

# Comments-Oriented Document Summarization Based on Multi-aspect Co-feedback Ranking

Lifu Huang, Hongjie Li, and Lian'en Huang\*

Shenzhen Key Lab for Cloud Computing Technology and Applications  
Peking University Shenzhen Graduate School  
Shenzhen, Guangdong 518055, P.R. China  
{warrior.fu,lihongjie99}@gmail.com, hle@net.pku.edu.cn

**Abstract.** With the popularity of Web 2.0, comments left by readers on web documents have drawn much attention. In this paper, we study the problem of comments-oriented document summarization, which aims to summarize a web document by considering not only its content but also the comments. Generally, most of the comments usually convey one or a few aspects of the document. Given a sentence set from both the web document and its corresponding comments to summarize, we can divide different sentences into different clusters (named “aspects”) according to the content. It is challenging and interesting to summarize the web document based on these clusters. Motivated by this, we propose a novel model: MultiAspectCoRank, for comments-oriented document summarization. Firstly we rank all the sentences based on the multiple aspects obtained from the whole document, and then provide each ranking list as feedback to others until the top-N results of each ranking list are unchanged. We get the final result by integrating these different ranking lists together. Experimental results on a set of real-world blog data with manually labeled sentences show the promising performance of our approach.

**Keywords:** Comments-Oriented Summarization, Co-Feedback Ranking, Affinity Propagation.

## 1 Introduction

Automatic Document summarization (ADS) has been widely studied in past decades. It aims to generate a compressed summary by extracting the major information from a document. Existing methods mostly focus on the document’s content information while ignoring readers’ opinions. With the development of web 2.0, readers are inspired to express and share their views by commenting. This kind of information coming from readers is much valuable for some special IR (Information Retrieval) tasks. On the one hand, these comments are generated based on readers’ understanding of the document, which to a certain extent reflect some aspects of the document. On the other hand, comments, which have

---

\* Corresponding author.

consistency with document text on the content, can be viewed as an additional part of the document. In this paper, we focus on comments-oriented document summarization (CODS), aiming to generate summaries for a single web document based on not only its content but also the comments generated by its readers.

The research of comments-oriented document summarization (CODS) is popular in recent years. Its task is to summarize a document by extracting representative sentences from the document based on the information hidden in its comments [3]. There are two common challenges for comments-oriented document summarization: the first is that a comment set might contain diverse information, which is either related or unrelated to the document content, so it is necessary to design effective methods to extract useful information from comments to help summarize document content; the second is that based on the information extracted from comments, how to rank the document sentences to get effective summaries. After all, sentence ranking is always the issue of most concern in summarization tasks.

Some existing CODS methods firstly derived the main information from comments with different strategies, such as, by identifying different relations between the comments to choose the most popular ones [3,4]. After this, based on the selected comments, chose top-N document sentences with different ranking strategies and generated the summaries. However, these methods ignored a key point that comments are supplement of document content but cannot always cover all aspects. Indeed, a large proportion of the comments are not directed to the content of the document or just act on a few aspects of the document. The selected comments may be related or unrelated to the main aspect of the document. So it is not always effective to generate the summaries based on these comments. Moreover, these works didn't consider that different comments might have different effects on the final ranking results.

Inspired by these, we propose a novel method called MultiAspectCoRank for CODS in this study. Firstly, in order to obtain useful information from comments and weaken the negative effect of the unrelated or one-sided ones, we try to fuse the document sentences and its comments together to get multiple aspects. Each aspect can be regarded as a part of the document. With these aspects, we can get multiple ranking lists, indicating different kinds of candidate summaries based on different angles. To overall consider the influence of different aspects on the ranking results, we use the co-feedback ranking framework, selecting top-N sentences from each ranking list and providing them as feedback to update other aspect based ranking lists in order to boost the ranking performance. This process continues iteratively until the top-N of each ranking list are unchanged. We get the final result by integrating these different ranking lists according to the weights of different aspects. Compared with previous work, our proposed MultiAspectCoRank method performs better on a real-world blog dataset with manually labeled sentences.

The rest of this paper is organized as follows. We briefly review the related work in Section 2. In Section 3 we present the details of multi-aspect co-feedback

ranking method for CODS. Experiments and results are discussed in Section 4. Finally we draw a conclusion of this study in Section 5.

