Research on Automatic Text Summarization Based on Latent Semantic Indexing

Dongmei Ai[1,2] , Yuchao Zhang[2] , Dezheng Zhang[2]

1 (School of Applied Science , University of Science and Technology Beijing, China)
2(School of Information Engineering, University of Science and Technology Beijing, China)

(Tel : 86-01-62332349; Fax : 86-01-62334071)
(aidongmei@sina.com)

Abstract: Automatic summarization is a common–concerned topic of computational linguistics and information science, the computer system of text summarization is considered as an effective means of processing information resources. In this paper, we propose an method of text summarization based on latent semantic indexing, which uses semantic indexing to calculate the sentence similarity. It improves the accuracy of sentence similarity calculation and subject delineation, and helps the generated abstracts universally cover the documents as well as reducing its redundancy. The effectiveness of the method is proved in the experimental results , compared to the traditional VSM-based method of automatic text summarization, the quality of generating abstracts was significantly improved.

Keywords : Latent Semantic Indexing ;vector space model; analysis of text structure; automatic text summary

I. INTRODUCTION

With the development of computer and information technology, especially the widely use of Internet, the amount of information available to the people increases rapidly. Facing such abundant information resources, it becomes more and more important for people to obtain useful information from them. Automatic Summarization is an effective tool to solve the above problem. It can significantly accelerate the speed of information filtering, and help readers get a basic understanding of the contents in related documents and find their necessary materials quickly and accurately [1].

In the traditional keyword-based Vector Space Model (VSM) [2], a document vector composed of m keywords $Di = \{d1, d2, \ldots\ldots, dm\}$ represents a document in the document set and it is the basis of text processing. The unstructured text can be represented in a form of vector, which makes a variety of mathematical processing be possible [3][4]. Its main advantage lies in that the process logic is simple and quick. However, VSM assumes that the relationship between words is independent (orthogonal assumption), which is difficult to be satisfied in the real environment. Generally, the words that appear in the text have a certain degree of correlation, which will impact the calculation results to some extent. Meanwhile, such kind of keyword-based text processing method is mainly based on word frequency information. The similarity between two texts is determined by the number of common words between them, thus making it hard to distinguish the semantic ambiguity of natural language. There exist a large number of synonyms and polysemant in the natural languages. The accurate representation of semantics not only depends on the proper use of vocabulary itself, but also on the contextual constraints on the meaning of the words. If ignoring the contextual constraints and only using isolated keywords to represent the content, it will surely impact the accuracy and completeness of text processing.

Latent Semantic Indexing (LSI) is a method used to achieve automatic knowledge extraction and representation. By performing statistical analysis of a large text set, it can extract the meaning of a word in the context. Technically, LSI is similar to VSM, as both of them use space vectors to represent texts. However, by using SVD decomposition to eliminate the impact of synonyms and polysemant, LSI improves the accuracy of post-processing. The basis of LSI is that, there exists a certain link between the words in the text, i.e., latent semantic structure. Such latent semantic structure is implicit in the pattern of word usage in the context. Therefore, when using statistical calculation method to analyze a large amount of texts to find such kind of latent semantic structure, we don't need deterministic semantic encoding. Simply by using the links of things in the context and representing words and texts with semantic

Supported by the National High-Tech Research and Development Plan of China under Grant No.2009AA01Z119 ; the National Natural Science Foundation of China under Grant No.60973065.

structure, we can eliminate the correlation between words and simplify the text vector.

In view of the above reasons, this paper will focus on the basic idea and features of LSI method, and propose LSI-based automatic text summarization method.

II. LSI Model

In the field of text processing, unstructured text can be formalized with the VSM model, thus making all kinds of mathematical processing be possible. However, the orthogonal assumption of mutual independence between words in the VSM model has obviously ignored the correlation between words in the natural languages. The text vector space composed of raw words will inevitably lead to the expansion of high-dimensional space and the inundation of text features.

To solve this problem, S. Deerwester proposed Latent Semantic Indexing (LSI) model [5] in 1990. Then, this model was widely used in many fields, such as information retrieval, text clustering, text filtering and others, and also has been improved continuously [6-9]. In this paper, we propose a method of text summarization based on LSI model.

