Hierarchical Segmentation of Chinese Text Based on Knowledge Elements (Postprint)

Authors: Wang Zhongyi, Shen Xueying, Huang Jing

Date: 2023-07-26T00:00:00+00:00

Abstract

[Purpose/Significance] To help users retrieve complete knowledge units with appropriate granularity and satisfy their multi-granularity knowledge requirements. [Method/Process] This paper proposes a hierarchical text segmentation method based on knowledge elements. The method first analyzes the types of knowledge elements and their description rules; then identifies various types of knowledge elements in entity resources according to these rules, and treats all knowledge elements and the connecting sentences between them as a single class; finally, based on the Fisher segmentation algorithm, performs hierarchical binary division of this class until all topics are identified, determines segmentation boundaries, and achieves hierarchical text segmentation. [Results/Conclusions] The Chinese text hierarchical segmentation method based on knowledge elements, on the one hand, extends the text segmentation unit from sentences to knowledge elements, improving segmentation efficiency, and on the other hand, advances the control unit of knowledge services from documents to knowledge chunks composed of knowledge elements and knowledge element sets, providing users with relevant knowledge services on demand, thereby moving from data retrieval and information retrieval toward knowledge retrieval, improving knowledge acquisition efficiency, and realizing the transformation from information services to knowledge services.

Full Text

Hierarchical Segmentation of Chinese Text Based on Knowledge Elements

Wang Zhongyi[1], Shen Xueying[1], Huang Jing[2]
[1]School of Information Management, Central China Normal University, Wuhan 430079
[2]Wuhan Polytechnic, Wuhan 430074

Abstract

[**Purpose/Significance**] This study aims to help users retrieve complete knowledge units of appropriate granularity to satisfy multi-granular knowledge requirements. [**Method/Process**] We propose a hierarchical text segmentation method based on knowledge elements. This method first analyzes the types of knowledge elements and their description rules; then identifies various types of knowledge elements in entity resources according to these rules, treating all knowledge elements and their connecting sentences as a single class; finally, applies the Fisher segmentation algorithm to hierarchically bisect this class until all topics are identified, thereby determining segmentation boundaries and achieving hierarchical text segmentation. [**Result/Conclusion**] The proposed method extends segmentation units from sentences to knowledge elements, improving segmentation efficiency, while deepening the control unit of knowledge services from documents to knowledge blocks composed of knowledge elements and their sets. This advancement moves retrieval from data and information levels to the knowledge level, enhances knowledge acquisition efficiency, and facilitates the transformation from information services to knowledge services.

1. Introduction

With the rapid development of network technologies, online information resources have expanded exponentially, and the pace of work and life has accelerated. Traditional retrieval systems struggle to meet users' knowledge needs because they typically retrieve at the document level, returning entire documents that cause information overload. Users must spend considerable time and effort reading through documents to locate relevant knowledge, wasting valuable time. In reality, users primarily seek appropriately-sized knowledge blocks that satisfy their specific needs. To address this problem and achieve multi-granular knowledge services that delve into document internals, hierarchical multi-granular text segmentation is required. This paper proposes a hierarchical segmentation method based on knowledge elements to enable multi-granular knowledge services at the knowledge element and knowledge element set levels, allowing users to obtain precisely needed knowledge modules rather than entire documents during knowledge retrieval.

2. Research Status of Text Segmentation

Text segmentation refers to the automatic identification of boundaries between meaningful, independent knowledge units within a written document, with important applications in information retrieval and intelligent text processing [1]. Current research has achieved preliminary results, generally divided into two categories: linear segmentation and hierarchical segmentation. Linear segmentation divides text into consecutive fragments without considering internal structure, while hierarchical segmentation iteratively divides documents into finer-grained segments with hierarchical structure.

2.1 Linear Text Segmentation Current linear segmentation methods mainly fall into four categories: linguistic feature-based, lexical cohesion-based, topic model-based, and hybrid approaches combining two or more methods.

Linguistic feature-based methods extract lexical features from text to examine their relationship with topic boundaries. J.C. Reynar proposed a text segmentation algorithm using features individually or in combination to identify topic shifts across documents [2]. Zou Jian and Zhong Maosheng developed a Chinese-specific text segmentation model that calculates word relevance based on corpora and dictionaries to analyze sentence similarity [3]. However, this approach only works for specific texts with explicit formal information and lacks portability.