## 2 Related Work

Comments-oriented document summarization is developed from automatic document summarization, so the related work will be introduced in two parts. Firstly we describe some representative summarization methods and then the special work about comments-oriented document summarization is presented briefly.

### 2.1 Automatic Document Summarization (ADS)

Traditional document summarization is a process to generate a summary by reducing documents in size while retaining the main characteristics of the original documents. In order to achieve this goal, different features and ranking strategies have been studied.

Feature-based sentence-ranking approaches are widely used in automatic document summarization. Radev et al. [10] proposed a centroid-based method, which implemented MEAD as a centroid-based summarizer by combining several predefined features including TF\*IDF, cluster centroid and position to score the sentences. Lin and Hovy [5] built the NeATS multi-document summarization system using term frequency, sentence position and stigma words. Nenkova et al. [8] proved that high-frequency words were significant in reflecting the focus of documents. Ouyang et al. studied the influence of different word positions in summarization [9].

Graph-based ranking algorithms nowadays are successfully applied in summarization. LexPageRank [1] is the representative work which is based on the PageRank algorithm. Graph-based ranking algorithms take global information into consideration rather than rely only on vertex-specific information, therefore have been proved successful in document summarization. Some methods have been proposed to extend the conventional graph-based models recently including multi-layer graph incorporated with different relationship [12], multi-modality graph based on the manifold-ranking method [13] and DivRank [7] introducing the time-variant matrix into a reinforced random walk to balance prestige and diversity.

### 2.2 Comments-Oriented Document Summarization (CODS)

Comments-oriented document summarization (CODS) is a special task developed from traditional document summarization. The task is to summarize a blog post using the information hidden in its comments. Yang et al. [15] explored an approach by modeling web documents and social contexts into a unified framework. They proposed a dual wing factor graph (DWFG) model, utilizing the mutual reinforcement between web documents and their associated social contexts to generate summaries. Hu et al. [4] firstly scored the importance of each

comment based on three relations (namely, topic, quotation, and mention) with a graph based method and a tensor based method, and then extracted sentences from the given web document with two approaches: feature-biased approach and uniform-document approach. For the former one, words appearing in comments but not in the document do not contribute to scoring sentences, which is more tolerant to noise in comments. For the latter one, when there is no or very few comments, the problem naturally degrades to single document summarization.

### 3 Approach Details

In this section, we propose a multi-aspect co-feedback ranking model (Multi-AspectCoRank), which utilizes the mutual reinforcement on the correlations based on each aspect. This MultiAspectCoRank model incorporates both the document content and its corresponding comments together to generate a high quality summary.

#### 3.1 Multiple Aspects Extraction

Comments represent the readers' feedback about the web document. However, not all comments are directed to the content of the document. In order to derive useful information from comments, we try to fuse the document sentences and its affiliated comments together and extract multiple aspects with clustering technique. In this way, most related comments will be clustered with document sentences together.

Affinity propagation is a graph based cluster algorithm [2]. Compared with traditional clustering algorithms such as K-means, singular value decomposition (SVD), graph-based approach using affinity propagation performs best in clustering short text data with minimal cluster error [11]. Affinity propagation takes as input a collection of real-valued similarities between data points. In order to identify the exemplars (centers of clusters) of a document, two kinds of message exchanged between data points: one is called "responsibility"  $\gamma(i, j)$ , sent from data point  $i$  to candidate exemplar point  $j$ , the other is "availability"  $\alpha(i, j)$ , sent from candidate exemplar point  $j$  to point  $i$ . To begin with, the availabilities are initialized to zero:  $\alpha(i, j) = 0$ . Then the responsibilities are computed using the rule:

$$\gamma(i, j) \leftarrow \text{sim}(i, j) - \max_{j' \neq j} \{\alpha(i, j') + \text{sim}(i, j')\} \quad (1)$$

where  $\text{sim}(i, j)$  represents the similarity between data point  $i$  and  $j$ . Whereas the above responsibility update lets all candidate exemplars compete for ownership of a data point, the following availability update gathers evidence from data points as to whether each candidate exemplar would make a good exemplar:

$$\alpha(i, j) \leftarrow \min\{0, \gamma(j, j) + \sum_{i' \notin \{i, j\}} \max\{0, \gamma(i', j)\}\} \quad (2)$$