LSI model first assumes that there exists certain latent semantic structure in the text, and then uses statistical methods to estimate such latent structure and project the text from the observed high-dimensional surface space (composed of raw words) to the low-dimensional latent semantic space (composed of concepts). In the way, it eliminates the impact of orthogonal independence assumption in the VSM, and determines the position of the text in the vector space more properly. In addition, while reducing the dimensionality of the text vector space, it still retains the semantic information of the original text to a large extent.

LSI uses the matrix Single Value Decomposition (SVD) method [10] in Linear Algebra to estimate the latent text vector matrix $\hat{X}$ from the text vector matrix X composed of original "word - document".

Mathematically, any t×d matrix with rank m can be decomposed as follows:

$$X = T_0 S_0 D_0^T \qquad (1)$$

In the above expression, T_0 and D_0^T are called the left singular vector and right singular vector respectively, which have orthogonal columns

$T_0^T T_0 = I, D_0^T D_0 = I$; S_0 is a diagonal matrix. $S_0 = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$, $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_m \geq 0$。

By choosing proper value for k, we can satisfy the following inequality:

$$\sum_{i=1}^{k} \lambda_i \Big/ \sum_{j=1}^{m} \lambda_j \geq \theta \qquad (2)$$

In the inequality, θ is the threshold that contains the original information. By deleting the relevant rows and columns in S_0, we can obtain S; by deleting the relevant rows and columns in T_0 and D_0, we can have T and D. Then we perform multiplication operations over three matrixes, and obtain a new matrix $\hat{X}$. Using Similarity matrix $\hat{X}$ to replace the original matrix X, the new matrix with rank k is closest to the original matrix in the sense of least squares.

$$X \approx \hat{X} = TSD^T \qquad (3)$$

Fig.1 is the illustration of the singular value decomposition of the "word - document" matrix. The new "word - text" similarity matrix $\hat{X}$ obtained by SVD has the following two features compared with the original matrix X: (1) $\hat{X}$ ignores the smaller singular value, which is equivalent to the exclusion of the noise in the original matrix X. (2) Since the element in the original matrix X is the word's eigenvalue in the document and there exists sparse data problem in the matrix, the removal of items with smaller singular values is equivalent to smoothing X.

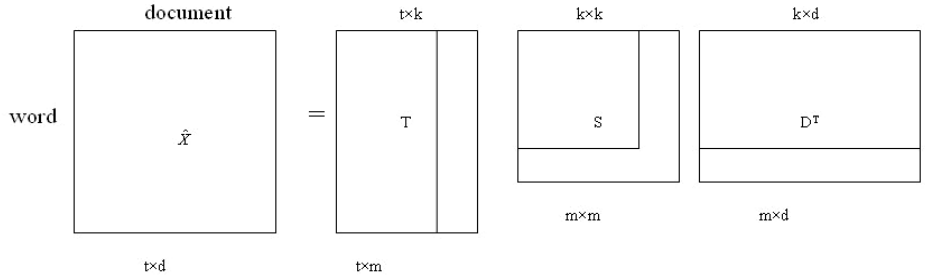

Fig.1. The illustration of SVD process

III. LSI-based Text Structure Analysis Method

The goal of text structure analysis is to automatically identify the boundary of paragraphs with independent meaning (meaning paragraphs) [11-12]. Generally, the meaning paragraph is greater than or equal to the natural paragraph and there is no clear boundary of meaning paragraphs as that of natural paragraphs. They are formed during the description of the article topic.

Note that, in fact, a meaning paragraph is composed of several successive text units (natural paragraphs or sentences). To support the ideas expressed in a meaning paragraph, the natural paragraphs (or sentences) contained in the same meaning paragraph often have great similarity in the choice of vocabulary as well as the term frequency. At the same time, considering that there exist "transform" sentences in the natural paragraphs of the article, we should select "sentences" as the text units to split the article from the perspective of subject delineation. Namely, we can process "sentences" as the above described "document" and construct the "word - sentence" matrix.

