International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-019-00319-y

ORIGINAL PAPER

Metro maps for efficient knowledge learning by summarizing massive electronic textbooks

Weiming Lu[1] · Pengkun Ma[1] · Jiale Yu[1] · Yangfan Zhou[1] · Baogang Wei[1]

Received: 8 June 2018 / Revised: 29 December 2018 / Accepted: 23 February 2019

Abstract

As the number of textbooks soars, people may be stuck into thousands of books when learning knowledge. In order to provide a concise yet comprehensive picture for learning, we propose a novel framework, called **MM4Books**, to automatically build metro maps for efficient knowledge learning by summarizing massive electronic textbooks. We represent each book in digital libraries as a sequence of chapters, and then obtain learning objects by clustering the semantically similar chapters via an unsupervised clustering method to create a learning graph, and then build the metro map by applying an integer linear programming-based technique to select a collection of high informative and fluent but low redundant learning paths from the learning graph. To the best of our knowledge, it is the first work to address this task. Experiments show that our proposed approach outperforms all the state-of-the-art baseline approaches, and we also implemented a practical **MM4Books** system to prove that users can really benefit from the proposed approach for knowledge learning.

Keywords Knowledge learning · Massive book summarization · Learning path · Digital library

1 Introduction

Textbooks are the primary conduits for delivering knowledge to the students and the teachers [8]. In the recent 10 years, several projects, such as Google Books[1] and Million Book Project,[2] have digitized millions of books, which greatly facilitate the users to find and read books. However, "Distringit librorum multitudo" (the abundance of books is a distraction), said Lucius Annaeus Seneca, who lived in the first century. When learning a subject, you may be stuck into thousands of books. Therefore, it would be very helpful for knowledge learning if all these books can be synthesized into a concise yet comprehensive picture.

Many approaches are already developed for summarizing and visualizing news [4,11,24,26–28,37,40,41], scientific literature [23,36], user-generated content (UGC) [6,25,29,32,42,43] and patent [38] to solve this issue of information overload. However, to the best of our knowledge, no previous work has been proposed to synthesize books with the same subject into a comprehensive picture for effective learning. In addition, all the above-mentioned approaches for summarizing news, scientific literature, UGC and patent are also not suitable to address our problem. The reasons are: (1) content similarity is a useful hint to link articles and further to generate storyline-like summarization. For example, Wang et al. [28] and Zhang et al. [38] applied standard "bag-of-words" representation and cosine measure to calculate content similarity. However, books are often lengthy with rich content. Even for each chapter, it is still more lengthy and complex than news, articles, comments and patents. Therefore, it is not suitable to compute content proximity by cosine measure with "bag-of-words." Furthermore, it is even hard to obtain the text of books in reality. So we have to represent each chapter by its title. But titles are very short sentences, which makes more challenge for semantic similarity computation. (2) Books lack temporal information and citation information, which are also often used for summarizing news, scientific literature and patent. Fortunately, chapters in books are organized in sequence, which can be used in summarization.

Knowledge learning is a complex and multi-staged process, which should exhibit a very nonlinear structure. Therefore, linear-structured storylines and timelines are not suitable for our problem. Inspired by Shahaf et al. [23,24],

---

[1] http://books.google.com.
[2] http://en.wikipedia.org/wiki/Million_Book_Project.

