|B)$ is estimated by the whole document set D, i.e., using $\frac{c(w,D)}{\sum_{w' \in W} c(w',D)}$. μ_s is the smoothing parameter.

However, note that as the length of a sentence is very short, smoothing is critical for addressing the term sparseness problem for sentences. The globalized smoothing from the whole corpus is coarse-grained. Therefore, we move on to estimate the fine-grained $p(w|\Theta_s)$ from multiple-layers by the hierarchical graph summarization.

4 Hierarchical Graph Summarization

4.1 Overview

In the basic graph summarization, all sentences are indistinguishable, i.e., the sentences are treated uniformly. As we mentioned in Section 1, there may be many factors that can have impact on the importance analysis of the sentences. This study aims to examine the impact of hierarchical linkage on graph summarization, by incorporating sentence-to-document relationship, as well as visible and invisible document clustering.

Besides **1)** the basic pair-wise *sentence-to-sentence* relationship, the hierarchical graph includes **2)** *sentence-to-document* relationship, **3)** *document-to-cluster* relationship from *visible* linkage and **4)** *document-to-cluster* relationship from *invisible* linkage by semantic clustering. **We number these four types of linkage correspondingly**

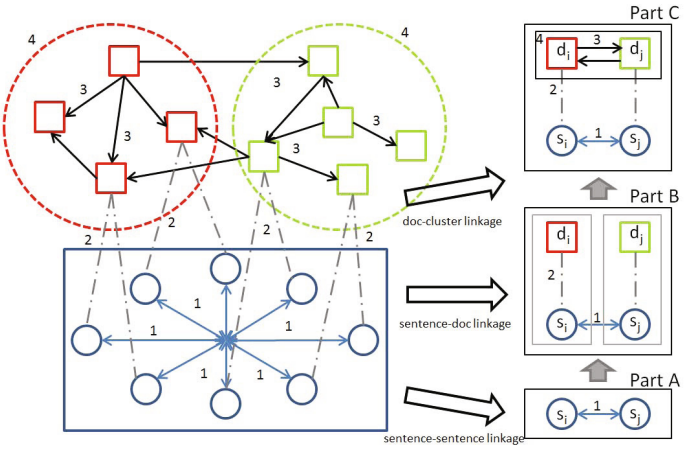

Fig. 1. Illustration of the hierarchical linkage graph. A circle denotes a sentence and a square denotes a document. Different lines denote different types of linkage, which are marked with Number **1-4**. Some lines are omitted due to the space limits.

in Fig. 1. As can be seen, the lowest layer is just the traditional link graph between sentences that has been well studied in previous work. The upper layer represents the documents. The dashed lines between these two layers indicate the conditional influence between the sentences and the documents: a link is established when the sentence is from the document. Documents are connected due to visible lines by anchor text arrows and are also grouped by invisible semantic clusters.

4.2 Incorporating Hierarchical Linkage

Sentence-to-Document Links. To incorporate the document-level information and the sentence-to-document relationship, the document-based graph model is proposed based on the two-layer link graph including both sentences and documents in Fig. 1.(Part B): the language model of sentence s is smoothed by the source document.

Visible Document-to-Cluster Links. Web documents are linked to each other through anchor texts and we keep such structural information. We start "walking" from a particular web document to all connected web documents until all linked documents are visited. These documents are clustered together as visible clusters, and the document within the visible cluster forms a visible document-to-cluster relationship.

Invisible Document-to-Cluster Links. Web documents can be clustered according to their semantic coherence, and the distance is calculated by the standard cosine similarity measurement. We use the popular clustering algorithms of K-means to produce the invisible semantic cluster. Given a document set, it is hard to predict the actual cluster number, and thus we empirically set the number k of expected clusters as k=$\sqrt{|D|}$, where $|D|$ is the number of documents.