Given a web document  $D$ , we denote the graph of the whole sentences set as  $G = (S, E_{ss})$  to represent the whole document including comments.  $S = \{s_i | 0 \leq i \leq n\}$  is the set of vertices in the graph and stands for the set of both the document sentences and its affiliated comment sentences, and  $E_{ss} = \{e_{ij} | s_i, s_j \in S, i \neq j\}$  corresponds to the relationship between each pair of sentences. Each  $e_{ij}$  is associated with a weight  $\omega_{ij}$  which indicates the similarity of the pair of sentences. The weight is computed by using the standard cosine measure between two sentences as follows:

$$\omega(i, j) = \text{sim}_{\text{cosine}}(s_i, s_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{|\vec{v}_i| \times |\vec{v}_j|} \quad (3)$$

where  $\vec{v}_i$  and  $\vec{v}_j$  are the term vectors corresponding to the sentence  $s_i$  and  $s_j$ , respectively. The graph we propose to build is undirected so we have  $\omega_{ij} = \omega_{ji}$  here and define  $\omega_{ii} = 0$  to avoid self transition. *TF-ISF* (term frequency - inverted sentence frequency) value of each term is applied to describe the elements in the sentence vector. Then the weight of each pair of sentences in the web document can be denoted as a symmetric matrix  $W$ , which is used as the input of the affinity propagation. For the affinity propagation with  $W$  and a specified number of exemplars  $K$ , the message passing procedure will be terminated after a fixed number of iterations. And then we get the result of  $K$  clusters. Each cluster is a sentence list standing for an aspect of the document.

### 3.2 Sentence Ranking Based on Each Aspect

As we get  $K$  aspects, each aspect can be viewed as a part of the document. We need to estimate the probability score  $p(s|A)$  of a sentence  $s$  being selected as a summary sentence given an aspect  $A$ . Instead of simply comparing the relevance between the sentence  $s$  and the aspect  $A$ , we take this as a task based on the whole background, where both the similarities between the sentence  $s$  and other sentences in the document and the similarity between  $s$  and the aspect  $A$  are considered. Formally,  $p(s|A)$  is computed by the following formula:

$$p(s|A) = d \frac{\sum_{z \in S_D} \text{sim}(s, z)}{\sum_{q \in S_D} \text{sim}(q, z)} p(z|A) + (1 - d) \frac{\text{rel}(s, A)}{\sum_{z \in S_D} \text{rel}(z, A)} \quad (4)$$

where  $A$  is a set of sentences standing for an aspect, and  $S_D$  is the set of all sentences in the document,  $d$  is a trade-off parameter in the interval  $[0, 1]$ , which is used to specify the relative contribution of the two parts in Eq.(4). For bigger value of  $d$ , more importance is given to the similarities between the sentence  $s$  and other sentences in the document  $D$  than that between  $s$  and the aspect  $A$ . The denominators in both parts are used for normalization. The formula in Eq.(4) can be written as:

$$p(k+1) = M^T p(k) \quad (5)$$

$$M = dC + (1 - d)R \quad (6)$$

where  $M$ ,  $C$  and  $R$  are all square matrices. Elements in  $C$  represent the similarities between sentences in the document. Elements in  $R$  represent the relevance

between each sentence in the document and each aspect.  $k$  represents the  $k$ th iteration, and  $p = [p_1, p_2, \dots, p_N]^T$  is the vector of sentence ranking scores that we are looking for, which corresponds to the stationary distribution of the matrix  $M$ . The iteration is guaranteed to converge to a unique stationary distribution given that  $M$  is a stochastic matrix.

To calculate the similarities between the sentences in the document, we use the Cosine similarity. To calculate the relevance between  $s$  and an aspect  $A$ , we use the average Cosine similarity as follows:

$$rel(s, A) = \frac{\sum_{a \in A} sim_{cosine}(s, a)}{count(A)} \quad (7)$$

where  $a$  represents a sentence belongs to the aspect  $A$ ,  $count(A)$  stands for the number of sentences in aspect  $A$ .