✉ Weiming Lu
luwm@zju.edu.cn

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, China

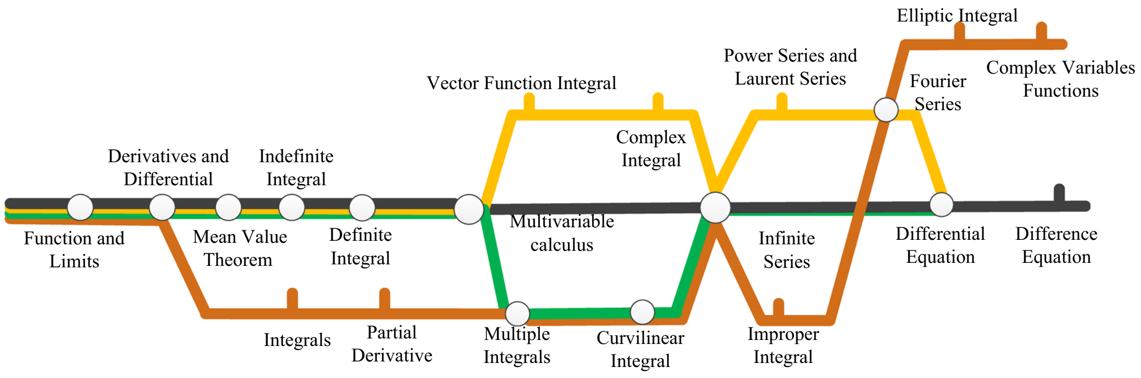


Fig. 1 Example of a Metro Map from massive electronic textbooks for learning *Calculus*

we want to build metro maps to summarize massive textbooks for efficient learning, where several learning paths are included for knowledge learning guidance. Figure 1 shows an example of a metro map for learning *Calculus*, which is generated by summarizing massive electronic textbooks about *Calculus*.

We think a good metro map for efficient knowledge learning should satisfy the following properties: (1) *High informativeness*. The metro map should contain knowledge-intensive learning paths. (2) *Good fluency*. The learning paths selected in the metro map should be fluent, so users can learn knowledge easily. (3) *High coverage*. The metro map should cover as much knowledge as possible and has less redundancy.

In this paper, we propose a novel framework, called **MM4Books**, to build metro maps for efficient knowledge learning by summarizing massive textbooks. The main contributions are summarized as follows:

- We are the first to propose a framework for massive textbooks summarization. The framework considers both informativeness, fluency and coverage of the metro map.
- We address the very short text similarity problem through *weighted word2vec* for learning object acquirement. The experiment validates it outperforms *paragraph2vec*.
- We create a practical dataset by searching books in a digital library, and our approach reaches the best performance compared to the state-of-the-art approaches. Moreover, we implement a practical **MM4Books** system in our digital library.

The rest of this paper is organized as follows. Section 2 provides an overview of the related work. We formally define the massive book summarization problem and elaborate the **MM4Books** in Sect. 3. Section 4 presents the experiments and results, and Sect. 5 demonstrates our **MM4Books** system and presents a human evaluation. Finally, Sect. 6 concludes the paper.

2 Related work

2.1 Learning aids with textbooks

Several approaches have been proposed to facilitate learning with textbooks.

On one hand, some approaches link relevant supplementary materials to textbooks for learning enhancement. For example, Csomai and Mihalcea [5] linked encyclopedic information to educational material. While Agrawal et al. [1] enriched textbooks by finding images from the web that are most relevant for augmenting a section of the textbook. Furthermore, they presented a diagnostic tool for algorithmically identifying those sections of a book that are not well-written and described techniques for algorithmically augmenting different sections of a book with links to selective articles and images mined from the Web [2]. Kokkodis et al. [14] assigned educational videos at appropriate locations in textbooks to further augment the educational experience.

Other approaches try to mine the textbooks to provide learning aids. For example, *study navigator* [3] tried to help students learn the textbooks better and faster by providing easy access to concepts explained elsewhere in the book that are most relevant for understanding the present section. Wu et al. [35] developed a back-of-the-book index generation method to help users locate information in a textbook.

The most similar works include [15,17,30,31]. Larranaga et al. [15] presented a system called DOM-Sortze for the semiautomatic generation of the Domain Module from electronic textbooks. They tried to identify the structural and sequential relationships between items of the outline. However, they resorted to a set of heuristics and only focused the local features of the two items without considering the global view of the massive textbooks. Wang et al. [30] extracted concepts in book chapters using Wikipedia and constructed a concept hierarchy for one book. They can only construct the concept hierarchy for one book, but do not consider multiple books in the same domain. BBookX [17] tried to automatically and collaboratively build free open online books using publicly available educational resources such as Wikipedia. However, the catalog of the book is created manually through a user interface. Wang et al. [31] proposed a model for concept map extraction from textbooks, but they also only considered one book.