Linkage Integration. After we introduce three types of hierarchical links, the estimation of the background language model Θ_B should be based on the source document

and source cluster where the sentence comes from, according to [8], the background model can be now written as:

$$p(w|B) = \frac{c(w,d) + \mu_c p(w|C)}{|d| + \mu_c} = \frac{c(w,d)}{|d| + \mu_c} + \frac{\mu_c}{|d| + \mu_c} \cdot p(w|C) \tag{5}$$

We take Equation (5) into Equation (4) and obtain the final representation:

$$\begin{aligned} p(w|\Theta_s) =& \frac{c(w,s)}{|s| + \mu_s} \cdot \frac{|s|}{|s|} + \frac{\mu_s}{|s| + \mu_s} \cdot \left(\frac{c(w,d)}{|d| + \mu_c} \cdot \frac{|d|}{|d|} + \frac{\mu_c}{|d| + \mu_c} \cdot p(w|C) \right) \\ =& \frac{|s|}{|s| + \mu_s} \cdot p(w|s) + \frac{\mu_s |d|}{(|s| + \mu_s)(|d| + \mu_c)} \cdot p(w|d) \\ &+ \frac{\mu_s \mu_c}{(|s| + \mu_s)(|d| + \mu_c)} \cdot p(w|C) \end{aligned} \tag{6}$$

μ_c can be interpreted as our confidence on the prior of how cluster information weighs. Thus setting μ_c=$|d|$ means that we put equal weights on the document-level and the cluster-level information. μ_c=0 yields no consideration of cluster-level information and μ_s=0 yields simple consideration of plain sentence relationships.

After simple calculation, we notice that the sum of all coefficients in Equation (6) equals to 1, and hence we change Equation (6) into a more concise format of

$$p(w|\Theta_s) = \alpha \cdot p(w|s) + \beta \cdot p(w|d) + \gamma \cdot p(w|C) \tag{7}$$

α, β, γ all belong to [0,1] and $\alpha + \beta + \gamma$=1. The cluster representation of $p(w|C)$ can be rewritten as a combination of visible cluster $p(w|C_v)$ and invisible cluster $p(w|C_{iv})$ controlled by λ:

$$p(w|C) = \lambda \cdot p(w|C_{iv}) + (1 - \lambda) \cdot p(w|C_v) \tag{8}$$

Special Cases:

(1) β=0 and γ=0: only plain relationship between two sentences are considered;
(2) $\beta \neq 0$, γ=0: plain linkage and document-to-sentence relationship included;
(3) $\gamma \neq 0$, λ=0 means no invisible clustering impact from visible linkage;
(4) $\gamma \neq 0$, λ=1 means no visible clustering impact from invisible linkage.

4.3 Estimation of Document/Cluster Importance

Documents and clusters are not equally important. Our assumption is that the sentences in an important document or cluster should be ranked higher and more likely to be chosen into the summary. The importance of documents (or clusters) is measured the relevance to the whole corpus. We examine such impact by incorporating the document importance and cluster importance into calculation of sentence linkage and ranking.

The function $\pi(d)$ aims to evaluate the importance of document d in the document set D. The following two methods are developed to evaluate the document importance.

π_{kl}: It uses the transformed KL-Divergence value between the document d and the whole document set D as the importance score of the document:

$$\pi_{kl}(d) = \mathcal{L}(D_{KL}(d||D)) \tag{9}$$

π_{pr}: It constructs a weighted graph between documents and uses the PageRank algorithm to compute the rank scores of the documents as the importance scores of the documents. The link structure among documents is established by the inherent visible linkage. The equation for iterative computation is the same with Equation (3).

The function $\phi(C)$ evaluates the importance of cluster C (both visible and invisible) in the document set D. Similarly we have two methods to evaluate cluster weights.

ϕ_{kl}: It uses the transformed KL-Divergence value between the cluster C and the whole document set D as the importance score of the cluster:

$$\phi_{kl}(C) = \mathcal{L}(D_{KL}(C||D)) \tag{10}$$

ϕ_{pr}: We add the PageRank scores of all the documents within the cluster C, i.e.,

$$\phi_{pr}(C) = \sum_{d \in C} \pi_{pr}(d) \tag{11}$$

By incorporating document and cluster importance, Equation (7) can be rewritten as Equation (12), substituting the unweighted $p(w|d)$ and $p(w|C)$. $p(w|\Theta_s)$ is estimated for all sentences and applied into Equation (1), (2), (3) to calculate the hierarchical sentence relationships and to rank sentences within the multiple-layer graph.

$$p(w|\Theta_s) = \alpha \cdot p(w|s) + \beta \cdot [\pi(d)p(w|d)] + \gamma \cdot [\phi(C)p(w|C)] \tag{12}$$

5 Experiments and Evaluation

5.1 Dataset

We use the data in [16] to test HGS on the real world datasets, which amounts to 5197 documents from various major news sites (such as BBC, CNN and Xinhua News, etc.). Our data includes 4 subjects, and each belongs to a different category of Rule of Interpretation (ROI) [4]. Reference summaries are created by editors [16].

