A Comprehensive Method for Text Summarization Based on Latent Semantic Analysis

Yingjie Wang and Jun Ma

School of Computer Science and Technology, Shandong University, Jinan, China
yingj_wang@hotmail.com, majun@sdu.edu.cn

Abstract. Text summarization aims at getting the most important content in a condensed form from a given document while retains the semantic information of the text to a large extent. It is considered to be an effective way of tackling information overload. There exist lots of text summarization approaches which are based on Latent Semantic Analysis (LSA). However, none of the previous methods consider the term description of the topic. In this paper, we propose a comprehensive LSA-based text summarization algorithm that combines term description with sentence description for each topic. We also put forward a new way to create the term by sentence matrix. The effectiveness of our method is proved by experimental results. On the summarization performance, our approach obtains higher ROUGE scores than several well known methods.

Keywords: Text Summarization, Latent Semantic Analysis, Singular Value Decomposition.

1 Introduction

The widespread use of the Internet has dramatically increased the amount of accessible information and it becomes difficult for users to sift through the multitude of sources to find out the right document. With the help of search engine, the majority of irrelevant documents are filtered out, however, users still hesitate to determine which particular search result should navigate to. Automated summarizing system can be used as an instrument for deciding whether a document is related to their needs.

The summary of a document is defined as: a text that is produced from the document that conveys important information in the original text, and that is no longer than half of the original text and usually significantly less than that [1].

Automatically generating the summary of a document has long been studied since 1950s and it is still a research hotpot until now [2, 3, 4]. One of the most famous approaches is using LSA [5, 6, 7, 8, 9, 10] to get the ideal summary.

The foundational work that uses LSA for text summarization selects one sentence for each topic according to topic importance [6]. The work in [7] starts with calculation of the length of each sentence vector and then chooses the longest sentences as the summary. In the work [9], the length strategy proposed in [7] is improved and a cross method is proposed. In [8], for each topic, the number of sentences to be collected is determined by getting the percentage of the related singular values over the sum of all singular values.

G. Zhou et al. (Eds.): NLPCC 2013, CCIS 400, pp. 394–401, 2013.

However, there are some disadvantages of the previous algorithms. The main drawback is that sentences that are closely related to the chosen topic somehow but do not have the highest index value will not be selected. Also, all chosen topics are composed of only one sentence [6], whereas the single sentence fails to fully express the topic. The length strategy [7, 9] requires a method of deciding how many LSA dimensions to include in the latent space. For the work in [8], if there is a wide gap between the current singular value and the next one, then there is little chance to include the topics whose corresponding singular values are less than the current one.

In our work, we propose a comprehensive method that combines term description with sentence description for each topic. We endeavor to select a set of sentences that not only have the best representation of the topic but also include the terms that can best represent this topic. Also, in order to utilize the mutual reinforcement between neighbor sentences, we put forward a new way to create the term by sentence matrix.

This paper is organized as follows: in Section 2, we introduce LSA briefly. Section 3 progresses to present our method in detail. In Section 4, the effectiveness of our method is confirmed by experimental results. Finally, we conclude this paper in Section 5.

2 Latent Semantic Analysis

LSA uses Singular Value Decomposition (SVD) to find out the semantic meaning of sentences. The SVD of a matrix A with the dimension of $m \times n$ $(m>n)$ can be defined as: $A = U \Sigma V^T$, where $U = [u_1, u_2, \cdots, u_n]$ is an $m \times n$ column-orthogonal matrix whose left singular vector u_i is an m-dimensional column vector, $V = [v_1, v_2, \cdots, v_n]$ is an $n \times n$ column-orthogonal matrix whose right singular vector v_j is an n-dimensional column vector. $\Sigma = diag\ (\sigma_{1,} \sigma_{2,} \cdots, \sigma_n)$ is an $n \times n$ diagonal matrix whose diagonal elements are non-negative singular values sorted in descending order.

From semantic perspective, we assume that SVD generates the concept dimension [11]. Each triplet (left singular vector and right singular vector) can be viewed as representing such a concept, the magnitude of its singular value represents the degree of importance of this concept.

