Query-Focused Multi-document Summarization Based on Query-Sensitive Feature Space

Wenpeng Yin, Yulong Pei, Fan Zhang, Lian'en Huang
Shenzhen Key Lab for Cloud Computing Technology and Application
Peking University Shenzhen Graduate School, Shenzhen 518055, China
{mr.yinwenpeng, paul.yulong.pei, fan.zhgf}@gmail.com, hle@net.pku.edu.cn

ABSTRACT

Query-oriented relevance, information richness and novelty are important requirements in query-focused summarization, which, to a considerable extent, determine the summary quality. Previous work either rarely took into account all above demands simultaneously or dealt with part of them in the dynamic process of choosing sentences to generate a summary. In this paper, we propose a novel approach that integrates all these requirements skillfully by treating them as sentence features, making that the finally generated summary could fully reflect the combinational effect of these properties. Experimental results on the DUC2005 and DUC2006 datasets demonstrate the effectiveness of our approach.

Categories and Subject Descriptors

H.3.1 [**Information Store and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Algorithms, Experimentation, Performance

Keywords

query-focused summarization, query-biased sentence feature, topical vector space model, LDA

1. INTRODUCTION

For query-focused multi-document summarization, a summarizer incorporates user-declared queries and generates summaries that not only deliver the majority of information content from a set of documents but also bias to the queries. A good query-biased summary is generally supposed to meet the following typical demands: query-biased relevance, biased information richness and biased novelty. Query-biased relevance requires that the sentences in the summary must overlap with the query in terms of topical content. Query-biased information richness denotes the information degree of the sentences with respect to both the sentence collection and the query. Query-biased information novelty is used to measure the content uniqueness of a sentence based on its capabilities in differentiating itself from other sentences as well as responding to the demands of the query.

Most existing work treats query-focused multi-document summarization as a sentence ranking problem at first, then exploit some strategies to deal with redundancy removal, coverage and balance during the sentence selection, which is a dynamic and greedy procedure, such as [4, 7]. In this paper, we treat some governing requirements as features of sentences. Hence we extract representative query-oriented sentence features entirely to form a feature space at the initial stage, accordingly, Gaussian Mixture Model (GMM) is utilized to cluster sentences over the feature space so as to reduce the size of the target cluster (i.e., the cluster containing the most query-biased salient sentences). This also makes the subsequent ranking of sentences more robust because it is less sensitive to outliers such as noise sentences. The rationale behind most existing clustering-based summarization methods is that different clusters represent different aspects of the documents, which should all be covered if possible [4]. Differently, we only consider one most eligible cluster as our target cluster in this study because the feature space we have defined in initial phase makes the sentences in the target cluster be the most suitable choices to construct a summary. The only thing left to do is to sort the sentences within the cluster. Extensive experiments on the DUC2005-2006 benchmark data sets demonstrate that our method outperforms some representative baseline approaches in all specified evaluation measures.

2. RELATED WORK

We briefly introduce some related summarization methods, especially those explicitly taking various requirements into consideration.

Query-oriented sentence relevance in [6] was acquired by opting for the trade-off of the sentence's initial relevance to the query and its similarities to other sentences in the document cluster. Li et al. [3] treated summarization as a supervised sentence ranking process, where coverage, balance and novelty properties were incorporated. Whereas, it focused on generic summarization rather than query-biased situation. Wan et al. [7] gave explicit definitions of biased information richness and novelty, then, they proposed to

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

compute biased information richness using manifold-ranking process and a greedy algorithm similar to [8] was applied to keep low information redundancy in the summaries. The method proposed in [4] is closely related to our work for it considered novelty, coverage and balance wholly. However, a common characteristic of existing methods in acquiring novelty or balance property lies in the assumption that we have already selected the first k-1 sentences $s_1, \cdots, s_{k-1}$ for a summary, which makes the novelty or balance acquiring must be conducted during the dynamic and greedy process of choosing sentences to generate summary.

3. TOPICAL VECTOR SPACE MODEL (TVSM)

In order to relieve the inconvenience cast by the inherent information shortage of sentences on the calculation of sentence relatedness, we come up with a new model named TVSM through exploiting Latent Dirichlet Allocation (LDA) [1] to modify Vector Space Model (VSM), which has been commonly utilized to represent texts.

