# TAC 2009 Update Summarization of ICL

**Sujian Li, Wei Wang, Yongwei Zhang**
**Key Laboratory of Computational Linguistics, Peking University**
{lisujian, wwei, zhangyongwei}@pku.edu.cn

## Abstract

For the update summarization task of TAC 2009, we submitted two runs using two different methods. The first one is manifold ranking method, which models all sentences as a graph. The topic description is deemed as the only labeled node and assigned with an initial score, then the scores of all the sentences in the documents are learned by spreading the initial score on the graph. The second method is called clustering method, which divides all sentences in one topic into several clusters. A good summary should care for all these clusters. The more significant a cluster appears, the more likely sentences will be chosen from this cluster. With these two methods, we conduct some experiments and discuss what has an effect on the performance of the update summarization task.

## 1. Introduction

The TAC[1] 2009 update summarization task is the same to that in TAC 2008 [Hoa 2008], which aims at generating short (no more than 100 words) fluent multi-document summaries of news articles with or without some related earlier articles considered. Our TAC 2008 summarization system is designed with feature-based sentence-extractive framework[Li 2008]. However, feature selection is nontrivial and weights tuning is difficult. For TAC 2009 summarization task, we changed our method in order to avoid the problem of feature weighting. The first one is called manifold ranking method, which models all the sentences including the topic description as a graph. The topic description is deemed as the only labeled node and assigned with an initial score. Then the scores of all the sentences in the documents are learned by spreading the initial score on the graph. When generating update summaries for document set *B*, sentences are penalized by their content overlap with document set *A*. The second method is called clustering method, in which all sentences in one topic are divided into several clusters. A good summary should care for all these clusters. The more significant a cluster appears, the more likely sentences will be chosen from this cluster. Here K-means method is used to conduct clustering and LexPagerank method [Erkan 2004] is used to score sentences within a cluster.

The rest of the paper is organized as follows. Section 2 briefly describes the two methods adopted. Section 3 presents our evaluation results in TAC 2009, and furthermore conducts more experiments to discuss what has an effect on the update summarization task. Section 4 shows the future work and concludes the paper.

## 2. Our methods

For a given topic, all the documents are separated into two docsets *A* and *B*. The documents in docset *A* are summarized directly and the documents in docset *B* are summarized with the assumption that the documents in

[1]http://www.nist.gov/tac/

docset *A* have been read. Our summarization system is designed with the extractive framework and important sentences are extracted from docset *A* and docset *B* respectively. For TAC 2008, a feature based method is adopted to rank the importance of sentences [Li 2008]. However, it is difficult to explore appropriate features and tune feature weights. Of course supervised methods can be used to solve this problem and a large amount of training corpus need to be collected. Then, for TAC 2009, we try unsupervised methods to rank the importance of sentences. Two graph based methods are considered. One is manifold ranking method proposed in [Zhou 2003, Wan 2007] and the other one is query-sensitive pagerank scoring method. In addition, documents for one topic usually are represented by several subtopics. Then we hope to use clustering techniques to divide documents into several clusters and each cluster can be seen as one subtopic, from which sentences are selected. These three methods, named Manifold [2], pagerank and clustering respectively, are experimented on TAC 2008 data, and the results are listed in Table 1. The feature-based method we adopted in TAC 2008 and the best system (TOP1) in TAC 2008 are also put in Table 1 for comparison.

| | R-2 | R-SU4 |
|---|---|---|
| TOP1[3] | 0.10382 (0.09530-0.11302) | 0.13625 (0.12875-0.14402) |
| Manifold | 0.09982 (0.08822 - 0.11183) | 0.13339 (0.12369 - 0.14431) |
| Pagerank | 0.09624 (0.08531 - 0.10871) | 0.13020 (0.12092 - 0.14078) |
| Clustering | 0.08996(0.07835 - 0.10224) | 0.12786(0.11814 - 0.13818) |
| Feature_based[4] | 0.08957 (0.08023 -0.10001) | 0.12699 (0.11857 - 0.13645) |

**Table 1: Comparison of different methods on TAC 2008**

From Table 1, we can see that all the three unsupervised methods are better than the feature_based method. Especially, manifold ranking method can almost reach the performance of the TOP1 system. Because only two runs can be submitted, we chose two different methods – manifold ranking and clustering method for our TAC 2009 update summarization task. Section 2.1 and section 2.2 briefly overview these two methods.