Our work is quite different from these works. First, these works only considered the relationships between concepts, while our work can also be appropriate for more fine-grained knowledge, such as *the definition of definite integral*, *the properties of definite integral* and *the calculation of definite integral*. Second, Wang et al. [30,31] can only extract relations from just one book, while our work considers multiple books in one domain, and can build a concise yet comprehensive view for massive textbooks. Third, our work is totally automatical, but BBookX [17] manually created the catalog of a book through a user interface.

2.2 Content organization

In accordance with human cognition, a natural way for learning is to provide a structured, concise yet comprehensive view of knowledge. Recently, researchers have made some progress by organizing news, scientific literature, user-generated content (UGC) and patent as storylines, metro maps and hierarchical structures.

Storylines aim to connect the disparate documents in sequence. For example, Wang et al. [28] generated pictorial storylines for given topics from text and image data on the Internet for providing a sketch of the topic evolution. Zhang et al. [38] proposed *PatentLine* to generate a technology evolution tree for a given set of granted patents. Wang et al. [29] generated socially informed timelines that contain both news article summaries and selected user comments. Tang et al. [26] proposed a hddCRP (hybrid distant-dependent Chinese Restaurant Process)-based hierarchical topic model for news story generation. Zhou et al. [40] proposed an unsupervised Bayesian model to extract structured representations and evolution patterns of storylines, and they further proposed a nonparametric generative model [41] to extract them simultaneously. Hu et al. [11] explored the interactions of storylines in a news topic. Sigurdsson et al. [25] generated storylines of visual concepts from web data using skipping recurrent neural network. While Chen et al. [4] generated multimodal storylines via generative adversarial imitation learning.

Metro maps can exhibit more complex stories. For example, Shahaf et al. [24] generated metro maps on news articles in aiding users navigate, consume, and integrate difference aspects of a multi-faceted information need. In [23], they further constructed metro maps of scientific literature to help users acquire knowledge efficiently.

In addition, hierarchical structures are also widely used for content organization. For example, Zhu et al. [42] automatically generated topic hierarchy to organize information from multiple UGC sources. Dou et al. [6] proposed *HierarchicalTopics* to support scalable exploration and analysis of document corpora based on a large number of topics. Zhu et al. [43] constructed a focused topic hierarchy to organize social media contents.

3 Proposed approach

3.1 Problem formulation and framework

In this section, we define several concepts and our problem and then give the overall procedure of our approach.

Textbook. A textbook contains its title and a table of content (TOC), which can be denoted as $b_i = \{t_i, \text{toc}_i\}$. TOC organizes chapters in either sequential or structural relationships. The structural relationships exist between a chapter and its subchapters, while the sequential relationships exist among chapters in the same level. In our approach, we focus on the sequential relationships, since knowledge is learned mainly in sequence. We observed that the top-level chapters in most books follow the sequential relationships, while subchapters of one chapter may follow the coordinate relationships, which will bring in noise. Therefore, we only consider the top-level chapters for simplicity. That is to say, toc_i can be considered as a sequence of top-level chapters: $\text{toc}_i = \{c_{i_j}\}_{j=1}^{|\text{toc}_i|}$. Here, the chapter is represented by its title. Thus, c_{i_j} is the title of the jth chapter in the textbook i in the sequential order in TOC. Finally, the collection of textbooks is denoted as $\mathcal{B} = \{b_i\}_{i=1}^{M}$.

Learning Object. A learning object o is a collection of chapters with the purpose of learning a specific knowledge item, which is an atomic piece of knowledge for learning. For example, two chapters *Indefinite Integral* and *Definite Integral* belong to two different learning objects, while *Indefinite Integral Overviews* and *Introduction to Indefinite Integral* belong to one learning object.

Learning Path. A learning path is a path for one to follow in order to reach his target knowledge. For example, if one wants to learn the knowledge about *Calculus*, he/she should follow a learning path such as *Function and Limits* → *Derivatives and Differential* → ⋯ → *Differential Equation* → *Difference Equation*. Therefore, a learning path is a sequence of learning objects for knowledge learning, and it can be denoted as $p = o_1 \rightarrow o_2 \rightarrow \cdots \rightarrow o_{|p|}$.