In our model, given a sentence a=(w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_4), using the posterior probabilities, we annotate each word in a with a topic assignment. Suppose this yields an ordered sequence of word-topic pairs: (w_1, z_1), (w_2, z_2), (w_3, z_3), (w_4, z_4), (w_5, z_1), (w_6, z_5), (w_7, z_2), (w_8, z_2), (w_9, z_2), (w_4, z_4). In this sequence, nine discriminative terms are mapped to five topics. Accordingly, we could represent a over topic space and the topic weights $f_a^{z_i}$ ($1 \le i \le K$, K denotes the total topic number in LDA) are derived as follows:

$$f_a^{z_1} = P(z_1|w_1) + P(z_1|w_5)$$
$$f_a^{z_2} = P(z_2|w_2) + P(z_2|w_7) + P(z_2|w_8) + P(z_2|w_9)$$
$$f_a^{z_3} = P(z_3|w_3);\ \ f_a^{z_4} = P(z_4|w_4) \times 2$$
$$f_a^{z_5} = P(z_5|w_6);\ \ f_a^{z_i} = \epsilon \quad (6 \le i \le K)$$

In general, we collect those words assigned topic z_i in sentence s as a set s_{z_i} ($1 \le i \le K$) and use symbol $C(w, s)$ to denote the number of occurrences of word w in s. The formula for calculating the topic impacts on s is defined as follows:

$$f_s^{z_i} = \begin{cases} \sum\limits_{w \in s_{z_i}} P(z_i|w) \cdot C(w,s) & \text{if } s_{z_i} \neq \emptyset \\ \epsilon & \text{otherwise} \end{cases} \quad (1)$$

where

$$P(z_i|w) = \frac{P(w|z_i) \cdot P(z_i)}{\sum_{j=1}^{K} P(w|z_j) \cdot P(z_j)} \quad (2)$$

Constant ϵ in Equation 1 is set to e^{-4} empirically. We give those topics not used to annotate s a small impact so as to smooth the vectors. Provided with topic vector representations of two sentences, we could make use of cosine measure or other strategies to determine their similarity. By using topic space, which is much smaller and denser than the original term vector space, we gain a reduced representation where similarity between short text can be more reliably estimated. Furthermore, the latent topic space reduces the impact of *noise* terms. In experiments, we use the GibbsLDA++[1], a C/C++ implementation of LDA using Gibbs Sampling.

[1]GibbsLDA++: http://gibbslda.sourceforge.net

4. FEATURE DESIGN

Query-oriented summarization not only requires the features to deliver the salient cluster content, demonstrate considerable consistency with the query motivation, it also prefers a low information redundancy in the generated summary.

4.1 Four Kinds of Relevance

Here, we make use of different types of information available in the DUC2005 and DUC2006 multi-document summarization datasets:

- t_s: Title of the document containing sentence s.
- d_s: The document containing sentence s.
- c_s: The document cluster containing sentence s.
- q_s: Query (attached with the topic statement) of the document collection containing sentence s.

Each of these elements is represented as a vector over latent topic space presented in Section 3. The vector representation of a document could be extracted easily from the output files of GibbsLDA++ and it is actually a multinomial distribution. The topic vector of a document cluster is obtained by averaging the distribution vectors of all the documents in the cluster. We do not conduct sentence segmentation if the query consists of more than one question, instead we treat it as a single, long sentence. Given the resulting representation of above elements as vectors, we calculate the following four sentence features:

- $r_{t,s}$: Relevance between sentence s and the title of the document to which s belongs.
- $r_{d,s}$: Relevance between sentence s and the document to which s belongs.
- $r_{c,s}$: Relevance between sentence s and the document cluster to which s belongs.
- $r_{q,s}$: Relevance between sentence s and the query.

