Improvement of Text Segmentation TextTiling Algorithm

Xin He¹, Jian Wang¹, Quan Zhang², Xiaoming Ju²

¹State Grid Zhejiang Electric Power Company Information and Communication Branch, HangZhou, 310007
²East China Normal University, School Computer Science and Software Engineering, shanghai, 200062

Author: Xin He; email: xinhe@163.com; phone: 13588095593
Author: Jian Wang; email: 12928004@qq.com; phone: 13757141976
Author: Quan Zhang; email: 1172935212@qq.com; phone: 18852951520
Author: Xiaoming Ju; email: xmju@sei.ecnu.edu.cn; phone: 13774272718

Abstract: Text segmentation has a wide range of applications in the fields of information extraction and abstract generation. The method uses local information to determine a specific segmentation, and cannot fully consider the information of the remaining paragraphs. In response to this problem, this paper improves the segmentation point selection of the algorithm. After the similarity values of the respective interval points are obtained by the similarity calculation, the sliding window is used to slide on the interval points, and the similarity value of the interval points at the center of the window is replaced with the average of all the similarity values in the window. The experimental results show that the text segmentation based on the improved method can suppress small local changes on the curve, thus highlighting the large similarity value changes, and effectively improving the performance of text segmentation.

1. Introduction

With the rapid development of the network, human beings have gradually entered a new era of network, and the growth of information resources has grown at an explosive rate. All kinds of massive text information bring convenience to human beings, but also bring great challenges to text processing and analysis. Text segmentation is one of the important steps to solve this problem. Text segmentation refers to the division of texts according to the principle of topic-related, so that each semantic paragraph has the greatest subject relevance within it, and there is minimal correlation between paragraphs [1], so as to find the boundaries of different topics.

Text segmentation has been widely used in the fields of information extraction and abstract generation [2-3]. For example, in information retrieval, text segmentation can help users narrow down the scope of document retrieval and accurately locate in a short time, and effectively achieve high information retrieval efficiency; In the summary generation, text segmentation can be based on dividing the text into multiple sub-themes, and using the representative sentences under each word theme to form the theme idea, so that it can cover the entire text information comprehensively.

At present, commonly used text segmentation methods include vocabulary aggregation based methods, language feature based methods, and topic model based methods. Vocabulary aggregation based methods suggest that words describing the same topic tend to appear within the same topic
In 1997, Hearst[4] proposed an algorithm. The main idea of this method is to divide the text into vocabulary sequences with fixed length based on vocabulary aggregation, and then calculate the cosine similarity between adjacent sequences according to the sliding window, and determine the boundary position accordingly. This algorithm is a classic text segmentation algorithm, and many scholars propose different improvement schemes for the problems existing in the algorithm. Zhang Yu et al. improved the research based on the user's new interest discovery [5]. Zhu Haijun and others introduced the vocabulary association calculation method for compactness calculation by introducing HowNet, and added some heuristic rules to improve the effect of text segmentation when selecting boundaries.

Partial information is used in Hearst's method to determine a specific segmentation, but often the differences in the various paragraphs in a longer article are large. The segmentation that determines whether the current position can be a semantic paragraph is often related to other chapters of the article. Therefore, this paper proposes an improved algorithm for the algorithm that cannot consider the rest of the paragraph. After the similarity values of the respective interval points are obtained by the similarity calculation, the sliding window is used to slide on the interval points, and the similarity value of the interval points at the center of the window is replaced with the average of all the similarity values in the window. Thereby highlighting the large similarity change and improving the segmentation accuracy.

2. Text Segmentation TextTiling Algorithm

2.1. Divide the unit length by text
First use the word segmentation tool to segment the text. The text data uses the news corpus, and the content-independent articles are hand-made. The average vocabulary in each article is about 3000, and the natural segment is more than 20. Manually determine the segmentation points of each article and mark them in the article. Such articles have prepared 200 articles on economics, military, sports, education, culture, entertainment, and politics. Count the total vocabulary in these articles. Use the Harbin Institute of Technology to stop the stop words in the article, and save the remaining words in the hash table to improve the retrieval speed. In Hearst's experiment, he showed that when 20 words were included in each sentence, the experimental results were better. In this paper, the results of the experimental vocabulary changes from 15 to 40, respectively, to analyze the effect of sentence length on the segmentation in the document.

2.2. Similarity calculation
The similarity calculation is an acquaintance calculation of the text on both sides of the interval. In order to divide text according to different subtopics, it is necessary to sequentially compare the similarities between adjacent two paragraphs. Since the length of each paragraph is different, the result of the similarity comparison will be different, so the paragraphs have to be re-divided before the comparison so that the two text blocks being compared have the same length. Calculating the similarity of each interval point is actually equivalent to sliding in the text with a window of length $2 \times \text{blocksize}$, where $\text{blocksize}$ is the size of the block, and the middle position of this window is the interval point. If the vocabulary on the left side of the interval is repeated multiple times on the right side, the subtopic continues to be discussed, and the interval is not a split point. If the vocabulary on the left side of the interval appears little on its right side or a large number of new words appear on its right side, the topic has changed here.