Lexical cohesion-based methods originate from Halliday and Hasan's work, which characterized lexical cohesion as word repetition, paraphrase, and semantic relationships [4]. H. Kozima proposed a lexical cohesion profile (LCP) method for linear text segmentation [5]. J.C. Reynar and M.A. Hearst developed the Dotplotting algorithm and TextTiling algorithm, respectively. Dotplotting relies entirely on word repetition to identify thematically similar regions and boundary points [6], while TextTiling calculates similarity between text units based on word repetition and word vector similarity [7]. F.Y.Y. Choi proposed the C99 algorithm, which builds on Dotplotting by constructing a similarity matrix from cosine similarities between all sentences, optimizing it through ranking to maximize internal density of segments [8]. J.M. Ponte and W.B. Croft used semantic relationships between words for local context analysis to find words and phrases related to each sentence [9]. Other notable methods include lexical chain-based segmentation, where J. Morris argued that lexical chain boundaries correspond to text structure [10]. These methods rely solely on textual information and may fail when sentences in a topic use synonyms without shared words, though semantically related words could indicate topic continuity.

Topic model-based methods address non-repetition issues by using semantic information and reducing word vector sparsity. F.Y.Y. Choi et al. proposed linear segmentation using latent semantic analysis (LSA) to estimate sentence similarity [11]. Shi Jing developed methods based on probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA), finding LDA more accurate [12-13]. M. Riedl and C. Biemann incorporated LDA into segmentation algorithms, showing significant performance improvements [14]. J. Eisenstein and R. Barzilay proposed a Bayesian unsupervised method that models each topic segment with a multinomial language model [15]. P. Mulbregt et al. applied hidden Markov models to text segmentation [16]. While effective, these methods typically require manual determination of topic numbers, which varies across datasets.

Hybrid methods combine multiple approaches for better results. T. Brants and F. Chen integrated PLSA with similarity-based boundary selection [17]. M. Riedl et al. proposed TopicTiling, combining TextTiling with LDA for more stable topic representation [18]. M.Y. Kan combined lexical cohesion features

with layout recognition elements [19].

2.2 Hierarchical Text Segmentation Although most documents have hierarchical structures, research on hierarchical segmentation remains limited. Y. Yaari proposed an unsupervised method using hierarchical agglomerative clustering on paragraphs, but this heuristic approach is fragile due to many manually-tuned parameters [20]. J. Eisenstein introduced a Bayesian framework using multi-scale lexical cohesion, though it doesn't extend to finer-grained segments like paragraphs due to sparse vocabulary [21]. Y.W. Teh et al. proposed the hierarchical Dirichlet process (HDP) model [22]. Li Tiancai and Wang Bo applied HDP with C99 for hierarchical segmentation, though errors remain high for short paragraphs [23].

In summary, while text segmentation research continuously improves, hierarchical segmentation studies are relatively scarce. Digital library collections exhibit hierarchical structures, requiring deeper investigation into hierarchical segmentation methods. Existing methods typically use sentences or paragraphs as minimal units, which cannot guarantee logically complete knowledge units—either being too fine-grained and breaking internal connections, or too coarse and blurring boundaries. Knowledge elements, as the smallest indivisible units with independent meaning, can effectively address these issues. This paper proposes hierarchical segmentation based on knowledge elements to advance digital library organization from coarse-grained documents to fine-grained knowledge elements.

3. Knowledge Element Identification

3.1 Description Rules for Knowledge Elements To achieve hierarchical segmentation based on knowledge elements, we must first identify them in documents. This paper employs a rule-based approach. Different knowledge element types have different description rules, requiring analysis of knowledge types.

Current scholars have varying classifications. Wen Youkui categorizes knowledge elements into descriptive types (information, noun explanation, numerical, problem description, citation) and procedural types (steps, methods, definitions, principles, experience) [24]. Zhang Jing classifies them into conceptual, principle, method, fact, and statement types for K-12 education [25]. Yuan Xiaoling divides them into theory/method, fact, and numerical types [26]. Zhao Rongying distinguishes between declarative (fact, definition, conclusion) and procedural (method, relationship) types [27].