3 Text Summarization Based on Latent Semantic Analysis

3.1 Document Analysis

This step contains two tasks: Document Representation and Singular Value Decomposition. First, each document needs to be represented by a matrix. The matrix is constructed by terms (words with stop words eliminated) that occurred in the document representing rows and sentences of the document representing columns, thus it is called term by sentence matrix. For a text with m terms and n sentences where without loss of generality $m>n$, it can be represented by $A = [a_{ij}]_{m \times n}$. The cell a_{ij} can be filled out with different approaches. We will elaborate on the weighting schemes in section 4.

Once the term by sentence matrix is constructed, SVD will be employed to break it into three parts: U, Σ and V^T. Based on the discussion in section 2, we take U as term by concept matrix, V^T as concept by sentence matrix while the magnitude of singular values in Σ suggests the degree of importance of the concepts.

3.2 Sentence Selection

As with [6], a concept can be represented by the sentence that has the largest index value in the corresponding right singular vector, we make another hypothesis: a concept can also be represented by a few of terms, and these terms should have the largest index values in the corresponding left singular vector. The two forms of description of a concept are called sentence description and term description. Here each concept is treated as an independent topic.

Since sentences are composed of terms, it is hoped that the most representative sentences of the current concept should include the terms that best represent this concept. Therefore, each topic in the summary can be reconstructed by selecting sentences according to the magnitude of the index values in the right singular vector until a few of most representative terms that have the largest index values in the left singular vector are fully included.

The process of selecting summary sentences can be illustrated as follows.

- **Formulation.** For a document D with m terms and n sentences, suppose $term_i$ ($1 \leq i \leq m$) denotes the *i-th* term, and $sent_j$ ($1 \leq j \leq n$) denotes the *j-th* sentence, then $D=\{sent_1, sent_2, \ldots, sent_n\}$. M is the maximum number of sentences to be selected, k is the number of concepts that can be selected and N_k is the number of sentences for the *k-th* concept, k and N_k are initialized to 1 and 0 respectively. Let set S contain the summary sentences and initialize S to null.
- **Sentence Selection and Term Selection.** While $|S|<M$, for the *k-th* concept, select the sentence that has the largest index value from the *k-th* right singular vector v_k . Get l that l satisfies v_{kl}=Argmax(v_{ki}), include the *l-th* sentence $sent_l$ into S and delete the *l-th* element v_{il} for v_i ($1 \leq i \leq n$), update V^T and increase N_k. Then select three terms u_{kp}, u_{kq}, u_{ks} that are represented by the *Top3* largest index values from the *k-th* left singular vector u_k , and let set $T=\{term_p, term_q, term_s\}$.
- **Combination.** Delete terms that appear both in T and $sent_l$ from T. While T is not null, if $N_k<3$ and $|S|<M$, continue to select sentences for this concept, update V^T and T, increase N_k, else set T to null. Then increase k and begin to select sentences for the next concept.

Based on the above discussion, we give the formal description of our Sentence Selection method in Algorithm 1.

Algorithm 1. Sentence Selection based on LSA

```
Input: Document $D$, Matrix $U$, Matrix $V^T$, $M$
Output: Set $S$
Initialize $S=\phi$, $k=1$
    while $|S|<M$
        get $l$ in $v_k$, $S=S \cup \{ sent_l \}$, update $V^T$, $N_k=1$
        get $p, q, s$ in $u_k$, $T=\{ term_p, term_q, term_s \}$
        $T_0=T \cap sent_l$, $T=T-T_0$
        while ($T \neq \phi$)
            if ($N_k<3$ and $|S|<M$)
                get $l$ in $v_k$, $S=S \cup \{ sent_l \}$, update $V^T$, $N_k = N_k + 1$
                $T_0=T \cap sent_l$, $T=T-T_0$
            else $T=\phi$
        end while
        $k=k+1$
    end while
Return $S$
```

4 Experiments and Evaluation

4.1 Weighting Schemes

In order to elaborate on the weighting schemes, we define:

$$a_{ij} = L(t_{ij}) * G(t_{ij}) + N(t_{ij}) \,, \tag{1}$$

where $L(t_{ij})$ is the Local Weight for $term_i$ in $sent_j$, $G(t_{ij})$ is the Global Weight for $term_i$ in the whole document, $N(t_{ij})$ is the Neighbor Weight of $term_i$ in $sent_j$.