The similarity calculation formula is as follows:

$$sim(b_1, b_2) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_t w_{t,b_2}^2}}$$
Where $t$ represents all words that appear within the window and $w_{t,b_1}$ represents the weight of the word $t$ in the $b_1$ block. When calculating the similarity, if the number of sentences around the interval is not the same, then both sides are compared with the minimum number of sentences.

Figure 1 Similarity--interval curve

Figure 1 shows the similarity at each interval in the text. This is a text with more than 4,000 words. There are 10 natural segments in the article, each of which comes from a different theme. After division, the text is divided into 180 sentences, each containing 20 words. The curve of Fig. 1 is the similarity curve of each interval point. The local highest point on the curve indicates that the text similarity on both sides is the largest, and the local lowest point indicates that the text similarity on both sides is the smallest.

2.3. Boundary determination

After the similarity calculation results, the graph is drawn according to the similarity values between the paragraphs, and the degree of similarity between the paragraphs is observed from the curve. When the similarity is lower than a certain threshold, the relationship between the current paragraphs can be cut off, and the two natural segments will belong to different semantic paragraphs.

In the process of calculating the similarity, the curve has a large fluctuation. We can only observe the trend of the curve from it, but it can be seen that the curve is sharp and it is difficult to judge the segmentation position through the curve. By calculating the depth value $p$ of each interval point $D_p$, the depth value of the TextTiling algorithm is to measure the minimum depth value by looking at the highest similarity between the left and right sides, and the judgment formula:

$$D_p = \frac{1}{2}(h_l(p) - C_p + h_r(p) - C_p)$$

The function $h_l(p)$ returns the highest similarity to the left of the sequence gap index $l$, and $h_r(p)$ returns the highest similarity to the right, and then searches for the local maximum position based on the depth value. These obtained maximum scores are then sorted. If a paragraph is entered, the highest depth value is used as the basis for judging the boundaries of the semantic paragraph. Otherwise, if the depth value is greater than $\alpha - \beta/2$, then the boundary can be predicted, where $\alpha$ is the average depth value and $\beta$ is the standard deviation at the depth value.

2.4. Insufficient TextTiling algorithm

Local information is used in Hearst’s method to determine a specific segmentation, but often in a long article, the paragraphs vary greatly. Whether the current position can become a semantic paragraph is often related to other paragraphs of the article. Determining whether the current paragraph is a subtopic or topic will involve a hierarchy of current content, which is obviously related to the central idea and other content of the article, so it is somewhat lacking to make judgments based on local information.

3. Improved TextTiling Algorithm

After the boundary determination step, the similarity of each interval point is obtained by the similarity calculation, we introduce the curve smoothing technique. The smoothing process uses a sliding window to slide over the interval points, averaging the similarities of all the spacing points.
contained in the window, and replacing the similarity value of the spacing points at the center point of
the window with the average value. Calculated as follows:
\[ sim_i = \left( \frac{sim_{i-(k-1)/2} + \ldots + sim_i + \ldots + sim_{i+(k-1)/2}}{k} \right) \]

Where \( sim_i \) is the similarity value at the interval point and \( k \) is the window length. By smoothing,
local variations on the curve can be suppressed, thereby highlighting large similarity variations. Using
a smaller \( k \) value can retain more original information. When dealing with shorter articles, the content
of the article can provide more data for depth value selection, but at the same time it will generate
excessive segmentation points due to excessive noise. Using a larger \( k \) value can filter small subtopic
changes and preserve larger subtopic boundaries, so it is suitable for longer documents. The text
segmentation flow chart based on the improved TextTiling algorithm is shown in Figure 2:

Figure 3 is a smoothing result using a window length of \( k=5 \). Figure 4 is a smoothing result using a
window length of \( k = 10 \), from which it can be seen that as the value of \( k \) increases, the curve becomes
smoother.

4. Test Results And Discussions

4.1. Data sets
In view of the lack of public data sets for Chinese text segmentation, the experiments of Shijing [6]
and Zou Jian [7] were studied, and two data sets were used for experiments. We analyzed Sogou's
classified news and used it as an experimental test corpus. The corpus is mainly from Sohu News from
June to July 2012, domestic, international, sports, social, entertainment and other channels, a total of
17,820 news texts.
From the data set, 4,000 pieces of experimental corpus were selected as experimental test texts. According to the text segmentation data set constructed in the Choi [5] data set format, an article containing more than 4000 words was manually constructed, and the text contained 12 paragraphs. The divided text is divided into 158 sentences, each containing 20 words.

4.2. Evaluation

In order to facilitate the comparison of topic model algorithms, this paper selects two metrics commonly used for text segmentation tasks to evaluate the performance of each algorithm: The $P_k$ [8] measure and the WindowDiff (WD) [9] measure.