### 2.1 Manifold Ranking method

For a topic, given a set of data points $\chi = \{x_0, x_1, ..., x_n\} \subset R^m$, the first point $x_0$ represents the topic description and the rest $n$ points represent the sentences in the documents. Let $f: \chi \rightarrow R$ denote a ranking function which assigns each point $x_i$ ($0 \le i \le n$) a ranking value $f_i$. We can view $f$ as a vector $f=[f_0,\ldots,f_n]$. We also define a vector $y=[y_0,\ldots,y_n]$, in which $y_0=1$ because $x_0$ is the topic description and $y_i=0$ ($0 \le i \le n$) for all the sentences in the documents. The manifold ranking algorithm is as follows:

1. Compute the pair-wise similarity values between sentences using the standard Cosine measure. The weight associated with term $t$ is calculated with the $tf_t * isf_t$, where $tf_t$ is the frequency of term $t$ in the sentence and $isf_t$ is the inverse sentence frequency of term $t$, i.e. $1+\log(N/n_t)$, where $N$ is the total number of

[2] For document set B, Manifold method is used directly and the penalty factor is set as 0.

[3] Top1 provides the best Rouge score on TAC 2008 evaluation.

[4] This data is the best result experimented using feature based methods and refer to Run 4 in our TAC 2008 report[Li 2008].

sentences and $n_t$ is the number of the sentences containing term $t$. Given two sentences $s_i$ and $s_j$, the Cosine similarity is denoted as $sim(s_i,s_j)$, computed as the normalized inner product of the corresponding term vectors.

2. Connect any two sentences with an edge if their similarity value exceeds 0. We define the affinity matrix $W$ by $W_{ij}=sim(s_i,s_j)$ if there is an edge linking $s_i$ and $s_j$. Let $W_{ii}=0$ to avoid loops in the graph.

3. Symmetrically normalize $W$ by $S=D^{-1/2}WD^{-1/2}$ in which $D$ is the diagonal matrix with $(i,i)$-element equal to the sum of the $i$-th row of $W$.

4. Iterate $f(t+1)= \alpha Sf(t)+(1-\alpha)y$. until convergence, where $\alpha$ is a parameter in [0,1).

5. Let $f_i^*$ denote the limit of the sequence $\{f_i(t)\}$. Rank each sentences $s_i$ $(0\le i\le n)$ according its ranking score $f_i$

With manifold ranking method, we can directly score all the sentences in docset A and generate the summary using the sentences with the highest scores.

To summarize docset *B*, we need care for the update content of the topic and penalize the information overlap between docset *B* and the older content in docset *A*. Then a problem appears: should the older content represent the documents or the summary for docset *A*? Another problem is the computation of the overlap. Should average similarity or maximum similarity strategy be used to compute the overlap. Then there will be four kind of possible combination with the following formula.

$$s(s_i) = f_i - b \bullet overlap(s_i, old\_content) \quad (1)$$

Where $b$ is a penalty factor between 0 and 1, *old_content* can be docset *A* or summary *A*, and the overlap can be computed by *maxsim* or *avgsim* methods. *Maxsim* computes all the similarity values between $s_i$ and each sentence in the *old_content*, and returns the highest one, while *avgsim* returns the average of all the scores. Still, the Cosine similarity is adopted.

### 2.2 Clustering method

For this method, the hypothesis is that one event is usually described from several subtopics, which can be represented as a cluster. A good summary should care for all these subtopics. The more important a subtopic is, more preferred the sentences in the subtopic. The principle of choosing sentences within a subtopic is, more sentences a sentence is closely related to, more important the sentence is. We use a clustering method to implement the hypothesis. Here K-means method is employed. LexPagerank method is used to score sentences and the sentence with the highest score is chosen into a summary. The algorithm is as follows.

1. All the sentences are collected and those which relate to the topic description are kept as a sentence set *S*.

2. Compute the pair-wise similarity values between sentences in Set *S* using the standard Cosine measure, as the manifold ranking method does.

3. We employ k-means method to cluster all sentences in set *S*. The number of clusters is set 12 by experience. Then, according to their size and their similarity to the topic description (*td*), each of these 12 clusters is given a weight using the formula:

$W(C_i)=0.1*Sim(ci,td)+0.9*|C_i|$, where *Ci* is a cluster. Finally, these clusters are sorted with their weight decreasing.

4. For each cluster, we construct an affinity matrix and score each sentence with Lexpagerank method. Then sentence $s_i$ has a score named lexpkscore($s_i$).