In order to initialize each  $p(s|A)$ , we can regard each aspect as a short document and construct a language model to estimate it using Dirichlet prior smoothing as follows:

$$p(s|A) = \sum_{w \in s} p(w|A) \quad (8)$$

$$p(w|A) = \frac{c(w, A) + u_s * p(w|B)}{|A| + u_s} \quad (9)$$

where  $|A|$  is the length of the sentence aspect  $A$ ,  $c(w, A)$  is the count of word  $w$  in  $A$ ,  $u_s$  is the smoothing parameter,  $p(w|B)$  is the background model used as smoothing factor. Generally  $p(w|B)$  is estimated by the whole document  $D$ , using following formula:

$$p(w|B) = \frac{c(w, D)}{|D|} \quad (10)$$

After estimating the score of each sentence in the document, we can get  $K$  lists of scores  $list_i = \{r_i(s_j) | 1 \leq j \leq |S_D|, 1 \leq i \leq K\}$ , where  $r_i(s_j)$  represents the ranking result of sentence  $s_j$  in ranking list  $list_i$ ,  $|S_D|$  is the number of sentences in the document, no comments included,  $K$  is the number of aspects.

### 3.3 Co-feedback Ranking Based on Multiple Aspects

Each ranking list indicates a kind of candidate summaries from one angle, but cannot determine the final ranking results. In this subsection, we are aiming to rank all the document sentences based on the  $K$  groups of ranking lists and selecting the top- $N$  ranked ones to generate summaries. Each aspect can be regarded as a part of the document. The summary we are aiming to generate should base on the whole document. So how to combine the ranking scores based on each aspect to get an overall score as the ranking criterion is the key problem to solve. Traditional strategy is to combine different ranking lists into a unified one with a linear function. However it fails to make full use of the information provided by the different ranking lists and neglects the interactions among them

before combination. Also we believe that each aspect based ranking list is able to provide valuable information to help promote other aspects based ranking performance with the mutual ranking refinement, which, in turn, may lead to an overall improvement in ranking.

Co-feedback ranking framework [14] is a good mutual reinforcement method to combine multiple ranking results into a unified one. In this paper, we select the top- $N$  ranked sentences from each ranking list as feedback to update other aspect based ranking results according to the relevance of sentences. So we update the ranking result of  $s_j$  in ranking list  $list_i$  as follows:

$$r_i^{t+1}(s_j) = \eta r_i^t(s_j) + (1 - \eta) \pi_{ij}^t, \quad 1 \leq i \leq K, 1 \leq j \leq |S_D| \quad (11)$$

where  $\eta$  is a balance parameter which can be viewed as the proportion of the dependence of the new ranking results on its initial ranking results.  $\pi_{ij}$  is the feedback value from other ranking lists to  $s_j$  of ranking list  $list_i$ .  $t$  represents the  $t$ th iteration.  $\pi_{ij}^t$  is computed as follows:

$$\pi_{ij}^t = \sum_{k=1, k \neq i}^K \frac{\alpha_k}{\sum_{k=1, k \neq j}^K \alpha_k} \pi_{ikj}^t \quad (12)$$

$$\pi_{ikj}^t = \sum_{l=1}^N \frac{\sum_{h=0}^N h - l}{\sum_{h=0}^N h} sim(s_j, \tau_k^l) \quad (13)$$

where  $\pi_{ikj}^t$  represents the feedback value on sentence  $s_j$  of ranking list  $list_i$  from ranking list  $list_k$ .  $\alpha_k$  represents the number of sentences in aspect  $k$ .  $\tau_k^N$  represents the top  $N$  sentences selected from ranking list  $list_k$ .  $sim(s_j, \tau_k^N)$  is defined as the similarity between  $s_j$  and the top  $N$  sentences from ranking list  $list_k$ . In this paper  $N$  is set to 10 as 10 sentences are basically sufficient for the summarization task we work on. This process continues iteratively when the top  $N$  results of each ranking list don't change. We get the final ranking results based on these ranking lists as follows:

$$r(s_j) = \sum_{i=1}^K \frac{\alpha_i}{|S_D|} r_i(s_j) \quad 1 \leq j \leq |S_D| \quad (14)$$

This co-ranking method is designed based on the principle that the new ranking result of the sentence  $s$  from one aspect consists of two parts: one is the initial ranking result based on the aspect, the other part is the similarities between  $s$  and the top  $N$  feedback sentences provided by other ranking lists. The top  $N$  ranked sentences based on one aspect are presumed to be highly supported by that aspect, which are more likely to be chosen as summary sentences. So these top  $N$  sentences are selected as feedback to promote the weight of these sentences in other ranking lists. Through this co-feedback and mutual reinforcement, these ranking lists are considered to be more reliable ranking results.