Metro Map. A metro map is a structured summaries of knowledge, which is formed by a collection of high informative and coherent but low redundant learning paths with the same

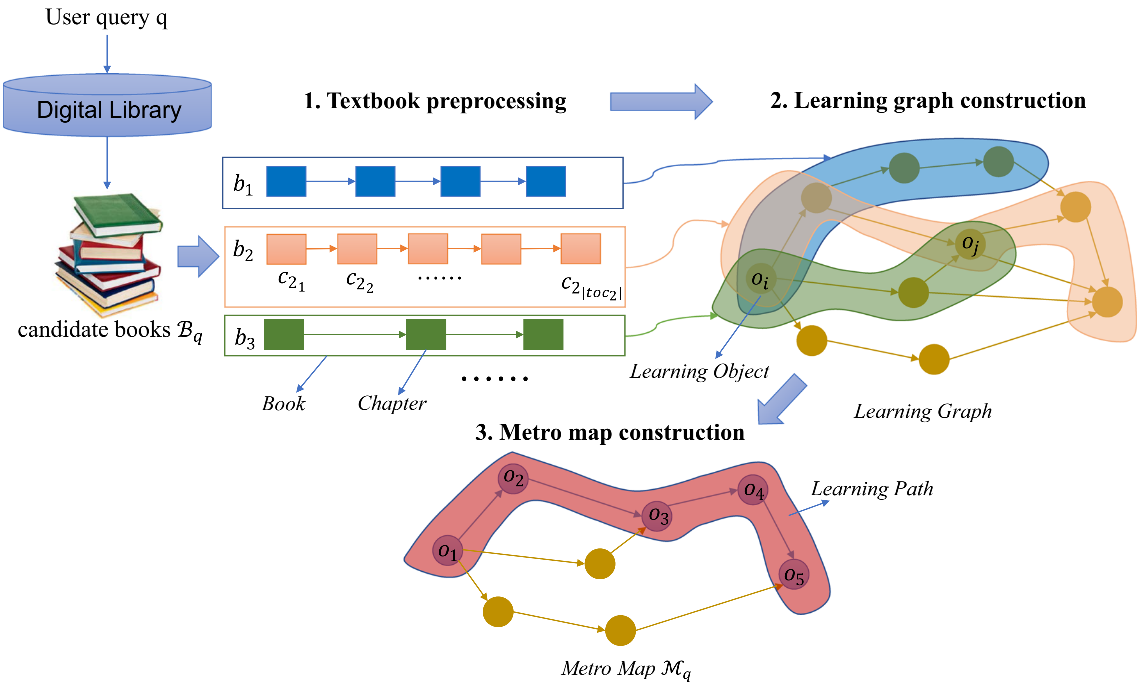


Fig. 2 The procedure of Metro Map generation for user query q. It consists of three steps: textbook preprocessing, learning graph construction, and metro map construction

learning objective. Thus, it can be denoted as $\mathcal{M} = \{G, \Pi\}$, where $G = \{V, E\}$ is a directed graph and Π is a set of learning paths in G. Each node $v \in V$ refers to a learning object, and each edge $e \in E$ belongs to at least one learning path.

Given the above concepts, we formally define our problem as follows.

Problem 1 (*Metro Map for Textbooks*) **Given** a query q and a set of electronic textbooks $\mathcal{B}$, the metro map generation task **aims to** build a metro map $\mathcal{M}_q$ for efficient learning the knowledge about q, where $\mathcal{M}_q$ contains a collection of high informative and coherent but low redundant learning paths Π_q.

The overall procedure of the MM4Books (as shown in Fig. 2) is summarized below:

1. *Textbook preprocessing*. Collect the candidate books $\mathcal{B}_q$ by searching the query q in the titles of $\mathcal{B}$, and then generate the sequence of chapters for each book in $\mathcal{B}_q$. As shown in Fig. 2, the chapters of the book $b_2 \in \mathcal{B}_q$ can be represented as $\{c_{2_j}\}_{j=1}^{|\text{toc}_2|}$.
2. *Learning graph construction*. Acquire the learning objects by clustering the semantical similar chapters via an unsupervised clustering method, and then construct the learning graph. As shown in Fig. 2, three chapters $c_{1_1} \in b_1$, $c_{2_1} \in b_2$ and $c_{3_1} \in b_3$ are mapped to the learning object o_i, and chapter $c_{2_3} \in b_2$ and $c_{3_3} \in b_3$ are mapped to the learning object o_j. Then, the chapters of b_1, b_2 and b_3 are all mapped to paths for learning graph construction by chaining the learning objects into a set of learning paths.
3. *Metro map construction*. Select several high informative and coherent but low redundant learning paths by applying an integer linear programming technology to build the metro map.