Given two sentences b=(b_{z_1}, b_{z_2}, b_{z_3}, $\cdots$, b_{z_i}, $\cdots$, b_{z_K}) and c=(c_{z_1}, c_{z_2}, c_{z_3}, $\cdots$, c_{z_i}, $\cdots$, c_{z_K}), b_{z_i} and c_{z_i} denote the weight of topic z_i in b and c, respectively. we choose to compute their similarity on the basis of Jensen-Shannon (JS) divergence. It is a symmetrized and smoothed version of the Kullback-Leibler (KL) divergence, calculated as the KL divergence of b, c with respect to the average of the two input distributions. The JS divergence is then defined as:

$$D_{JS}(b,c) = \frac{1}{2}[KL(b,m) + KL(c,m)] \quad (3)$$

where $m = \frac{b+c}{2}$. To use the JS divergence as a relevance measure, we scale it to [0, 1] and invert by subtracting from 1, hence:

$$Rel_{JS}(b,c) = 1 - D_{JS}(b,c) \quad (4)$$

4.2 Biased Information Richness (BIR)

Given a sentence collection and a query q, the BIR of sentence s is used to indicate the information degree of the sentence s with regard to both the sentence set and q, i.e., the richness of information contained in the sentence s biased towards q.

This feature score for each sentence is obtained via a variant version of the manifold-ranking process proposed in [7]. Points $\{s_0, s_1, \cdots, s_n\}$ denote the query statement (s_0) and all the sentences in the document collection ($\{s_i|1 \leq i \leq n\}$) in a manifold space. The ranking function is denoted by $f = [f_0, f_1, ..., f_n]$. Wan et al. [7] hypothesized that all the sentences had blank prior knowledge so their initial scores were all set to zero. Whereas in this study, it is rational to treat the relevance between a sentence and the query discussed in above section as prior knowledge of the sentence. Since s_0 denotes the query description, the initial score vector of these sentences is y =$[y_0, y_1, ..., y_n]$, where $y_0 = 1$ and y_i = Rel(s_i, s_0) (1≤ i≤ n). The manifold ranking can be performed iteratively using the following equation:

$$f(k+1) = \alpha S f(k) + (1-\alpha)y \tag{5}$$

where S is the symmetrically normalized similarity/relevance matrix as for $\{s_0, s_1, \cdots, s_n\}$, trade-off parameter α is set to 0.6, and k indicates the k^{th} iteration. Obviously, modified initial scores will exert a greater influence to sentence scores than the settings in [7] at each step of the iteration process. After convergence, let $f_i^\star$ denotes the limit of the sequence $\{f_i(t)\}$, then the BIR of sentence s_i is:

$$BIR(s_i) = f_i^\star \quad (1 \leq i \leq n) \tag{6}$$

4.3 Biased Information Novelty (BIN)

A desired summary should have low information redundancy. In order to devise a feature to differentiate those sentences that have a degree of similarity, we perform the modified MMR algorithm in [7] on the basis of feature BIR discussed Section 4.2. Lots of work (e.g., [7, 2]) used some strategies to impose penalty to the remaining sentences once a new sentence has been added to the summary. However, that treatment easily neglects the underlying positive effects on summary quality brought probably by other sentence features. In our approach, we just treat the MMR result as a sentence feature named *biased information novelty*. Hence, a proportion of existing methods are approximately equivalent to our current section. Our algorithm for BIN acquiring goes as follows:

1. Initiate two sets A=∅, B=$\{s_i|$i=1,2, $\cdots$, n$\}$, and each sentence's initial novelty is set to its biased information richness obtained in Equation 9, i.e., BIN(s_i)= BIR(s_i).

2. Sort the sentences in B by their current BIN scores in descending order.

3. Suppose s_i is the highest ranked sentence, i.e., the first sentence in the ranked list. Move sentence s_i from B to A and update the BINs of the remaining sentence(s) in B as follows:

   For each sentence s_j in B:

   $$BIN(s_j) = BIN(s_j) - \omega \cdot S_{ij} \cdot BIN(s_i) \tag{7}$$

   where $\omega > 0$ is a parameter factor, and S is the same normalized similarity matrix in Equation 8.

4. Go to step 2 and iterate until $|B|$=0.

5. CLUSTERING AND RANKING

Six representative sentence features have been extracted in Section 4. We use GMM to cluster sentences into four clusters over their feature space. The main motivation for the clustering is to reduce the size of the target cluster and make the subsequent sentence ranking more robust. It should be noted that the cluster number is flexible. Whereas, too many clusters probably results that the sentences in the target cluster are too few to meet the summary length. Therefore, we set it to 4 empirically. The following subsection describes GMM in general and how we used it in our setting.