Some classifications are overly detailed with overlapping categories (e.g., Wen's noun explanation and definition types share similar structures), while others are too coarse. This paper synthesizes previous work, dividing knowledge elements into descriptive and procedural types based on content expression. Descriptive types include concept, fact, and numerical knowledge elements; procedural types include method and relationship knowledge elements, as shown in [Figure 1: see

original paper].

We selected 650 journal papers from 13 disciplines (top 5 core journals per discipline, top 10 most-cited papers from each journal in recent five years) as training corpus. After document parsing and conversion to plain text, we extracted abstracts, keywords, and main bodies. Keywords and abstracts provided initial terminology. We then extracted knowledge element statements containing these terms, filtered domain words, obtained linear sentence patterns, and manually verified and categorized them by knowledge element type to generate description rules, as shown in [Figure 2: see original paper].

3.1.1 Descriptive Knowledge Element Rules (1) Concept Knowledge Element Rules. Concept knowledge elements are abstract, organized descriptions that explain the essential characteristics or extensions of objects, showing how disciplinary fields organize objects systematically. Characteristic words include "是" (is), "是指" (refers to), "定义为" (defined as). Description rules are summarized in .

(2) Fact Knowledge Element Rules. Fact knowledge elements include natural and social existence and evolution information, describing research backgrounds, existing problems, expert opinions, etc. Following [27], we categorize them into opinion, sequence, direct statement, analysis/prediction, and event types. Opinion-type elements express viewpoints with simple structures: opinion holder + content + interpretation. Sequence-type elements use formal descriptions with sequential connectives. Direct statement-type elements have no specific rhetorical patterns. Prediction-type elements use words like "发现" (discovered) and "根据" (according to). Event-type elements complete statements involving time, location, and participants. Rules are shown in .

(3) Numerical Knowledge Element Rules. Numerical knowledge elements describe properties and laws from numerical perspectives (length, height, currency, time, weight, percentages). Literature contains three categories: cardinal numbers, quantity information, and numerical knowledge elements. The latter builds on quantity information with quantifiers or symbols. Time expressions include date formats, eras, dynasties; units include measure words; indicators include terms like "论文" (paper) and "文献" (document). We adapted Wen's extraction method [28] to identify valuable numerical sentences and summarize patterns, as shown in .

3.1.2 Procedural Knowledge Element Rules (1) Method Knowledge Element Rules. Method knowledge elements describe usage processes, steps, and conditions. Characteristic words include sequential markers like "首先" (first) and "然后" (then). Description rules are summarized in .

(2) Relationship Knowledge Element Rules. Relationship knowledge elements describe connections between objects, including parallel, hierarchical, improvement, evolution, progressive, inheritance, substitution, and causal rela-

tionships. Static relationships include parallel and hierarchical; dynamic relationships include improvement, evolution, etc., marked by words like "提出了" (proposed) and "改进" (improved). Rules are shown in .

3.2 Knowledge Element Recognition Process Based on the above rules, we match sentences sequentially. Successful matches mark sentences or sentence groups as knowledge elements; otherwise, they are marked as connecting sentences. Before recognition, non-text resources require document parsing and conversion. Pure text undergoes preprocessing (word segmentation, sentence splitting) and storage. The matching process:

1. Check if the text corpus contains more sentences. If yes, select the next sentence in order; if no, proceed to step 5.
2. Check if rule base contains more rules. If yes, select the next rule; if no, mark the sentence as a connecting sentence and return to step 1.
3. Match the sentence against the rule. If successful, mark as candidate and proceed to step 4; if failed, return to step 2.
4. Since matching uses sentences as units, but some knowledge elements span multiple sentences (e.g., connected by "然后", "其", "它", "这", "比如", "而且"), check for these connectives. If present, extend the sentence to include the next one as a single knowledge element; otherwise, mark as knowledge element and return to step 1.
5. Store matched knowledge elements and connecting sentences in database by their positions. The process is illustrated in [Figure 3: see original paper].

4. Hierarchical Segmentation Method Based on Knowledge Elements

For clarity, we refer to knowledge elements and connecting sentences collectively as "short texts." The method is shown in [Figure 4: see original paper]. First, we calculate similarity between short texts using longest common substrings to build a similarity matrix, where each row represents a short text vector. Then, we treat all short texts as one class and apply Fisher optimal segmentation for hierarchical bisection until all topics are identified, grouping topic-related short texts into semantic paragraphs with maximum internal similarity and maximum dissimilarity between adjacent paragraphs, thereby identifying boundaries.