5. To generate the summary, we choose the cluster with the highest weight, from which we further select the sentence with the highest score. The

sentence score is computed as:

$S(s_i) = 0.15*\text{sim}(s_i, td) + 0.85*\text{lexpkscore}(s_i)$

We use MMR algorithm to determine whether the sentence should enter into the summary.

6. After one cluster is selected, the weight of this cluster is reduced by half. We sort all these clusters by the new weights.

7. If the summary reach 100 words, the procedure stops, else go to 5.

In this paper, we only describe our clustering method for the summarization task and do not discuss it in detail.

## 3 Experiments and Discussion

TAC 2009 test datasets comprises of 44 topics. Each topic has a topic description (title and narrative) and 20 relevant documents which have been divided evenly into 2 sets: docset *A* and docset *B*. This section will first introduce our evaluation result on TAC 2009. Furthermore, with these data, more experiments are conducted on manifold ranking method to discuss the following problems: (1) What affect the summarization of docset *B* more - documents in docset *A* (Set *A*) or summaries for *A* (Sum A)? (2) Is Average or Maximum strategy better for penalizing the overlap with the older content?

### 3.1 Evaluation on TAC 2009

NIST assessors wrote 4 model summaries for each document set. All submitted systems are either manually or automatically evaluated, including linguistic quality, responsiveness, ROUGE-2, ROUGE-SU4 [Lin 2004], BE and Pyramid. Each system is required to submit no more than two runs. We submitted two runs, named *Run1* and *Run2* respectively. *Run1* uses the manifold ranking method directly and *Run2* uses the clustering method. Table 2 illustrates the automatic evaluation results of our system, the best submitted system (named TOP1) and Baseline 3 (named BASE) provided by the organizer, where Baseline 3 returns a summary consisting of sentences that have been manually selected from a docset. The manual evaluation results are listed in Table 3. From Table 2 and Table 3, we can see that our results are better for automatic evaluation than for manual evaluation. This is because our system just extracts the sentences and does not conduct any post-processing. Our automatic evaluation can almost reach the performance of Baseline 3. From this point, we can conclude that, in order to achieve better results, preprocessing and postprocessing are necessary.

| | R-2 | R-SU4 | BE |
|---|---|---|---|
| Top1_A | 0.12163 (0.10974 - 0.13486) | 0.15101 ( 0.14009 - 0.16271) | 0.06356 ( 0.05245 - 0.07479) |
| BASE_A | 0.10655 (0.09636 - 0.11819) | 0.13843 ( 0.12861 - 0.14890) | 0.05350 (0.04449 - 0.06375) |
| Run1_A | 0.10440 (0.09286 – 0.11630) | 0.13960 (0.12997 – 0.15026) | 0.05580 ( 0.04596 – 0.06644) |
| Run2_A | 0.09284 (0.08418 – 0.10113) | 0.13011 (0.12245 – 0.13752) | 0.04970 ( 0.04103 – 0.05881) |
| Top1_B | 0.10386 ( 0.09190 – 0.11580) | 0.13948 (0.12877 – 0.14984) | 0.06389 ( 0.05200 – 0.07581) |
| BASE_B | 0.09820 (0.08633 - 0.11004) | 0.13631 ( 0.12517 - 0.14679) | 0.05690 (0.04687 - 0.06707) |
| Run1_B | 0.09093 (0.07994 - 0.10236) | 0.13008 (0.11925 - 0.14093) | 0.04714 ( 0.03739 - 0.05634) |
| Run2_B | 0.07668 (0.06640 - 0.08732) | 0.11903 (0.10924 - 0.12934) | 0.03974 ( 0.03136 - 0.04864) |

**Table 2: Automatic Evaluation in TAC 2009**

| | Pyramid | Ling. quality | Resp |
|---|---|---|---|
| Top 1_A | 0.383 | 5.932 | 5.159 |
| Run1_A | 0.331 | 4.705 | 4.341 |
| Run2_A | 0.275 | 4.136 | 4.114 |
| Top 1_B | 0.307 | 5.886 | 5.023 |
| Run1_B | 0.258 | 4.75 | 4.25 |
| Run2_B | 0.197 | 4.318 | 3.75 |

**Table 3: Manual Evaluation in TAC 2009**

### 3.2 Discussion

This subsection mainly experiments the four combinations involved in formula (1) and the penalty factor *b*. Table 4, 5, 6 and 7 illustrate the combination of docset *A* or summary *A* with maximum or average strategy respectively. From Table 4 and 6, we can see that the penalty factor (*b*) value can more affect the results using maximum strategy. However, the average strategy is insensitive to the penalty factor, which can be seen from Table 5 and 7. To contrast using documents in docset *A* or summaries for docset *A*, using summaries has a more stable performance.