In view of these facts, we propose a LSI-based text structure analysis method in this section. By exploiting the similarity of adjacent text units in the LSI measure, we can determine the interval points with the lowest compact degree and thus complete the delineation of the document structure.

1. Calculation of Various Similarities

By using SVD, we can obtain the best approximation $\hat{X}$ of the "word - sentence" matrix of all the documents. With $\hat{X}$, it is easy to obtain the semantic relationship between words and sentences. In the latent semantic analysis, we mainly discuss three relationships, i.e., the relationship between words and sentences, the relationship between words and words, the relationship between sentences and sentences. Next, we will introduce the calculation method of these three relationships:

1) Distance between words and sentences

The similarity between sentences and words is $\hat{X}$ itself. The i-th row and the j-th column in $\hat{X}$ indicate the similarity degree between word i and sentence j.

2) Distance between words and words

When calculating the similarity degree between words, we need to perform "forward" multiplication on $\hat{X}$.

$\hat{X} * \hat{X}^T = T * S * D^T * D * S^T * T^T$,

$\because S$ is a diagonal matrix, $\therefore S = S^T$,

$\because D$ is an orthogonal matrix, $\therefore D * D^T = I$

Thus,

$$\hat{X} * \hat{X}^T = T * S * I * S * T^T = T * S^2 * T^T \qquad (4)$$

The i-th row and the j-th column in the matrix $\hat{X} * \hat{X}^T$ indicates the similarity between word i and word j.

3) Distance between sentences and sentences.

To calculate the similarity degree between sentences, we need to perform "reverse" multiplication on $\hat{X}$.

$\hat{X}^T * \hat{X} = D * S^T * T^T * T * S * D^T$

Same as above, $S = S^T, T * T^T = I$

Thus:

$$\hat{X}^T * \hat{X} = D * S^T * I * S * D^T = D * S^2 * D^T \qquad (5)$$

The i-th row and the j-th column in the matrix $\hat{X}^T * \hat{X}$ indicates the similarity between sentence i and sentence j.

2.Calculation of Compact Degree

Suppose that the text D contains n sentences, k meaning paragraphs. Let H be the text meaning paragraph and s be the sentence. We have the following relationship:

$D = \{H_1 H_2 \cdots H_k\} = \{s_{i_1} \cdots s_{i_2 - 1}\}\{s_{i_2} \cdots s_{i_3 - 1}\} \cdots \{s_{i_k} \cdots s_{i_{k+1} - 1}\}$,in which $i_1 = 1 \le i_2 \le \cdots \le i_k \le i_{k+1} - 1 = n$. To facilitate the writing, the notations are simplified as s_i 、 s_{i+1} .

If taking the similarity degree between adjacent sentences in the text (i.e., $sim(s_i, s_{i+1})$) as the measure of compact degree, we can still reflect the consistency of the subject by using the similarity between adjacent sentences. Obviously, the splitting points of the text structure are naturally the interval points with a low similarity degree, which implies poor continuity in the context and can be taken as the candidates of delineation. Therefore, the interval points

with low similarity are the objects that deserve more attention [14].

If regarding a text as a complete text vector space, we can use the sentences in the text as the basic processing unit and formalize it as a text vector. Assume that variable pair <s, w> means that the word w appears in

The sentences, $s \in S = \{s_1, s_2 \cdots, s_n\}$, $w \in W = \{w_1, w_2, \cdots, w_m\}$. We can use LSI model to construct the best approximation matrix $\hat{X}$ of the "word - sentence" matrix and calculate the similarity matrix between sentences and sentences (i.e., $\hat{X}^T * \hat{X}$). When extracting the related element in the similarity matrix, its value can be taken as the similarity degree between adjacent sentences (i.e., $sim(s_i, s_{i+1})$), as shown in Fig.2 A higher similarity degree means a higher compact degree in the context; otherwise, it means a lower compact degree in the context and implies the end of a subject.

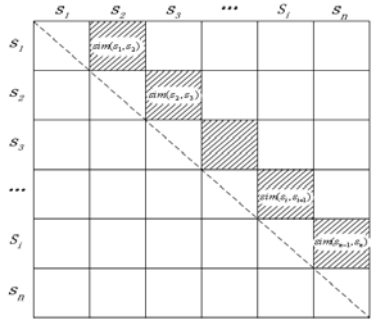

Fig.2. the illustration of the calculation of similarity degree between adjacent sentences.