In the following, we use tf_{ij} denotes the number of times that $term_i$ occurs in $sent_j$, tf_{max} denotes the frequency of the most frequently occurring term in $sent_j$, n is the total number of sentences, n_i is the number of sentences that contain $term_i$, gf_i is the number of times that $term_i$ occurs in the whole document..

For Local Weight, we choose to use the following four alternative strategies:

- **Binary Representation** (BR): If $term_i$ appears in $sent_j$, $L(t_{ij}) = 1$, otherwise 0.
- **Term Frequency** (TF): $L(t_{ij}) = tf_{ij}$.
- **Augment weight** (AW): $L(t_{ij}) = 0.5 + 0.5 * (tf_{ij} / tf_{\max})$.
- **Logarithm Weight** (LW): $L(t_{ij}) = \log(1 + tf_{ij})$.

For Global Weight, possible weighting schemes can be:

- **No Global Weight** (NG): $G(t_{ij}) = 1$.
- **Inverse Sentence Frequency** (ISF): $G(t_{ij}) = 1 + \log(n / n_i)$.

- **Entropy Frequency** (EF): $G(t_{ij}) = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$, where $p_{ij} = \frac{tf_{ij}}{gf_i}$.

In order to make use of terms that occur in the neighbor sentences, we put forward the concept of Neighbor Weight and define Neighbor Weight as $N(t_{ij}) = \lambda[L(t_{i,j-1}) * G(t_{i,j-1}) + L(t_{i,j+1}) * G(t_{i,j+1})]$, where λ is a parameter which we will explore in the following experiments. So in the weighting schemes, we may add Neighbor Weight (AN) or just let Neighbor weight equals to 0 (NN).

Neighbor Weight is added mainly by the following three notable considerations: (1) Neighbor sentences can be affected by each other thus form clusters to make the topics more convince. (2) It helps to resolve anaphora resolution, since most of the time a pronoun and what it demonstrates appear in the adjacent sentences. (3) With neighbor weight added, it helps to resolve the issue of data sparsity.

4.2 Datasets and Evaluation Methods

The datasets that are used for the evaluation of our LSA-based summarization approach are DUC2002 dataset and DUC2004 dataset[1]. DUC2002 dataset contains 567 documents, each document is provided with two 100-word human summaries. The dataset of DUC2004 includes 5 tasks, while in our work, we only use task 2. In this task, documents are clustered into 50 topics of 10 documents each.

Two kinds of metrics that F score and ROUGE toolkit [12] are adopted.

$$P = \frac{S_{cand} \cap S_{ref}}{S_{cand}}, R = \frac{S_{cand} \cap S_{ref}}{S_{ref}}, F = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{2}$$

$$ROUGE - N = \frac{\sum_{S \in S_{ref}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in S_{ref}} \sum_{gram_n \in S} Count(gram_n)}, \tag{3}$$

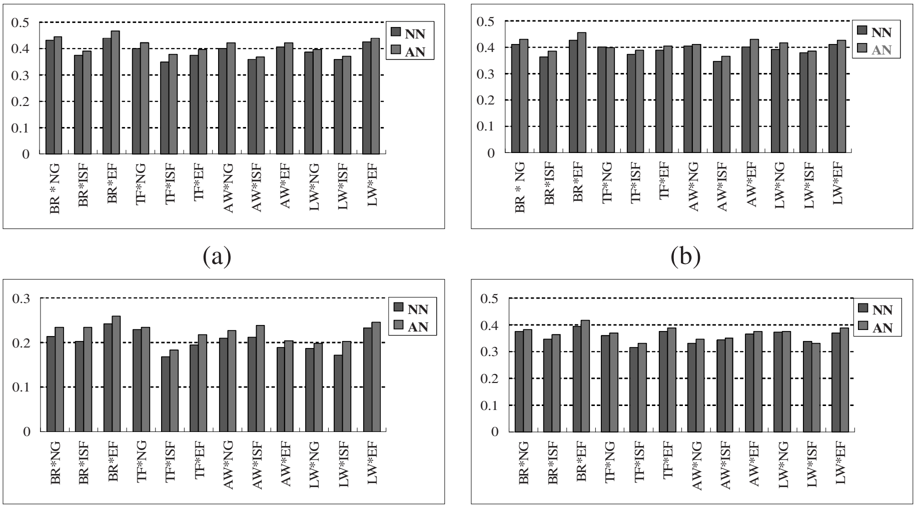
where S_{cand} denotes the candidate summary and S_{ref} denotes the reference summary, n stands for the length of the n-gram, $Count(gram_n)$ is the number of n-grams in the reference summaries, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and the reference summaries.