$P_k$ can be defined as:

$$P_k = P(\text{seg}) \times P(\text{miss}) + (1 - P(\text{seg})) \times P(\text{false alarm})$$

Where $P(\text{seg})$ denotes the probability that two sentences of distance $k$ belong to different subject fragments; On the contrary, $1 - P(\text{seg})$ indicates the probability that two sentences with distance $k$ belong to the same subject. In this experiment, $P(\text{seg}) = 0.5$ is set according to the setting in [10]. $P(\text{miss})$ is the probability that the result of the algorithm segmentation lacks a paragraph; $P(\text{false alarm})$ is the probability of adding a paragraph to the result of the algorithm segmentation.

WindowDiff can be defined as:

$$\text{WindowDiff}(\text{ref}, \text{hyp}) = \frac{1}{N - k} \sum_{i=1}^{N-k} |b(\text{ref}_{i}, \text{ref}_{i+k}) - b(\text{hyp}_{i}, \text{hyp}_{i+k})| > 0$$

Where ref represents the true segmentation of the document; hyp represents algorithm segmentation; $b(i, j)$ represents the number of boundaries between the entire sentence $\text{sentence}_i$ and the entire sentence $\text{sentence}_j$; $N$ represents the number of whole sentences in the text; $k$ takes half of the average length of the segments in the real segmentation.

4.3. Experimental result

The data is pre-processed prior to the experiment, using regular expressions to remove noise such as the scripting language retained by these news data. At the same time, removing those that are very frequent, but not helpful for segmentation of text, then segmentation of the dataset to obtain a segmentation of the word segmentation.

Table 1 Comparison of TextTiling and improved TextTiling segmentation evaluation indicators

| Method              | $K = 3$ | $K = 5$ | $K = 10$ | $K = 15$ |
|---------------------|---------|---------|----------|----------|
|                     | $P_k$   | $WD$    | $P_k$    | $WD$    |
| TextTiling          | 0.507   | 0.317   | 0.470    | 0.335    |
| Improved TextTiling | 0.453   | 0.100   | 0.425    | 0.046    |

The experimental results are shown in Table 1. From Table 1, the effect of different $k$ values on the segmentation effect can be seen. The segmentation result of the improved algorithm is obviously superior to the traditional algorithm. For the TextTiling algorithm without the smoothing process, no matter how the value of $k$ changes, the changes of the two indicators $P_k$ and $WD$ are small, resulting in more segmentation points. In contrast, the improved TextTiling algorithm, after the smoothing process is introduced, the $P_k$ and $WD$ indices change greatly, and the small local variation on the curve is suppressed by the smoothing process, thereby highlighting the large similarity change. Using a smaller $k$ value preserves more of the original information, but can result in excessive split points due to excessive noise. As shown in Table 2, $WD$ varies greatly at $k = 3$ and $k = 5$, indicating that $k = 3$ produces excessive split points.

5. Conclusion

The traditional TextTiling algorithm uses local information to determine a specific segmentation, but often in a long text, the paragraphs vary greatly, and the specific segmentation position is related to
other parts of the text. In response to this problem, this paper proposes an improved TextTiling algorithm, which introduces smoothing on the basis of the original. Firstly, the segmented text is preprocessed, and the text is converted into the corresponding word frequency matrix. Then, the similarity value of each interval point is obtained by the similarity calculation, and then the curve is smoothed to realize the text segmentation. The experimental results show that the curve smoothing process can suppress the small local variation on the curve, thus highlighting the large similarity change, which provides a strong basis for the selection of the segmentation point. Therefore, the method can reduce the error rate of text segmentation.

Acknowledgments
This work was financially supported by the State Grid Corporation of China Technology Project (5211XT17000F)

References
[1] Wu J W, Tseng J C R, Tsai W N. A hybrid linear text segmentation algorithm using hierarchical agglomerative clustering and discrete particle swarm optimization[J]. Integrated Computer Aided Engineering, 2014, 21(1):35-46.
[2] Joty S, Carenini G, Murray G, et al. Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails[C]// Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010.
[3] Paula C F C, Maite T, Thiago A S P. Subtopics annotation in a corpus of news texts: Steps towards automatic subtopic segmentation[c]// Proceedings of the 7th Brazilian symposium in Information and Human Language Technology. Fortaleza: Brazilian Computer Society, 2013:108-118
[4] Hearst M A. TextTiling: segmenting text into multi paragraph subtopic passages[J]. computational Linguistics, 1997, 23(1):33-64
[5] Choi F Y Y. Advances in domain independent linear text segmentation[J]. Proc. NAACL-00, 2000:26-33
[6] Jing S, Guozhong D. Text Segmentation Based on PLSA Model[J]. Journal of Computer Research & Development, 2007, 44(2):242-248.
[7] Jian Z, Mao-Sheng Z, Li M. Method of chinese text segmentation model acquisition and its optimization[J]. Journal of Nanchang University, 2001.
[8] Beeferman D, Berger A, Lafferty J. Statistical Models for Text Segmentation[M]. Kluwer Academic Publishers, 1999.
[9] Pevzner L, Hearst M A. A critique and improvement of an evaluation metric for text segmentation[M]. MIT Press, 2002, 28(1):19-36
[10] Tiancai L , Bo W , Yaoyi X , et al. Text Segmentation Based on Hierarchical Dirichlet Processes[J]. Journal of Data Acquisition and Processing, 2017.