After we get the final ranking lists, we apply the variant version of MMR algorithm proposed in [12]. This method is to penalize the sentences that highly overlap with the sentences that have been chosen as the summary. In this way, we can choose more informative but less redundant sentences as the final summary.

## 4 Experiments

### 4.1 Dataset

To evaluate our method for comments-oriented document summarization, we conduct our experiments on the dataset used in [4]. This dataset was collected from two famous blogs, i.e., Cosmic Variance<sup>1</sup> and IEBlog<sup>2</sup>, both receiving large number of comments. To guarantee the validity of the data, we randomly picked up 50 posts from each blog, and constructed our experiment dataset with these 100 articles. Each article consists of two parts, document content and its affiliated comments. Table 1 gives the brief description of the 100 posts. To generate reference summaries, each post has 7 labeled sentences by 4 human summarizers.

**Table 1.** A brief description of the dataset

Source	Cosmic Variance and IEBlog
Number of blog posts	100
Average sentence number per post	22.22
Average sentence length	19.71
Average comment number per post	26.04

### 4.2 Performance Metric

We use the ROUGE<sup>3</sup> [6] (Recall Oriented Understudy for Gisting Evaluation) toolkit to evaluate our proposed method, which has been widely applied for summarization evaluation. It evaluates the quality of a summary by counting the overlapping units between the candidate summary and reference summaries. There are many kinds of ROUGE metrics to measure the system-generated summarization such as ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-U, of which the most important one is ROUGE-N with 3 sub-metrics: precision, recall, and F-score.

$$ROUGE - N = \frac{\sum_{S \in RS} \sum_{N-gram \in S} Count_{match}(N - gram)}{\sum_{S \in RS} \sum_{N-gram \in S} Count(N - gram)} \quad (15)$$

where RS represents the reference summaries.  $N-gram \in RS$  in the metrics denotes the N-grams in reference summary.  $Count_{match}(N - gram)$  is the maximum

<sup>1</sup> <http://cosmicvariance.com>

<sup>2</sup> <http://blogs.msdn.com/ie/>

<sup>3</sup> <http://www.isi.edu/licensed-sw/see/rouge/>



number of N-grams co-occurring in the candidate summary and in the set of reference summaries.  $Count(N-gram)$  is the number of N-grams in the reference summaries.

The ROUGE toolkit can report separate scores for 1, 2, 3, and 4-gram. In the experimental results we report four ROUGE F-measure scores: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), ROUGE-4 (extension of ROUGE-S, which is the skip-bigram co-occurrences statistics), and ROUGE-W (based on the weighted longest common subsequence) metrics. The higher the ROUGE scores, the similar the two summaries are.

**Table 2.** Results of different systems

Methods	ROUGE-1	ROUGE-2	ROUGE-4	ROUGE-W
Random	0.3944	0.2984	0.2705	0.3075
LexPageRank	0.5961	0.4593	0.4627	0.5601
LexPageRank**	0.5731	0.4386	0.4416	0.5174
MultiAspctCoRank	0.6295	0.4911	0.4893	0.5847