In our experiments, Longest Common Subsequence ROUGE-L together with ROUGE-SU4 [12] are also being used.

4.3 Experimental Results and Analysis

First, in order to compare the different weighting schemes we conduct experiments on DUC2002 dataset. We set λ in the Neighbor Weight to 0.5 initially.

[1] `http://www-nlpir.nist.gov/projects/duc/data.html`

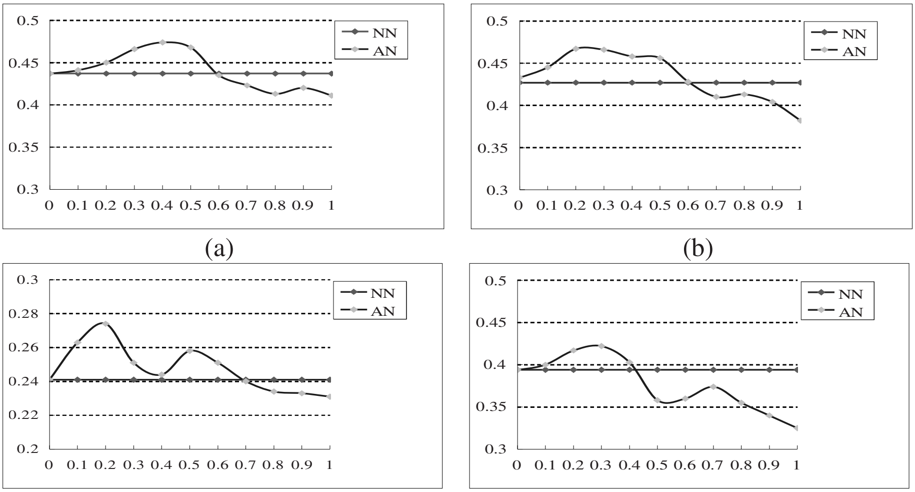


(a) (b)

(c) (d)

Fig. 1. Comparison of different weighting schemes (Local Weight*Global Weight +Neighbor Weight) for (a) F-1 score, (b) ROUGE-1, (c) ROUGE-2 and (d) ROUGE-L

From Figure 1 we can tell: the best combination of Local Weight and Global Weight is BR*EF, it performs better than other combinations at large. With Neighbor Weight added, nearly the results of all combinations acquire an improvement.



(a) (b)

(c) (d)

Fig. 2. Relationship between λ and (a) F-1 score, (b) ROUGE-1, (c) ROUGE-2 and (d) ROUGE-L

We apply the weighting scheme of BR*EF+AN in our experiments to show the impact that λ makes on the performance and get Figure 2.

In Figure 2, x-axis denotes the range of λ from 0 to 1, y-axis denotes the corresponding metrics value. From this figure, we can tell: with the Neighbor Weight added, the corresponding metric value increases firstly and then decreases with the raise of λ. Generally it is beneficial for λ in a small interval between 0 and 0.5. In order to get the most satisfying performance, we assign 0.25 to λ to make a compromise.

In the following, we take the weighting scheme of BR*EF+AN, and set the parameter λ in the Neighbor Weight to 0.25 to conduct experiments on DUC2002 dataset and DUC2004 dataset. Four LSA-based methods: GLLSA [6], SJLSA [7], MRCLSA [8] and OCALSA [9] together with three other latest models: DSDR-non [13], SATS [14] and MCMR [15] are adopted for comparison with our method. The ROUGE metrics of ROUGE-1(R-1), ROUGE-2 (R-2), ROUGE-SU4 (SU4) and ROUGE-L (R-L) are used for evaluation.

Table 1 shows different ROUGE scores on DUC2002 dataset and DUC2004 dataset. It can be observed that our LSA-based method achieves higher ROUGE scores and outperforms the other ones. As seen from this table, on DUC2002 dataset, ROUGE-1 score of our method is close to DUC-best, the scores of other three metrics are competitive with the DUC-best. On DUC2004 dataset, the scores of ROUGE-1, ROUGE-2 and ROUGE-SU4 of our method are higher than the DUC best. More importantly, our approach, nearly in terms of all ROUGE scores, outperforms the other methods that are based on LSA and is better than the three other latest modes.