5.1 Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) is a simple linear superposition of Gaussian components for the purpose of providing a richer class of density models than the single Gaussian. Clustering based on GMM is probabilistic in nature and aims at maximizing the likelihood function with regard to the parameters (comprising the means and covariances of the components and the mixing coefficients). Consider n data points $S = \{s_1, s_2, \cdots, s_n\}$ in d-dimensional space, the probability density of s_i can be defined as follows:

$$p(s_i|\pi, \mu, \Sigma) = \sum_{z=1}^{k} \pi_z \cdot N(s_i; \mu_z, \Sigma_z) \tag{8}$$

where k is the component number, π_z is the prior probability of the z^{th} Gaussian component. $N(s_i; \mu_z, \Sigma_z)$ is defined as:

$$N(s_i; \mu_z, \Sigma_z) = \frac{exp\{-\frac{1}{2}(s_i - \mu_z)^T \sum_z^{-1}(s_i - \mu_z)\}}{((2\pi)^d |\Sigma_z|)^{\frac{1}{2}}} \tag{9}$$

Under the assumption that the data points are independent and identically distributed (i.i.d), the log of the likelihood function is given by:

$$\begin{aligned} lnP(S|\pi, \mu, \Sigma) &= ln \prod_{i=1}^{n} P(s_i|\pi, \mu, \Sigma) \\ &= ln \prod_{i=1}^{n} \sum_{z=1}^{k} \pi_z \cdot N(s_i; \mu_z, \Sigma_z) \\ &= \sum_{i=1}^{n} ln\{\sum_{z=1}^{k} \pi_z \cdot N(s_i; \mu_z, \Sigma_z)\} \end{aligned} \tag{10}$$

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is Expectation Maximization algorithm (EM). EM is an iterative algorithm in which each iteration contains an E-step and a M-step. In the E-step, we compute the probability of the z^{th} Gaussian component given the data point s_i using the current parameter values:

$$p(z|s_i, \pi, \mu, \Sigma) = \frac{\pi_z \cdot N(s_i; \mu_z, \Sigma_z)}{\sum_{j=1}^{k} \pi_j \cdot N(s_i; \mu_j, \Sigma_j)} \tag{11}$$

In the M-step, we re-estimate the parameters using the current responsibilities, as follows:

$$\mu_z^{new} = \frac{1}{n_z} \sum_{i=1}^{n} s_i \cdot p(z|s_i, \pi, \mu, \Sigma) \tag{12}$$

$$\Sigma_z^{new} = \frac{1}{n_z}\sum_{i=1}^{n} p(z|s_i,\pi,\mu,\Sigma)(s_i-\mu_z^{new})(s_i-\mu_z^{new})^T \tag{13}$$

$$\pi_z^{new} = \frac{n_z}{n} \tag{14}$$

where

$$n_z = \sum_{i=1}^{n} p(z|s_i,\pi,\mu,\Sigma) \tag{15}$$

The EM algorithm runs iteratively until the log likelihood reaches (approximate) convergence. And we use K-means to determine the initial model parameters: μ, Σ and prior probabilities of the components π_z $(1 \le z \le k)$. In general, GMM performs better than classical hard clustering algorithms such as K-means as it is less sensitive to outliers. In experiments, we simply run GMM until the increment of log likelihood is less than 0.000001 and pick the one with largest log likelihood as the best estimate of the underlying clusters.

The above GMM algorithm gives probabilistic assignments of sentences belonging to a given cluster $p(z|s,\pi,\mu,\Sigma)$. For each cluster, we pick all the sentences with this probability to be greater than 0.9. This is done as we want to determine the true representative sentences per cluster. Using these sentences, we compute the new average features per cluster and pick the cluster with the larger features (or best of all) as our target cluster. In order to sort sentences within the target cluster, we propose the following ranking algorithm.

5.2 Gaussian Ranking Algorithm

We assume features to be Gaussian distributed (which is true in most case, though with a bit of skew in some cases). For any given feature s_i^f, we compute the μ_f and σ_f based on the sentences in the target cluster. The Gauss rank R_G of a given sentence s_i is then defined as follows:

$$R_G(s_i) = \sum_{f=1}^{d} \omega_f \int_{-\infty}^{s_i^f} N(x;\mu_f,\sigma_f)dx \tag{16}$$

where $N(x;\mu_f,\sigma_f)$ is the univariate Gaussian distribution with model parameters as μ_f and σ_f, and ω_f is the weight of feature s_i^f. The inner integral in this equation computes the Gaussian cumulative distribution at s_i^f. Gaussian CDF (cumulative distribution function) is a monotonically increasing function well suited to our ranking problem as we prefer a higher value over low value for each feature. Alternately, if a low value is preferred for some features, then s_i^f could be replaced by $-s_i^f$ in the above formula.