4.1 Short Text Vector Construction Traditional vector space models suit long texts but suffer from severe data sparsity for short texts and fail to capture word dependencies. We address this by computing longest common substrings for similarity calculation.

Let $G = \{s_1, s_2, \dots, s_m\}$ denote a document with m short texts, where s_i is the i-th short text. Let $s = \{w_1, w_2, \dots, w_n\}$ denote a short text with n words, where w_i is the i-th word.

First, for two short texts $s_1 = \{w_{11}, w_{12}, \dots, w_{1i}\}$ and $s_2 = \{w_{21}, w_{22}, \dots, w_{2j}\}$

with lengths i and j, we use dynamic programming to identify the longest common substring:

$$L[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ L[i-1, j-1] + 1 & \text{if } w_{1i} = w_{2j} \\ \max\{L[i-1,j], L[i,j-1]\} & \text{if } w_{1i} \neq w_{2j} \end{cases}$$

where $L[i,j]$ represents the longest common substring length between s_1 and s_2.

Then, similarity is calculated as:

$$\text{sim}(s_1, s_2) = \frac{L(i,j) + (\max(i,j) - \min(i,j))}{\max(i,j)}$$

where $\max(i,j)$ and $\min(i,j)$ are the maximum and minimum of i and j.

Next, we build similarity matrix A from all pairwise similarities and extract short text vectors from A.

4.2 Fisher Optimal Segmentation for Hierarchical Text Segmentation To preserve short text order, we use Fisher optimal segmentation [30].

4.2.1 Segment Diameter Definition. Let n be the total number of short texts $\{s_1, s_2, \ldots, s_n\}$. Each segment maintains linear order and can be represented as $\{s_i, s_{i+1}, \ldots, s_{i+k}\}$ or $\{i, i+1, \ldots, i+k\}$. For a segment $\{s_i, s_{i+1}, \ldots, s_j\}$ where $1 \leq i < j \leq n$, its vector mean $\bar{s}_{ij}$ and diameter $D(i,j)$ are:

$$\bar{s}_{ij} = \frac{1}{j-i+1} \sum_{r=i}^{j} s_r$$

$$D(i,j) = \sum_{r=i}^{j} (s_r - \bar{s}_{ij})^T (s_r - \bar{s}_{ij})$$

4.2.2 Loss Function Definition. Dividing n ordered short texts into k segments yields:

$$\{\{i_1, i_1+1, \ldots, i_2-1\}, \{i_2, i_2+1, \ldots, i_3-1\}, \ldots, \{i_k, i_k+1, \ldots, n\}\}$$

where $1 = i_1 < i_2 < \ldots < i_k < n$. The loss function is:

$$L[b(n,k)] = \sum_{r=1}^{k} D(i_r, i_{r+1} - 1)$$

Smaller values indicate better segmentation. The optimal segmentation $p(n, k)$ minimizes this function.

4.2.3 Optimal Segmentation Solution. We employ hierarchical bisection:

$$L[b(n, 2)] = \min_{2 \le j \le n} \{D(1, j-1) + D(j, n)\}$$

Select j as the split point minimizing the sum of diameters. Continue bisecting the resulting segments (e.g., G_1, G_2) recursively until reaching optimal solution $p(n, k)$.

4.2.4 Optimal Segment Number Determination. Finer segmentation isn't always better. We plot $L[p(n, k)]$ versus k ($k > 1$) and identify the inflection point to determine k. The curve's slope difference is calculated as:

$$\alpha(k) = \frac{L[p(n, k-1)] - L[p(n, k)]}{(k-1) - k} - \frac{L[p(n, k)] - L[p(n, k+1)]}{k - (k+1)}$$

When $|\alpha(k)|$ reaches maximum, k indicates the inflection point. We select this and nearby values as candidate segment numbers.

5. Experiments

5.1 Test Corpus Selection We selected the paper "A Survey on Text Segmentation" from CNKI as our test case. This paper contains multiple knowledge types (fact, method, conclusion, numerical) with independent yet connected paragraphs. Five researchers manually segmented the text, following majority rule to establish a standard of $k = 17$ segments.