3.2 Textbook preprocessing

Each electronic textbook in digital libraries follows the Dublic Core (DC) and Open eBook (OEB) standards, where the metadata such as title, authors and publisher of a book is encoded, and the TOC of a book is organized with a hierarchical structure in XML format by chapters.

On one hand, in order to quickly collect the candidate books from massive books for user's query q, we index the title of books in $\mathcal{B}$ using Lucene,[3] and then search the query q through the index to get the $\mathcal{B}_q$.

On the other hand, we parse the TOC by a XML parser to obtain the sequence of top-level chapters $\text{toc}_i = \{c_{i_j}\}_{j=1}^{|\text{toc}_i|}$ for each book $b_i \in \mathcal{B}_q$. Since the prefix strings in the chapter titles such as *Chapter 1*, *1.1*, *1.1.1* and ♯1 will confuse the similarity computation between chapters, so we use a regular

---

[3] http://lucene.apache.org.

expression to refine the title of each chapter by removing these prefix strings in advance.

3.3 Learning object acquirement

In this section, we mainly focus on learning object acquirement by clustering chapters in $\mathcal{B}_q$ into groups. Then, each group can be considered as a learning object.

The main challenge of learning object acquirement is the similarity computation among chapters. Actually, it is the very short text similarity problem, since each chapter is represented by its short title for simplicity, and the similarity between two chapters is considered as the similarity between their titles.

However, it is nontrivial to calculate the similarity between these titles for learning object acquirement.

On one hand, there are no sufficient training data to train a supervised approach, so recent short text similarity learning approaches for short text [10,12,13,18,22,33,34] are not suitable.

On the other hand, unsupervised approaches like sentence embeddings [16], which was inspired by the Skip-gram model, can derive sentence vectors for sentence similarity calculation. However, using sentence embedding approach like Paragraph2vec directly for chapter similarity computation would cause mistakes. Taking the following simple example, suppose there are three titles *Introduction to Indefinite Integral*, *Indefinite Integral Overviews* and *Introduction to Definite Integral*. Because *Indefinite Integral* and *Definite Integral* are very related to each other, they could have similar embedding vectors according to the skip-gram model, which would lead to that the similarity between *Introduction to Indefinite Integral* and *Introduction to Definite Integral* is larger than the similarity between *Introduction to Indefinite Integral* and *Indefinite Integral Overviews*.

According to the above example, we observe that words in sentences are not equal. For example, *Introduction* $\sim$ *Overviews* and *Indefinite Integral* $\sim$ *Definite Integral* ($\sim$ means *similar to*) should not be treated equally. Therefore, we differentiate words in titles (after removing stop words) into two types: *topic words* and *aspect words*. *topic word* indicates the topic of a chapter, while *aspect word* indicates the aspect of a topic in a chapter. For example, in Chapter *Introduction to Indefinite Integral*, *Indefinite Integral* is the *topic word*, and *Introduction* is the *aspect word*. Obviously, it requires more stringent for the similarity between *topic word*s. Formally, we use $t = \{\{\mathrm{tw}_i\}_{i=0}^{|\mathrm{tw}|}, \{\mathrm{aw}_i\}_{i=0}^{|\mathrm{aw}|}\}$ to represent a title, where tw and aw are topic words and aspect words in the title t, and each of them is represented by word2vec as v_{tw_i} and v_{aw_i}. Finally, title t can be represented by the average vectors of *topic word*s and *aspect word*s as $t = \{v_{\mathrm{tw}}(t) = \sum_i v_{\mathrm{tw}_i}/|\mathrm{tw}|, v_{\mathrm{aw}}(t) = \sum_i v_{\mathrm{aw}_i}/|\mathrm{aw}|)\}$.