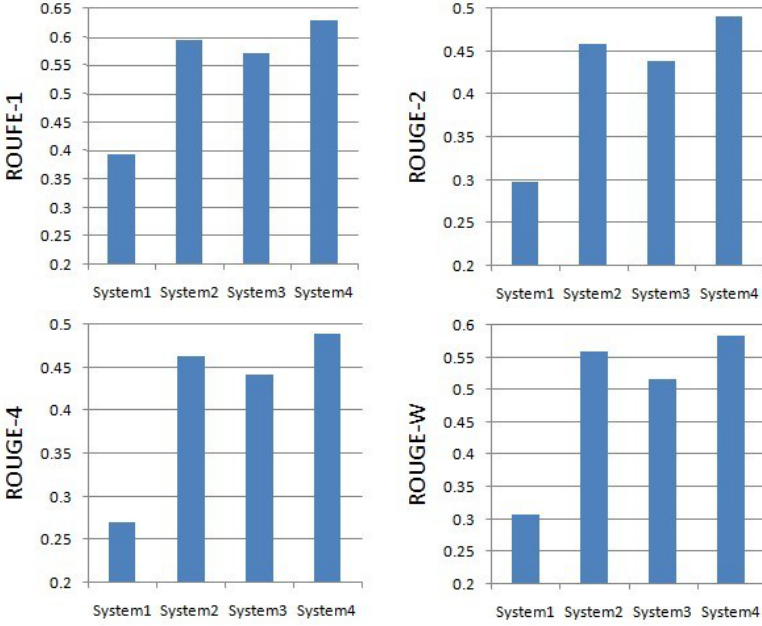
### 4.3 Experimental Results

Given a web document and its corresponding comments, we first decompose them into sentences and then remove the stop-words and words stemming are performed. After these steps, we implement the following algorithms which are widely used in comments-oriented document summarization. We conduct the same preprocessing for all algorithms for fairness.

- Random (system1): The method selects sentences randomly for the sentence set including comments.
- LexPageRank [1] (system2): This system applies Cosine similarity in standard LexPageRank method based on the document content sentence set (no comments is given) to summarize the document.
- LexPageRank\*\* (system3): This system applies Cosine similarity in standard LexPageRank method based on the whole document sentence set (including comments) to generate the summary.
- MultiAspectCoRank (system4): As we have showed, this method extracts multiple aspects with affinity propagation firstly, and then utilizes mutual reinforcement co-feedback ranking framework based on these aspects to promote the ranking performance.

Table 2 and Fig 1 show the performance of these systems on a same dataset. The parameters of MultiAspectCoRank approach are set as follows: the number of aspects  $K = 5$ , the balance parameter  $u_s = 0.1$ ,  $d = 0.6$  and  $\eta = 0.4$ . From Table 2 and Fig 1 we have following observations:

- Generally, the Random gets the worst performance;
- From the results of system 2 and system 3, both of these two systems use LexPageRank, but the results of system 2 in which no comments is given is better than those of system 3 which combines document contents and comments



**Fig. 1.** Comparison results in ROUGE

together. This is mainly because most of the comments are not directed to the content of the post and comments are not equally useful for identifying important document sentences.

• From Fig 1, we can see that on the same dataset, our system (system 4) outperforms other systems on different evaluation measures, which indicates the effectiveness of the proposed MultiAspectCoRank model.

All the comparison results suggest that it's effective to derive useful information from comments by fusing the document sentences and its corresponding comments together and detecting multiple aspects from them. And it's better to promote the performance when utilizing mutual reinforcement co-feedback ranking method on the same data based on multiple aspects.

#### 4.4 Parameter Tuning

There are mainly four parameters in our proposed method: number of clusters  $K$ , balance parameter  $u_s$ ,  $d$  and  $\eta$ . In this section, we are trying to evaluate the influence of these four parameters on the experimental results respectively. First we compare the four kinds of ROUGE measure results with aspect number  $K$  ranging from 3 to 11 in Fig 2. It is observed that when the aspect number  $K$  is 7 the performance is best. In the same way, the four kinds of ROUGE measure results with balance parameter  $u_s$ ,  $d$  and  $\eta$  are shown in Fig 2. It is obvious that the value of parameter  $u_s$  has less effect on the performance of our proposed