5.2 Experimental Content First, we built a database to store identified knowledge elements and connecting sentences in text order, as shown in [Figure 5: see original paper]. After word segmentation preprocessing and hierarchical bisection, the loss function trend is shown in [Figure 6: see original paper] and slope differences in [Figure 7: see original paper].

The maximum slope difference occurs at $k = 10$, with nearby points also showing large differences, making $k = 10, 11, 12, 13$ candidates. [Figure 6: see original paper] shows the minimum loss function at $k = 13$, which we selected as the algorithm's output. The segmentation result is shown in [Figure 8: see original paper], where each number represents a short text (92 total), with shaded areas indicating final segments.

5.3 Evaluation We use precision (P), recall (R), and F-score:

$$P = \frac{\text{Correctly identified boundaries}}{\text{Algorithm boundaries}}$$

$$R = \frac{\text{Correctly identified boundaries}}{\text{Standard boundaries}}$$

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

For proximity evaluation, we adopt WindowDiff [31]:

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

where $b(i, j)$ counts boundaries between short texts s_i and s_j, N is total short texts, ref is manual segmentation, hyp is algorithm segmentation, and k is half the average segment length. Lower WindowDiff indicates better performance.

5.4 Results Analysis We compared our method with Hearst's classic TextTiling (TT) algorithm [7], as shown in .

Our algorithm achieves higher precision ($0.5 > 0.4$), recall ($0.57 > 0.50$), and F-score ($0.58 > 0.44$), with lower WindowDiff ($0.422 < 0.483$), demonstrating superior effectiveness. Key reasons:

1. Longest common substring similarity calculation overcomes vector space model limitations (data sparsity, no dependency capture) for more accurate similarity measurement.
2. Knowledge element-based segmentation ensures logically complete units, reducing boundary identification errors.
3. Our top-down hierarchical bisection achieves global optimization, while TT's local similarity approach struggles with long texts.

6. Conclusion

To hierarchically segment structured documents and improve accuracy and efficiency, we use knowledge elements as processing units. We summarize knowledge element types and description rules, identify them via cue words, treat them as a class, and apply Fisher optimal segmentation through hierarchical bisection to form hierarchical structures. This advances knowledge service control from documents to knowledge element blocks, meeting multi-granular user needs. Compared with TextTiling, our method shows advantages in precision and proximity evaluation, proving reasonable and effective.

This study focuses on hierarchical segmentation using knowledge elements. Due to varying writing habits and disciplinary differences, we manually summarized description rules; automated extraction remains for future research.

References

[1] Shi Jing. A survey on text segmentation[J]. Computer Engineering and Applications, 2006, 42(35): 155-159.
[2] REYNAR JC. Topic segmentation: algorithms and applications[D]. Computer and information science, Philadelphia: University of Pennsylvania, 1998.
[3] Zou Jian, Zhong Maosheng, Meng Li. Chinese text segmentation pattern acquisition and optimization method[J]. Journal of Nanchang University (Natural Science), 2011, 35(6): 597-601.
[4] HALLIDAY M AK, HASAN R. Cohesion in English[M]. London: Routledge, 1976.
[5] KOZIMA H. Text segmentation based on similarity between words[C]//Proceedings of the 31st annual meeting on association for computational linguistics. Stroudsburg, PA, USA: association for computational linguistics, 1993: 286-288.
[6] REYNAR JC. An automatic method of finding topic boundaries[C]//Proceedings of the 32nd annual meeting on association for computational linguistics. Stroudsburg, PA, USA: association for computational linguistics, 1994: 331-333.
[7] HEARST M A. Multi-paragraph segmentation of expository text[C]//Proceedings of the 32nd annual meeting on association for computational linguistics. Stroudsburg, PA, USA: association for computational linguistics, 1994: 9-16.
[8] CHOI FYY. Advances in domain independent linear text segmentation[C]//Proceedings of the 2000 NAACL-ANLP workshop on Text summarization. Stroudsburg, PA, USA: association for computational linguistics, 2000: 26-33.
[9] PONTE JM, CROFT WB. Text segmentation by topic[M]//Research and advanced technology for digital libraries. Heidelberg: Springer Berlin Heidelberg, 1997: 113-125.
[10] MORRIS J, HIRST G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text[J]. Computational linguistics, 1991, 17(1): 21-48.
[11] CHOI FYY, WIEMER-HASTINGS P, MOORE J. Latent semantic analysis for text segmentation[C]//Proceedings of emnlp, 2001, 4(3): 109-117.
[12] Shi Jing, Dai Guozhong. Text segmentation based on PLSA model[J]. Journal of Computer Research and Development, 2007, 44(2): 242-248.
[13] Shi Jing, Hu Ming, Shi Xin, et al. Text segmentation based on LDA model[J]. Chinese Journal of Computers, 2008, 31(10): 1865-1873.
[14] RIEDL M, BIEMANN C. How text segmentation algorithms gain from topic models[C]//Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: human language technologies. Stroudsburg, PA, USA: association for computational linguistics, 2012: 553-557.
[15] EISENSTEIN J, BARZILAY R. Bayesian unsupervised topic segmentation[C]//Proceedings of the conference on empirical methods in natural language processing. Stroudsburg, PA, USA: association for computational linguistics, 2008: 334-343.
[16] MULBREGT P, CARPI, GILLICK L, et al. Text segmentation and topic