Table 1 Top 20 *aspect word*s in the ranking list, and the English words in brackets are the translated words

方法 (approach)	概述 (overview)
应用 (application)	分类 (classification)
问题 (problem)	功能 (function)
结构 (structure)	作用 (effect)
技术 (technology)	原理 (principle)
研究 (research)	概念 (concept)
特点 (characteristic)	特性 (characteristic)
过程 (procedure)	条件 (condition)
发展 (development)	性质 (property)
方式 (way)	组成 (composition)

Thus, the similarity between title t_x and t_y can be calculated as weighted word embedding aggregation as follows.

1. if $v_{\mathrm{aw}}(t_x) = v_{\mathrm{aw}}(t_y) = \mathbf{0}$, then $\mathrm{sim}(t_x, t_y) = \cos(v_{\mathrm{tw}}(t_x), v_{\mathrm{tw}}(t_y))$.
2. if $v_{\mathrm{aw}}(t_x) = \mathbf{0}$ or $v_{\mathrm{aw}}(t_y) = \mathbf{0}$, then $\mathrm{sim}(t_x, t_y) = \alpha \cos(v_{\mathrm{tw}}(t_x), v_{\mathrm{tw}}(t_y))$.
3. if $\cos(v_{\mathrm{tw}}(t_x), v_{\mathrm{tw}}(t_y)) \geq \eta$, $\cos(v_{\mathrm{aw}}(t_x), v_{\mathrm{aw}}(t_y)) \geq \zeta$, then $\mathrm{sim}(t_x, t_y) = \beta \cos(v_{\mathrm{tw}}(t_x), v_{\mathrm{tw}}(t_y)) + (1 - \beta) \cos(v_{\mathrm{aw}}(t_x), v_{\mathrm{aw}}(t_y))$.
4. otherwise $\mathrm{sim}(t_x, t_y) = 0$.

We set $\alpha = 0.9$, $\beta = 0.8$, $\eta = 0.8$ and $\zeta = 0.6$ experimentally. Finally, we use repeated bisection clustering approach [39] for title clustering, and retain clusters that have an intra-cluster similarity of at least $\mathrm{ics} = 0.9$.

The remaining problem is how to determine *topic word*s and *aspect word*s in titles. Here, we use a simply heuristic-based approach to build two word sets: *topic word set* and *aspect word set*. At first, we segment each title into words, and remove the stop words. Then, we calculate the *inverse book frequency* (IBF) for each word. The IBF measures whether the word is common or rare across all books. The words with small IBF value are more likely to be *aspect word*s. So we rank words according to IBF values in ascending order, and select top 5000 words for human to judge whether it is indeed an *aspect word*. Other words are considered as *topic word*s. Table 1 shows the top 20 *aspect word*s.

3.4 Metro map construction

In this section, we mainly focus on metro map construction. We firstly construct a *learning graph* for $\mathcal{B}_q$, and then select a set of candidate learning paths from the *learning graph*. Later, we define *informativeness* and *fluency* for the learning path. Finally, we apply integer linear programming (ILP) to select a collection of high informative and fluent but low redundant learning paths to build the metro map.

3.4.1 Learning graph construction

Learning graph is a directed graph with learning objects serving as nodes and edges representing adjacency relations.

Given the candidate books $\mathcal{B}_q$, the graph is constructed by first creating a start and end node, called s and e node, and then iteratively adding each book to it. When adding a book $b_i \in \mathcal{B}_q$, the approach first checks each chapter $c_{i_j} \in \text{toc}_i$ to see if it can be mapped onto existing nodes in the graph. If c_{i_j} belongs to a learning object o and o has been the node in the graph, then c_{i_j} is mapped onto the o in the graph. Otherwise, o is inserted into the graph as a new node, and then c_{i_j} is mapped onto it. With the sequence $\{c_{i_j}\}_{j=1}^{|\text{toc}_i|}$, a path $s \rightarrow o_{i_1} \rightarrow o_{i_2} \rightarrow \cdots \rightarrow o_{i_{|\text{toc}_i|}} \rightarrow e$ can be added to the graph, where c_{i_j} is mapped onto the node o_{i_j}. An example of *learning graph* is shown in Fig. 2.