This algorithm helps in devising a total ordering under $\le$ over all the sentences in the target cluster. Accordingly, we observe that weight normalization doesn't change the ordering of sentences. Hence, the only constraint we put on these feature weights is $\{\forall f : 0 \le \omega_f \le 1\}$.

6. EXPERIMENTAL STUDY

6.1 Data Sets and Evaluation Metrics

We evaluate our proposed approach for query-focused multi-document summarization on the main tasks of DUC2005[2] and DUC2006[3]. Each task has a gold standard dataset consisting of document sets and reference summaries. Documents are pre-processed by segmenting sentences and splitting words. Stop words are removed and the remaining words are stemmed using Porter stemmer[4]. Table 1 gives a short summary of the two datasets.

Table 1: Summary of datasets

	DUC2005	DUC2006
Number of clusters	50	50
Documents per cluster	25-50	25
Summary length	250 words	250 words

In experiments, we use the ROUGE [5] (version 1.5.5) toolkit[5] for evaluation, which is officially adopted by DUC for evaluating automatic generated summaries. It measures summary quality by counting the overlapping units between system-generated summaries and human-written reference summaries. We report three common ROUGE scores in this paper, namely ROUGE-1, ROUGE-2 and ROUGE-SU4 which base on Uni-gram match, Bi-gram match, and unigram plus skip-bigram match with maximum skip distance of 4, respectively.

6.2 Experimental Results

6.2.1 System Comparison

We compare our method G_TVSM_JS with the following algorithms. (1)Rel: a method considering only the sentence relevance towards the query and choosing the most relevant sentences to produce summary until length limit is reached. (2)Rel_MMR: similar to (1) except that we utilize MMR algorithm to reduce redundancy. (3)Coverage: a baseline clustering-based method. It clusters sentences and selects the most relevant sentences from different clusters. (4)Manifold: ranking the sentences according to the manifold ranking scores. (5)Random: a baseline producing summary by random sentence selection. (6)top three systems with the highest ROUGE scores that participated in the DUC2005 (S4, S15, S17) and the DUC2006 (S12, S23, S24) for comparison, respectively. It should be noted that JS divergence based on TVSM is applied to some baseline systems above if they need measure sentence similarity so as to enhance the comparability and persuasiveness of our experiments.

Tables 2 and 3 show the experimental results. From those statistics, following conclusions can be drawn: (1)Method *Rel* has relatively poor performance. It is merely better than *Random* on all three evaluation metrics, which suggests that considering only query-biased relevance is not enough. Similarly, only taking *coverage* property into account does not benefit the summary quality either. (2)Rel_MMR outperforms *Rel*. Clearly, combining novelty with biased relevance indeed boosts the summary quality. (3)Rel_MMR is inferior to $Manifold$. Recall that Wan et al. [7] integrated manifold-ranking process and MMR algorithm to implement their system. Hence biased information richness is a more effective indicator than query-oriented relevance in respect of reflecting the potential value of sentences. (4)G_TVSM_JS outperforms the comparative systems on all evaluation metrics. It confirms our judgment that various query-aware sen-

[2]http://www-nlpir.nist.gov/projects/duc/duc2005/tasks.html
[3]http://www-nlpir.nist.gov/projects/duc/duc2006/tasks.html
[4]http://tartarus.org/ąńmartin/PorterStemmer/
[5]http://www.isi.edu/licensed-sw/see/rouge/

Table 2: Comparison results on DUC2005

Systems	ROUGE-1	ROUGE-2	ROUGE-SU4
Random	0.30821	0.03976	0.10625
Coverage	0.37324	0.07113	0.12805
Rel	0.36775	0.07092	0.12777
Rel_MMR	0.37416	0.07160	0.12856
Manifold	0.37497	0.07423	0.12907
S17	0.36933	0.07286	0.12937
S4	0.37584	0.07063	0.12868
S15	0.37656	0.07383	0.13248
G_TVSM_JS	**0.39008**	**0.07991**	**0.13527**