| *b* | R-2 | R-SU4 |
|---|---|---|
| 0.1 | 0.03980 | 0.08353 |
| 0.05 | 0.05440 | 0.09861 |
| 0.01 | 0.09040 | 0.12957 |
| 0.005 | **0.09249** | **0.13227** |
| 0.001 | 0.09139 | 0.13047 |

**Table 4: Set A + Maximum**

| *b* | R-2 | R-SU4 |
|---|---|---|
| 0.1 | 0.07730 | 0.11961 |
| 0.05 | 0.08813 | 0.12924 |
| 0.01 | 0.08868 | 0.12891 |
| 0.005 | 0.08940 | 0.12876 |
| 0.001 | **0.09091** | **0.12997** |

**Table 5: Set A + Average**

| *b* | R-2 | R-SU4 |
|---|---|---|
| 0.1 | 0.06085 | 0.10195 |
| 0.05 | 0.06875 | 0.11105 |
| 0.01 | 0.08453 | 0.12479 |
| 0.005 | 0.08684 | 0.12788 |
| 0.001 | **0.09071** | **0.13034** |

**Table 6: Sum A + Maximum**

| *b* | R-2 | R-SU4 |
|---|---|---|
| 0.1 | 0.09121 | 0.13038 |
| 0.05 | 0.09096 | 0.13014 |
| 0.01 | 0.09147 | 0.13043 |
| 0.005 | 0.09147 | 0.13036 |
| 0.001 | **0.09154** | **0.13036** |

**Table 7: Sum A + Average**

Table 8 compares the four combination (*b*=0.001) with *Run1* (with no penalty), and these five results do not have obvious gap. From Table 4 -7, we can see that, the lower the penalty factor *b*, the better the performance. *Run1* with *b*=0 have a comparable performance to those results with *b*=0.001. We find that a bigger penalty for the overlap does not have a positive effect on the results. When no penalty is used, the summarization performance can also reach a good result.

| | R-2 | R-SU4 |
|---|---|---|
| Set A + Avg | 0.09091 | 0.12997 |
| Set A + Max | 0.09139 | 0.13047 |
| Sum A + Avg | 0.09154 | 0.13036 |
| Sum A + Max | 0.09071 | 0.13034 |
| Run1 | 0.09093 | 0.13008 |

**Table 8: Evaluation Comparison (b=0.05)**

## 5 Conclusions and Future Work

In this paper, we introduce the two methods, manifold ranking and clustering method, which are adopted for the update summarization task in TAC 2009. These two methods can overcome the problem of searching a large set of

appropriate features and tuning feature weights. In fact, these two methods are not novel. However, the experimental results show that they are effective for summarization. Then, our emphasis is mainly put on the part of update summary. Through TAC 2009 evaluation and further experiments, we find that summarization for docset *B* does not care for much the overlap with the content in docset *A*.

In our future work, more methods will be tried for update summarization task. We will consider more about how docset *A* can help better summarizing docset *B*.

## Acknowledgements

This work is supported by NSFC programs (No: 60603093, 60875042 and **9**0920011), and 973 National Basic Research Program of China (2004CB318102).

### References

Erkan G. and Radev D.. 2004. Lexpagerank: Prestige in multi document text summarization. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, July.

Hoa T. D.. 2008, Overview of the TAC 2008 Update Summarization Task. Text Analysis Conference 2008 http://www.nist.gov/tac/, 2008.

Li, S.J, Wang W., Wang C.. 2008, TAC 2008 Update Summarization Task of ICL, In Proceedings of TAC 2008.

Li, S.J., OuYang, Y., Wang, W., Sun B., 2007. Multi-document Summarization Using Support Vector Regression, In Proceedings of DUC 2007.

Lin.C.Y., 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization, Barcelona. ACL.

Radev, D. R., Jing, H.Y., and Budzikowska, M., 2000. Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies. Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics, Seattle, WA, April 2000.

Wan X., Yang J. and Xiao J.. 2007. Manifold-Ranking Based Topic-Focused Multi-Document Summarization. IJCAI 2007, pp. 2903-2908.

Zhou, D., Weston J., Gretton A., Bousquet O. and SchOlkopf B.. 2003. Ranking on data manifolds. In Proceedings of NIPS’2003.