Table 1. Detailed basic information of 4 datasets

Subjects	#Sentences	#Documents	#Visible Links	#RefSum (Avg. Length)
1.Influenza A	115026	2557	5108	5 (83)
2.BP Oil Spill	63021	1468	2493	6 (76)
3.Haiti Earthquake	12073	247	115	2 (32)
4.Michael Jackson Death	37819	925	1627	3 (64)

5.2 Evaluation Metrics

The ROUGE measure is widely used for evaluation [7]: the DUC contests usually officially employ ROUGE for automatic summarization evaluation. In ROUGE evaluation, the summarization quality is measured by counting the number of overlapping units, such as N-gram, word sequences, and word pairs between the candidate summaries CS

and the reference summaries RS. There are several kinds of ROUGE metrics, of which the most important one is ROUGE-N with 3 sub-metrics: precision, recall and F-score.

$$\text{ROUGE-N-R} = \frac{\sum_{\text{S}\in\text{RS}} \sum_{\text{N-gram}\in\text{S}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{S}\in\text{RS}} \sum_{\text{N-gram}\in\text{S}} \text{Count (N-gram)}}$$

$$\text{ROUGE-N-P} = \frac{\sum_{\text{S}\in\text{CS}} \sum_{\text{N-gram}\in\text{S}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{S}\in\text{CS}} \sum_{\text{N-gram}\in\text{S}} \text{Count (N-gram)}}$$

$$\text{ROUGE-N-F} = \frac{2 \times \text{ROUGE-N-P} \times \text{ROUGE-N-R}}{\text{ROUGE-N-P} + \text{ROUGE-N-R}}$$

S denotes a summary. N in these metrics stands for the length of N-gram and N-gram$\in$RS denotes the N-grams in reference summary while N-gram$\in$CS denotes the N-grams in the candidate summary. $\text{Count}_{\text{match}}$(N-gram) is the maximum number of N-gram in the candidate summary and in the set of reference summaries. $\text{Count}_{\text{(N-gram)}}$ is the number of N-grams in reference summaries or candidate summaries.

According to [7], among all sub-metrics, unigram-based ROUGE (ROUGE-1) has been shown to agree with human judgment most and bigram-based ROUGE (ROUGE-2) fits summarization well. We report three ROUGE F-measure scores: ROUGE-1, ROUGE-2, and ROUGE-W, where ROUGE-W is based on the weighted longest common subsequence. The weight W is set to be 1.2 in our experiments by ROUGE package (version 1.55). The higher the ROUGE scores, the similar the two summaries are.

5.3 Algorithms for Comparison

Pre-processing. Given a collection of documents, we first decompose them into sentences. Then the stop-words are removed and words stemming is performed. After these steps, we implement the following widely used summarization algorithms as baseline systems. They are designed for traditional summarization without hierarchical linkage. For fairness we conduct the same preprocessing for all algorithms.

Random: The method selects sentences randomly for each document collection.

Centroid: The method applies MEAD algorithm [3] to extract sentences according to the following parameters: centroid value, positional value, and first-sentence overlap.

GMDS: The plain graph MDS proposed by [11] first constructs a sentence connectivity graph based on cosine similarity and then selects important sentences based on the concept of eigenvector centrality.

PGMDS: Wan et al. present a two-layer pair-wise graph summarization methods in [12], utilizing sentence-to-sentence and sentence-to-document linkage without a consideration of simultaneous document-to-cluster links.

HGS: HGS is an algorithm with three-layer hierarchical linkage information and at the same time, both visible and invisible document clustering are performed.