Table 1. ROUGE results on datasets of DUC2002 and DUC2004

Algorithm	DUC2002 dataset				DUC2004 dataset			
	R-1	R-2	SU4	R-L	R-1	R-2	SU4	R-L
Baseline	0.411	0.211	0.166	0.375	0.221	0.064	0.102	0.117
DUC-best	**0.498**	**0.252**	**0.284**	**0.468**	**0.382**	**0.092**	**0.132**	**0.387**
GLLSA	0.432	0.174	0.137	0.352	0.341	0.065	0.120	0.350
SJLSA	0.410	0.207	0.158	0.382	0.356	0.064	0.138	0.347
MRCLSA	0.408	0.205	0.161	0.371	0.364	0.055	0.119	0.327
OCALSA	0.358	0.179	0.144	0.331	0.205	0.045	0.100	0.337
DSDR-non	0.466	0.267	0.138	0.352	0.385	0.098	0.118	0.329
SATS	0.448	0.209	0.164	0.374	0.295	0.076	0.124	0.354
MCMR	0.422	0.240	0.165	0.395	0.391	0.072	0.126	0.348
Ours	**0.472**	**0.261**	**0.170**	**0.423**	**0.384**	**0.097**	**0.137**	**0.355**

5 Conclusion

In this paper, we propose an improved LSA-based summarization algorithm that combines term description with sentence description for each topic. We select three sentences at most for each topic and the sentences selected not only have the best representation of the topic but also include the terms that can best represent this topic. We also put forward the concept of Neighbor Weight and propose a novel way that tries to utilize the mutual reinforcement between neighbor sentences to create the term by sentence matrix. Experimental results prove that our method achieve higher ROUGE scores than several well known methods.

References

1. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Computational Linguistics-Summarization 28(4), 399–408 (2002)
2. Vodolazova, T.: The role of statistical and semantic features in single-document extractive summarization. Artificial Intelligence Research 2(3), 35–44 (2013)
3. Gupta, V., Lehal, G.S.: A survey of Text Summarization Extractive Techniques. Journal of Emerging Technologies in Web Intelligence 2(3) (2010)
4. Das, D., Martins, A.: A Survey on Automatic Text Summarization. In: Literature Survey for the Language and Statistics II Course at CMU (2007)
5. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society for Information Science and Technology, 391–407 (1990)
6. Gong, Y.H., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 19–25. ACM, New York (2002)
7. Steinberger, J., Ježek, K.: Text Summarization and Singular Value Decomposition. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 245–254. Springer, Heidelberg (2004)
8. Murray, S.: Renals, and J. Carletta. Extractive Summarization of Meeting Recordings. In: Proceedings of the 9th European Conference on Speech Communication and Technology, pp. 593–596 (2005)
9. Ozsoy, M.G., Clicekli, I., Alpaslan, F.N.: Text summarization of Turkish Texts using Latent semantic analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, pp. 869–876 (2010)
10. Ai, D., Zheng, Y., Zhang, D.: Automatic text summarization based on latent semantic indexing. Artif. Life Robotics 15, 25–29 (2010)
11. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. SIAM Review 37(4), 575–595 (1995)
12. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Proceedings of the ACL Text Summarization Workshop, pp. 74–81 (2004)
13. He, Z., Chen, C., Bu, J., Wang, C., Zhang, L.: Document Summarization Based on Data Reconstruction. In: Proceeding of the Twenty-Sixth AAAI Conference on Artificial Intelligence, pp. 620–626 (2012)
14. Chandra, M., Gupta, V., Paul, S.K.: A statistical approach for Automatic Text Summarization by Extraction. In: 2011 International Conference on Communication Systems and Network Technologies, pp. 268–271 (2011)
15. Alguliev, R.M., Aliguliyev, R.M., Hajirahimova, M.S., Mehdiyev, C.A.: MCMR: Maximum coverage and minimum redundant text summarization model. Expert Systems with Applications 38(12), 514–522 (2011)