3. Methods of Boundary Recognition

The objective of boundary recognition is to determine the candidate points of text delineation according to the compact degree between interval points. There are the following policies of boundary recognition [15]:

1) Threshold Method

Set a constant θ. If the similarity value between sentences satisfies $sim(s_i, s_j) < \theta$, we can think that s_i and s_{i+1} belong to different segments. The method is easy to implement and has a lower error ratio if choosing a proper θ.

2) Dynamic Constant Method

Although the threshold method is simple, the constant θ should be set manually and it is hard to provide the best value. Therefore, we can consider changing the value of θ dynamically according to the similarity degree between adjacent sentences. Suppose that the text to be split has n sentences, and then the similarity degree between adjacent sentences can be represented as:

$$SimTable = \{Sim_1, Sim_2, \cdots, Sim_i, \cdots, Sim_{n-1}\}$$

in which $Sim_i = sim(s_i, s_{i+1}), 1 \le i \le n-1$, Let,

$$avgSim = (Sim_1 + Sim_2 + \cdots + Sim_i + \cdots + Sim_{n-1})/(n-1) \quad (6)$$

$$avgmSim = ((Sim_2 - Sim_1) + \ldots + (Sim_{n-1} - Sim_{n-1}))/(n-2) \quad (7)$$

If $avgmSim \le sim(s_i, s_{i+1}) \le avgSim$, we can think that s_i、s_{i+1} belong to different pieces.

3) Local Minimum Method

Select the local minimum value $sim_{sim}(s_i, s_j)$ in the similarity table SimTable. Start from each local minimum value and move towards the left and the right to search the closet larger value Sim_l、Sim_r, and then use Equation (8) to calculate the relative depth:

$$d_{rel}(s_i, s_{i+1}) = (Sim_l + Sim_r)/(2 \times sim_{\min}(s_i, s_j)) - 1 \quad (8)$$

Set a threshold value α. If the relative depth $d_{rel}(s_i, s_{i+1}) > \alpha$, s_i、s_{i+1} belong to different paragraphs. Our paper proposes to use the local minimum method to find the interval points of splitting text structure.

To summarize, our proposed LSI-based text structure analysis algorithm can be described as follows:

Step1： Construct the "word-sentence" matrix in the original text;

Step2： Use LSI model to decompose the original matrix and obtain the best approximation matrix $\hat{X}$ of the "word-sentence" matrix;

Step3： Calculate the similarity degree matrix between sentences and sentences, i.e., $\hat{X}^T * \hat{X}$. Extract the corresponding elements in the similarity degree matrix. The value can be taken as the similarity degree between adjacent sentences $sim(s_i, s_{i+1})$;

Step4： Use the boundary recognition method to find the interval points of splitting text structure and achieve the goal of text structure analysis.

IV. GENERATING SUMMARY

1.Calculation of Sentence Weight

The sentence weight is calculated after determining the subject. The equation to calculate the j-th sentence in the subject $H(i)$ (i.e., S_{ij}) is as follows:

$$I(S_{ij}) = \lambda_h \times \lambda_{pos} \times \sum_{k=1}^{n} sim(S_{ij}, S_{ik}) \quad (9)$$

in which n is the number of sentences contained in the subject $H(i)$. It is the sum of similarity degree between this sentence and other sentences in the same subject and reflects the representation degree of the sentence for the belonging subject. The higher the value is, the more information of the subject the sentence contains, which implies a higher representation degree.