RefSum: As we have used separate reference summaries from human evaluators, we not only provide ROUGE evaluations of the competing systems but also of the reference summaries against each other, which provides a good indicator of not only the upper bound ROUGE score that any system could achieve.

5.4 Overall Performance Comparison

We use a **cross validation** manner among 4 datasets, i.e., to train parameters on one subject set and to examine the performance on the others. After 4 training-testing processes, we take the average F-score performance in terms of ROUGE-1, ROUGE-2 and ROUGE-W on all sets. The details are listed in Tables 2∼5.

Table 2. Overall performance comparison on *Influenza A*. ROI* category: Science.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.491	0.112	0.159	0.44958
Random	0.197	0.039	0.081	0.75694
Centroid	0.241	0.050	0.094	0.45073
GMDS	0.252	0.059	0.098	0.33269
PGMDS	**0.303**	0.060	0.099	0.53123
HGS	0.298	**0.063**	**0.101**	0.53459

Table 3. Overall performance comparison on *BP Oil Leak*. ROI category: Accidents.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.517	0.135	0.183	0.48618
Random	0.202	0.041	0.096	0.64406
Centroid	0.259	0.052	0.098	0.34743
GMDS	0.267	0.057	0.102	0.43877
PGMDS	0.273	**0.061**	0.107	0.77245
HGS	**0.299**	0.058	**0.111**	0.39236

Table 4. Overall performance comparison on *Haiti Earthquake*. ROI category: Disasters.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.528	0.139	0.167	0.30450
Random	0.206	0.043	0.093	0.75694
Centroid	0.252	0.050	0.099	0.43045
GMDS	0.251	0.058	0.098	0.33694
PGMDS	0.275	0.055	0.106	0.64198
HGS	**0.307**	**0.060**	**0.115**	0.67312

Table 5. Overall performance comparison on *Jackson Death*. ROI category: Legal Cases.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.482	0.115	0.163	0.47052
Random	0.189	0.039	0.084	0.52426
Centroid	0.255	0.048	0.089	0.21045
GMDS	0.267	0.055	0.095	0.30070
PGMDS	0.281	**0.063**	0.107	0.67825
HGS	**0.294**	0.059	**0.113**	0.42148

*ROI: news categorization defined by Linguistic Data Consortium (http://www.ldc.upenn.edu/projects/tdt4/annotation).

From the results in Table 2 to Table 5, we have following observations:

• Generally Random has the worst performance.

• The results of Centroid are better than those of Random, mainly because the Centroid method takes into account positional value and first-sentence overlap, which facilitate main aspects summarization. However, the flat clustering-based summarization is proved to be less useful [15].

• The GMDS system outperforms centroid-based summarization methods. This is due to the fact that PageRank-based framework ranks the sentence using eigenvector centrality which implicitly accounts for information subsumption among all sentences.

• In general, the PGMDS algorithm outperforms GMDS system. It indicates that the two-layer hierarchical summarization is more useful than plain graph summarization and richer linkage structure indeed facilitates graph summarization.

• HGS under our proposed framework outperforms baselines, indicating that the overall properties we use for three layers of hierarchical linkage are beneficial for summarization tasks.