Table 3: Comparison results on DUC2006

Systems	ROUGE-1	ROUGE-2	ROUGE-SU4
Random	0.34821	0.05297	0.11908
Coverage	0.36814	0.07113	0.12833
Rel	0.36775	0.07092	0.12777
Rel_MMR	0.37416	0.07160	0.12856
Manifold	0.38867	0.08308	0.13307
S23	0.40973	0.09785	0.16162
S12	0.41053	0.09633	0.16074
S24	0.41081	0.09957	0.15248
G_TVSM_JS	**0.42367**	**0.10105**	**0.16425**

tence features could help to explore potential sentences from multiple aspects to meet the user's information needs.

6.2.2 *Influence of Topic Number K*

LDA is a crucial tool in devising TVSM. In order to investigate how the topic number K exerts influence to system performance, K is varied from 20 to 170.

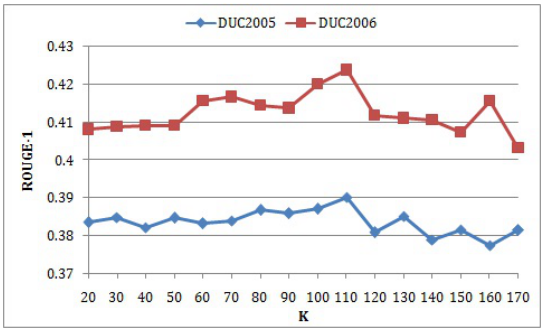

Figure 1: ROUGE-1 vs. K

Figure 1 shows the influence of K over ROUGE-1 on two data sets. On the whole, the evaluation scores rise slowly with the increase of K, and reach their respective maximum around $K = 110$, then gradually go down when K continues to become larger. Despite some fluctuations, they don't matter to the overall conclusion. The observed performance variations can be explained by the way latent topic models map words to topics: Given a fixed number of latent topics to *fill*, the algorithm may split up a single topic into two distinct ones, or vice versa merge different topics, which in turn affects similarity scores and the feature extraction. Another point worth mentioning is that even with $K = 110$ latent topics, the dimensionality of TVSM is much lower than that of VSM.

6.2.3 *Estimating Optimal Weights*

In order to estimate the weight parameters in Equation 16, for each weight vector ω, we run our summarization algorithm and measure the ROUGE-1 value. This result is henceforth called the score of the weight vector ω. To identify the weights that maximize the score, we use stochastic hill climbing along with simulated annealing in the unit hypercube that encloses all possible weight vectors (recall that weights are bounded by [0, 1]). We skip the details of the algorithm and just report an approximate global optimal weight vector adopted in our experiments: ($r_{t,s}$:0.05, $r_{d,s}$:0.15, $r_{c,s}$:0.1, $r_{q,s}$:0.2, BIR:0.15, BIN:0.35), which is a *feature:weight* sequence. It suggests that the most dominant features in our algorithm are the biased information novelty *BIN* and sentence-query relevance $r_{q,s}$. On the other hand, the sentence-title relevance $r_{t,s}$ does not have a significant influence on quality of the resulting summaries.

7. ACKNOWLEDGMENTS

This research is financially supported by NSFC of China (Grant No. 60933004 and 61103027) and HGJ (Grant No. 2011ZX01042-001-001).

8. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] X. Cai and W. Li. A context-sensitive manifold ranking approach to query-focused multi-document summarization. *PRICAI 2010: Trends in Artificial Intelligence*, pages 27–38, 2010.

[3] L. Li, K. Zhou, G. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international conference on World wide web*, pages 71–80. ACM, 2009.

[4] X. Li, Y. Shen, L. Du, and C. Xiong. Exploiting novelty, coverage and balance for topic-focused multi-document summarization. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1765–1768. ACM, 2010.

[5] C. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the workshop on text summarization branches out (WAS 2004)*, pages 25–26, 2004.

[6] J. Otterbacher, G. Erkan, and D. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 915–922. Association for Computational Linguistics, 2005.

[7] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of IJCAI*, volume 7, pages 2903–2908, 2007.

[8] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W. Ma. Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511. ACM, 2005.