λ_h is the score determined by the number of prompting words contained in the sentence. Prompting words can be categorized into two types: ① Weight-added prompting words, which prompt that the sentence contains important contents, such as "in summary", "the main purpose of research", "this paper explores", "overall", "experiments show", etc. The sentences guided by such kinds of words should be given a positive weight to improve the sentence's

importance; ② Weight-reduced prompting words, which indicate that the sentence doesn't contain important substantive content, such as "for example", "for instance", "take an example", "in other words", etc. For the sentences started with such words, we should reduce their weight to decrease its importance. This paper adopts the following empirical values:

$$\hat{C}(s_i)=\begin{cases}0.5 & \text{Weight - added words}\\ 0 & \text{Others}\\ -0.5 & \text{Weight - reduced words}\end{cases} \quad (10)$$

λ_{pos} is the score determined by the location of the sentence. The sentences that appear in the first paragraph or the last paragraph have a higher score. In the application of automatic text summarization system, the location feature is still very effective heuristic information. Our paper provides extra weights for the sentences in the special locations, such as the beginning paragraph, the ending paragraph, the paragraph after each sub-heading, the sentence at the beginning of a paragraph, the sentence at the end of a paragraph, to exhibit their importance. The location weight λ_{pos} of sentence s_i is defined as:

$$\lambda_{pos}=l_s(s_i)\times l_p(s_i), \quad (11)$$

in which $l_s(s_i)$ is the weight tuning parameter of the location of s_i and $l_p(s_i)$ is the weight tuning parameter of the paragraph location of s_i.

Our proposed automatic text summarization method can use the following weight parameters, as shown in Table 1.

Table 1. Weight tuning parameters of paragraph location and sentence location

	First paragraph First sentence	First paragraph Other sentences	First paragraph Last sentence	Other paragraphs First sentence	Other paragraphs Other sentences	Other paragraphs Last sentence	Last paragraph First sentence	Last paragraph Other sentences	Last paragraph Last sentence
l_s	1.4	1.0	1.3	1.4	1.0	1.3	1.4	1.0	1.3
l_p	1.5	1.5	1.5	1.0	1.0	1.0	1.2	1.2	1.2

2. Extracting Sentences to Generate the Summary

In Summary, the LSI-based automatic text summarization algorithm proposed in this paper is as follows:

Step1: $N=[n(s,w)]_{m\times n}$; Construct the "word – sentence" matrix $N=[n(s,w)]_{m\times n}$ in the original text;

Step2:Use LSI model to decompose the original matrix and obtain the best approximation matrix $\hat{X}$ of the "word – sentence" matrix;

Step3: $sim(s_i,s_{i+1})$; Calculate the similarity degree matrix between sentences and sentences $\hat{X}^T * \hat{X}$. Extract the related elements in the similarity degree matrix and the value is taken as the similarity degree between adjacent sentences $sim(s_i,s_{i+1})$;

Step4: Use the boundary recognition method to find the interval points of splitting text structure and achieve the goal of text structure analysis and subject splitting.

Step5:Calculate the weight $I(S_{ij})$ of each sentence in the text;

Step6:First allocate the required summary length proportionally to each subject; then extract a comparable amount of summary sentences from each subject based on the ranking of sentence weights and generate the summary.

IV. EXPERIMENTAL RASULTS

To validate the effectiveness of our proposed Chinese automatic text summarization method, we build two Chinese automatic text summarization systems to conduct comparison experiments.

System I: the system uses our proposed automatic text summarization method;

System II: calculate the sentence similarity by constructing word-based VSM and then cluster the sentences with hierarchical clustering method. The sentence extraction method is the same as our proposed method.

Here, we use three measures – recall, precision, F-measure – to evaluate the text summarization system. Among them, recall refers to the ratio of correct recognition, and precision refers to the ratio of precise recognition. Their detailed expressions are as follows:

Recall = the number of sentences extracted by both the text summarization system and the expert / the number of sentences extracted by the expert

Precision = the number of sentences extracted by both the text summarization system and the expert / the number of sentences extracted by the text summarization system

To comprehensively evaluate the summary quality, we adopt a combined evaluation measure, i.e., F-measure.

$$F\_measure=\frac{2\times P\times R}{P+R}$$

For example, in a given document, the summary length accounts for 15% of the whole document. If the number of sentences extracted by the system is 8, the number of sentences extracted by the expert is 12 and the number of common sentences is 5, we have:

Recall =5/12=0.417

Precision = 5/8=0.625

$$F\_measure = (0.417 * 0.625 * 2)/(0.417 + 0.625) = 0.5$$

In this paper, we randomly select 50 articles from the corpus. We first obtain their standard summary by asking the experts to manually label them, and then we use System I and System II to conduct tests on them. We extract the summary with a percentage of 10%、20%、25%、30% respectively and get the average precision, the average recall, the average F-measure, as shown in Table 2.