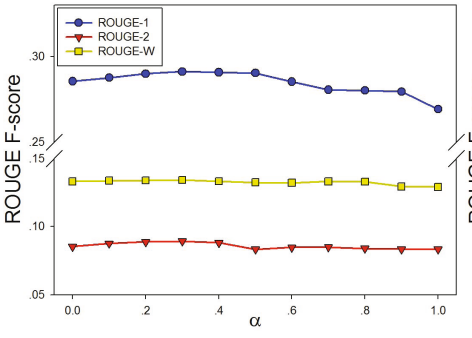

Fig. 2. α: weight of sentence-to-sentence links

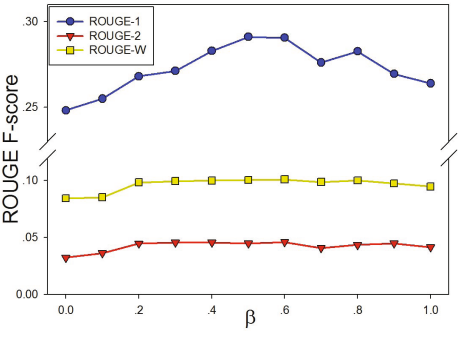

Fig. 3. β: weight of sentence-to-document links

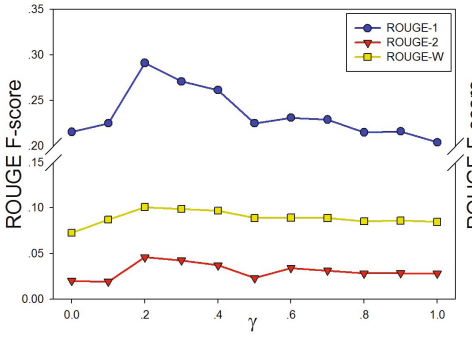

Fig. 4. γ: weight of document-to-cluster links

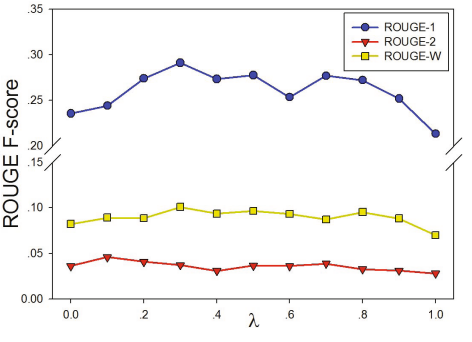

Fig. 5. λ: tradeoff of visible/invisible cluster

Having proved the effectiveness of our proposed methods, we carry the next move to identify how different layers of information take effects to enhance the quality of a summary in parameter tuning of α, β, γ and λ.

5.5 Parameter Tuning

Keeping other parameters fixed, we vary one parameter at a time to examine the changes of its performance from all 4 datasets. The first group of key parameters in our framework is α, β and γ where $\alpha + \beta + \gamma$=1. Every time we tune a parameter at a step of 0.1 and vary the other two for the best performance to achieve. Experimental results indicate the sentence-level relationship have stable but little impact on the summarization performance (illustrated in Fig. 2). The positive influence of documents and clusters are confirmed in Fig. 3 and Fig. 4 when $\beta \neq 0$ and $\gamma \neq 0$. Compared with document-level information, cluster-level information has a relatively weaker influence. Excessive use of higher level information impairs performance. Over smoothing from source texts might make the language models divergent from the original ones. We set α=0.3, β=0.5, γ=0.2 in our experiments.

Another key parameter in our framework is λ in Equation (8) to measure the tradeoff between visible and invisible cluster information. We gradually change λ from 0 to 1 at the step of 0.1 to examine the effect in Fig. 5. The combination of visible and invisible cluster outperforms the performance in isolation (λ=1 or 0). It is understandable that these two clustering metrics denote separate document organization methods and introduce different smoothing backgrounds. In general, a larger weight from visible cluster is preferable (λ=0.3).

Finally we examine the impact of document and cluster weights and the results are summarized in Table 6. From Table 6, we conclude that to distinguish the weight of documents and clusters is useful to measure sentence relationships because the usage of both weights brings prominent improve compared with $\pi(d)$=OFF and $\phi(C)$=OFF. We find that document weight by $\pi_{pr}(d)$ is much better than $\pi_{kl}(d)$, indicating that the web organization structure is helpful to find the centric documents within the corpus. The usage of $\pi_{kl}(d)$ has been proved in [12]. We also try different combinations of $\phi(C)$ for visible and invisible clusters. ϕ_{kl} means both clusters are weighed by KL-Divergence, and ϕ_{pr} means both clusters are weighed by PageRank score. ϕ_{kl+pr} means using KL-Divergence for visible clusters and using PageRank for invisible clusters, while ϕ_{pr+kl} means using PageRank score for visible clusters and using KL-Divergence for invisible clusters. We have an interesting finding that for visible clusters organized by anchor texts, the weights measured by PageRank seems to make more sense than using semantic coherence, and vice versa. Therefore, in general, the performance of ϕ_{pr+kl} is the most plausible weighting strategy.