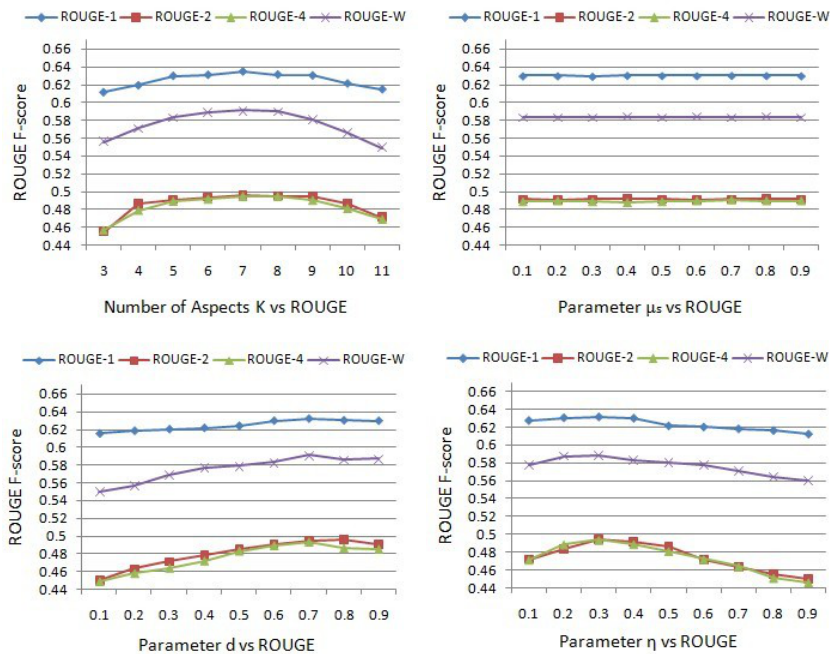


Fig. 2. Parameters vs ROUGE

method, and when the balance parameter  $d$  is set to 0.7 and  $\eta$  is set to 0.3 this method performs best.

5 Conclusion

In this work, we propose a novel model named MultiAspectCoRank for comments-oriented document summarization task. More specifically, by fusing the document content and its comments together, we extract multiple aspects from the given web document, and then utilize mutual reinforcement co-feedback ranking method to promote the performance. Experimental results on a set of real-world blog data prove the effectiveness of our proposed method in improving the summary quality. In future work, we will try to find more and deeper relationships between document contents and comments.

**Acknowledgments.** We thank the anonymous reviewers for their valuable and constructive comments. This work is financially supported by NSFC Grant 61272340.

## References

1. Erkan, G., Radev, D.: Lexpagerank: Prestige in multi-document text summarization. In: *Proceedings of EMNLP*, vol. 4 (2004)
2. Frey, B., Dueck, D.: Clustering by passing messages between data points. *Science* 315(5814), 972–976 (2007)
3. Hu, M., Sun, A., Lim, E.: Comments-oriented blog summarization by sentence extraction. In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pp. 901–904. ACM (2007)
4. Hu, M., Sun, A., Lim, E.: Comments-oriented document summarization: understanding documents with readers feedback. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 291–298. Citeseer (2008)
5. Lin, C., Hovy, E.: From single to multi-document summarization: A prototype system and its evaluation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 457–464. Association for Computational Linguistics (2002)
6. Lin, C., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 71–78. Association for Computational Linguistics (2003)
7. Mei, Q., Guo, J., Radev, D.: Divrank: the interplay of prestige and diversity in information networks. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1009–1018. ACM (2010)
8. Nenkova, A., Vanderwende, L., McKeown, K.: A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 573–580. ACM (2006)
9. Ouyang, Y., Li, W., Lu, Q., Zhang, R.: A study on position information in document summarization. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 919–927. Association for Computational Linguistics (2010)
10. Radev, D., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. *Information Processing & Management* 40(6), 919–938 (2004)
11. Rangrej, A., Kulkarni, S., Tendulkar, A.: Comparative study of clustering techniques for short text documents. In: *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 111–112. ACM (2011)
12. Wan, X.: Document-based hits model for multi-document summarization. In: Ho, T.-B., Zhou, Z.-H. (eds.) *PRICAI 2008*. LNCS (LNAI), vol. 5351, pp. 454–465. Springer, Heidelberg (2008)
13. Wan, X., Yang, J., Xiao, J.: Manifold-ranking based topic-focused multi-document summarization. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2903–2908. Morgan Kaufmann Publishers Inc. (2007)
14. Wei, F., Li, W., He, Y.: Co-feedback ranking for query-focused summarization. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 117–120. Association for Computational Linguistics (2009)
15. Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., Li, J.: Social context summarization. In: *Proceedings of the 34th ACM SIGIR Conference* (2011)