tracking on broadcast news via a hidden markov model approach[C]//Fifth international conference on spoken language processing, Sydney, Australia: ISCA Archive, 1998: 2519-2522.
[17] BRANTS T, CHEN F, TSCHANTARIDIS I. Topic-based document segmentation with probabilistic latent semantic analysis[C]//Proceedings of the eleventh international conference on information and knowledge management. New York, NY, USA: ACM, 2002: 211-218.
[18] RIEDL M, BIEMANN C. TopicTiling: a text segmentation algorithm based on LDA[C]//ACL 2012 student research workshop. USA: association for computational linguistics, 2012: 37-42.
[19] KAN MY. Combining visual layout and lexical cohesion features for text segmentation[C]//Proceedings of the 31st Workshop on graph theoretic concepts in computer science? WG2005, 2001: 187-198.
[20] YAARI Y. Segmentation of expository texts by hierarchical agglomerative clustering[EB/OL]. [2018-03-21] https://arxiv.org/pdf/cmp-lg/9709015v1.pdf.
[21] EISENSTEIN J. Hierarchical text segmentation from multi-scale lexical cohesion[C]//Human language technologies: the conference of the North American chapter of the association for computational linguistics. Stroudsburg, PA, USA: association for computational linguistics, 2009: 353-361.
[22] TEH YW, JORDAN MI, BEAL MJ, et al. Hierarchical dirichlet processes[J]. Journal of the American statistical association, 2006, 101(476): 1566-1581.
[23] Li Tiancai, Wang Bo, Xi Yaoyi, et al. Text segmentation based on hierarchical Dirichlet process model[J]. Journal of Data Acquisition and Processing, 2017, 32(2): 408-416.
[24] Wen Youkui. Knowledge organization and retrieval based on "knowledge element"[J]. Computer Engineering and Applications, 2005, 41(1): 55-57.
[25] Zhang Jing, Liu Yanshen, Wei Jinlei. On the construction of multimedia knowledge element database for primary and secondary schools[J]. Modern Educational Technology, 2005, 15(5): 68-71.
[26] Yuan Xiaoling. Knowledge indexing based on knowledge element[J]. Library Science Research, 2007(6): 45-47.
[27] Zhao Rongying, Zhang Xinyuan. Research on description rules of Chinese think tank achievements based on knowledge element extraction[J]. Library and Information, 2017(1): 119-127.
[28] Wen Youkui, Wen Hao, Xu Duanyi, et al. Text knowledge indexing based on knowledge element[J]. Journal of the China Society for Scientific and Technical Information, 2006, 25(3): 282-288.
[29] Hua Bolin. Research on types and description rules of method knowledge elements in academic papers[J]. Journal of Library Science in China, 2016, 42(1): 30-40.
[30] Xiao Cong, Gu Shengping, Cui Wei, et al. Application of Fisher optimal segmentation method in flood season staging of Lixian River basin[J]. Water Resources and Power, 2014(3): 70-74.
[31] PEVZNER L, HEARST MA. A critique and improvement of an evaluation metric for text segmentation[J]. Computational linguistics, 2002, 28(1): 19-36.

Author Contributions:
Wang Zhongyi: Conceptualization, methodology design
Shen Xueying: Writing, data collection, experimentation
Huang Jing: Writing review and refinement

Note: Figure translations are in progress. See original paper for figures.