It can be observed that, System I, which is built based on our method, has a higher value of recall, precision and F-measure than that of System II under different percentages of summary length. It implies that the summary generated by our method well balances the requirements of coverage and precision, and has a better quality.

Table 2.Comparison of precision, recall, F-measure of different systems (%)

Summary Percentage	Average Precision		Average Recall		Average F-measure	
	System I	System II	System I	System II	System I	System II
10%	41.81	41.40	22.94	21.33	27.50	25.60
20%	60.31	58.96	32.98	26.41	40.79	35.40
25%	61.56	58.49	40.96	29.91	46.87	37.61
30%	62.14	60.59	50.58	38.54	53.25	45.38

V. CONCLUSION

This paper proposed a LSI-based text summarization method. It exploits LSI to obtain the semantic structure of sentences and calculate the sentence similarity in the semantic space, and thus avoids the "bias" phenomenon caused by the word-based VSM. The experimental results show that the summary generated by our proposed method has better quality than that generated by the traditional VSM-based approach, and thus validate the effectiveness of our method.

The problems observed in the current experiments lie in the process of referencing relationship. Due to the use of statistical methods, it is hard to determine the referencing relationship in the context. It will affect the coherence and understandability of the sentences. This is also what we should improve in the next step when using semantic analysis method.

REFERENCES

[1] Tianshun Yao, Qianbo Zhu, Li Zhang (2002), Understanding Natural Languages Second Edition. Beijing: Tsinghua University Press

[2] Salton G, MeGill M J (1983), Introduction to modern Information Retrieval .New York: McGraw-Hill Book Company

[3] Eric W Brown, James P Callan, W Bruce Croft(1994), Fast Incremental ndexing for Fu11-Text Information Retrieval[C]. In: Proceedings of the 20th VLDB Conference Santiago, Chile:1-2

[4] Clifford A Lynch (1995), Networked Information Resource Discovery: An Overview of Current Issues. IEEE[J] .TOURNAL ON SELECTED AREAS IN COMMUNICATIONS,13 (8):1505-1522

[5] S. Deerwester, S. Dumais, G. Furnas, T (1990), Landauer and R. Harshman Indexing by Latent Semantic Analysis[J]. Journal of the American Society for Information Science, 41(6):391-407

[6] Chengqing Zong (2008), Statistical Natural Language Processing [M]. Beijing: Tsinghua University Press

[7] Jiantao Sun(2005), Dimension Reduction and Classification Method Research in Web Mining [D]. Beijing: Tsinghua University

[8] Qiaozhu Mei, ChengXiang Zhai(2006), A Mixture Model for Contextual Text Mining[C]. In Proceedings of the 2006 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'06):649-655

[9] Yihong Gong, Xin Liu (2001), Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis[C]. In Proceedings SIGIR2001:19-25

[10] Jen-Yuan Yeh, Hao-Ren Ke, and Wei-Pang Yang, (2002), Chinese Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis[C]. Digital Libraries: People, Knowledge, and Technology: 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Proceedings: 76-87

[11] Hongfei Lin(2000), Text Structure Analysis Method Based on Concept [J]. Computer Research and Development, 37(3):324-328

[12] Xiongguan Wei (1997), Diagram-based Text Analysis Method [J]. Pattern Recognition and Artificial Intelligence, 10(2):112-117

[13] Zechner K, Lavie A (2001), Increasing the Coherence of Spoken Dialogue Summaries by Cross-Speaker Information Linking[C]. In Proceedings of the NAACL-01 Workshop on Automatic Summarization. Pittsburgh, PA: Association for Computational Linguistics:22-31

[14] Hearst, Marti A(1997), TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages[J]. Computational Linguistics:33-64

[15] Jing Shi, Guozhong Dai(2007). Text Segmentation Based on PLSA Model [J]. Computer Research and Development, 44(2):242-248