Table 6. The impact of document weights and cluster weights, measured by KL-Divergence (kl), PageRank score (pr) and their different combinations

π \ ϕ	ON				OFF
	ϕ_{pr}	ϕ_{kl}	ϕ_{pr+kl}	ϕ_{kl+pr}	
π_{pr}	0.282	0.289	**0.291**	0.286	0.266
π_{kl}	0.268	0.271	0.273	0.265	0.254
OFF		0.242			0.237

6 Conclusions

In this paper we propose a Hierarchical Graph Summarization method, incorporating hybrid linkage information from multiple levels simultaneously into traditional graph summarization models. We utilize *sentence-to-sentence* relationship, *sentence-to-document* relationship and *document-to-cluster* relationship. We also investigate the web document structural information by explorations of **visible** and **invisible** document clusters, and the visible clusters earn heavier weights than invisible clusters (λ=0.3). Further more, we distinguish document/cluster by measuring their corresponding weights, calculating KL-Divergence and PageRank scores.

Abundant experiments are conducted on 4 real datasets, comparing 5 rival algorithms. Experimental results demonstrate the effectiveness of our proposed HGS. The benefits of visible and invisible clustering are also confirmed. Documents and clusters

should be distinguished by their significance. We also find that the semantic coherence for invisible clustering has not shown as promising effects as visible clustering does.

Acknowledgments. This work was partially supported by HGJ 2010 Grant 2011ZX01042-001-001 and NSFC with Grant No. 61073081. Xiaojun Wan was supported by NSFC with Grant No.61170166, and Rui Yan was supported by the MediaTek Fellowship.

References

1. Allan, J., Gupta, R., Khandelwal, V.: Temporal summaries of new topics. In: Proceedings of the 24th Annual International ACM SIGIR Conference, pp. 10–18 (2001)
2. Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP 2004, pp. 1–7 (2004)
3. Fukumoto, F., Suzuki, Y.: Extracting key paragraph based on topic and event detection: towards multi-document summarization. In: NAACL-ANLP 2000, pp. 31–39 (2000)
4. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference, pp. 297–304 (2004)
5. Li, L., Zhou, K., Xue, G.-R., Zha, H., Yu, Y.: Enhancing diversity, coverage and balance for summarization through structure learning. In: WWW 2009, pp. 71–80 (2009)
6. Lin, C.-Y., Hovy, E.: From single to multi-document summarization: a prototype system and its evaluation. In: Proceedings of ACL 2002, pp. 457–464 (2002)
7. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using N-gram co-occurrence statistics. In: Proceedings of NAACL-HLT 2003, pp. 71–78 (2003)
8. Mei, Q., Zhai, C.: Generating Impact-Based Summaries for Scientific Literature. In: Proceedings of ACL 2008, pp. 816–824 (2008)
9. Mihalcea, R., Tarau, P.: A language independent algorithm for single and multiple document summarization. In: Proceedings of IJCNLP 2005, pp. 19–24 (2005)
10. Shen, C., Wang, D., Li, T.: Topic aspect analysis for multi-document summarization. In: Proceedings of CIKM 2010, pp. 1545–1548 (2010)
11. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of AAAI 2008, pp. 855–860 (2008)
12. Wan, X.: An Exploration of document impact on graph-based multi-document summarization. In: Proceedings of EMNLP 2008, pp. 755–762 (2008)
13. Wan, X., Xiao, J.: Graph-based multi-modality learning for topic-focused multi-document summarization. In: Proceedings of IJCAI 2009, pp. 1586–1591 (2009)
14. Wang, D., Zhu, S., Li, T., Gong, Y.: Multi-document summarization using sentence-based topic models. In: Proceedings of ACL/AFNLP 2009 (Short Papers), pp. 297–300 (2009)
15. Wang, D., Li, T.: Document update summarization using incremental hierarchical clustering. In: Proceedings of CIKM 2010, pp. 279–288 (2010)
16. Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., Zhang, Y.: Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In: Proceedings of the 34th Annual International ACM SIGIR Conference, pp. 745–754 (2011)
17. Yan, R., Nie, J.-Y., Li, X.: Summarize what you are interested in: an optimization framework for interactive personalized summarization. In: EMNLP 2011, pp. 1342–1351 (2011)
18. Zhai, C., Lafferty, J.D.: A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In: Proceedings of SIGIR 2001, pp. 334–342 (2001)