3.4.2 Learning paths selection

We can use dynamic programming algorithm to traverse the *learning graph* from s to e to obtain all learning paths. However, there would be too many paths in a graph, since hundreds of learning objects are contained in the graph. Therefore, we have to ignore some paths to reduce the number of paths.

Firstly, we set the minimum path length (in learning objects) to six. Then, during the traversal, we also bypass the edges which only appear once in all books in $\mathcal{B}_q$. Moreover, no node can be added twice in one learning path, which can guarantee no loop in the path. Now, we obtain a set of learning paths called $\mathcal{P}_q$. Our aim becomes to select the top K best paths from $\mathcal{P}_q$ to form a metro map.

We introduce two factors *Informativeness* ($I(p_i)$) and *Fluency* ($F(p_i)$) for the path selection.

Informativeness measures whether a path is knowledge-intensive. Inspired by [7], we use $w(o_i, o_{i+1})$ to measure the informativeness of a learning object pair $\{o_i, o_{i+1}\}$, and the formula is shown as:

$$w(o_i, o_{i+1}) = \left[\frac{\text{frq}(o_i) + \text{frq}(o_{i+1})}{2}\right] \cdot \sum_{b \in \mathcal{B}_q} \text{dif}(b, i)^{-1}$$

where $\text{frq}(o_i)$ is the number of chapters in o_i, $\text{dif}(b, i)$ is the distance between the offset positions of o_i and o_{i+1} in the sequence of chapters of book b. That is to say, if two chapters c_{b_x} and c_{b_y} are in the same book b, and they are mapped into o_i and o_{i+1}, respectively, then $\text{dif}(b, i) = max(1, \text{offset}(c_{b_y}) - \text{offset}(c_{b_x}))$. Otherwise, $\text{dif}(b, i) = \infty$. Here, $\text{offset}(c_{b_x})$ is the location of chapter c_{b_x} in the sequence of top-level chapters $\text{toc}_b = \{c_{b_i}\}_{i=1}^{|\text{toc}_b|}$ in the book b. So $\text{offset}(c_{b_x}) = x$.

Then, the informativeness of the path $p_i = s \rightarrow o_1 \rightarrow o_2 \rightarrow \cdots \rightarrow o_{|p_i|} \rightarrow e$ in $\mathcal{P}_q$ can be calculated as:

$$I(p_i) = \frac{\sum_{i=1}^{|p_i|-1} w(o_i, o_{i+1})}{|p_i|}$$

Fluency measures whether a learning path can be easily followed. We use a language model to compute the *Fluency* for a path $p_i = s \rightarrow o_1 \rightarrow o_2 \rightarrow \cdots \rightarrow o_{|p_i|} \rightarrow e$, and the formula is shown as:

$$F(p_i) = \frac{1}{1 - \log_2 \sum_{i=3}^{|p_i|} p(o_i|o_{i-2}o_{i-1})/(|p_i| - 2)}$$

where $p(o_i|o_{i-2}o_{i-1})$ is the probability of o_i when given $o_{i-2}o_{i-1}$, and it can be calculated by $p(o_i|o_{i-2}o_{i-1}) = \frac{c(o_{i-2}o_{i-1}o_i)}{c(o_{i-2}o_{i-1})}$. Here, $c(o_{i-2}o_{i-1}o_i)$ is the number of chapter sequences $c_{i_m}c_{i_n}c_{i_o}$ when c_{i_m}, c_{i_n} and c_{i_o} ($m < n < o$) are mapped into o_{i-2}, o_{i-1} and o_i, respectively. It is similar for $c(o_{i-2}o_{i-1})$. Finally, $I(p)$ and $F(p)$ are both normalized to the maximum value of its metro map.

To select the top K best paths, we combine *Informativeness* and *Fluency* together in an integer linear programming (ILP) optimization framework. At first, the score of path p_i is $s(p_i) = I(p_i) \cdot F(p_i)$. Then, we maximize the following objective function:

$$\sum_{i=1}^{|\mathcal{P}_q|} s(p_i) \cdot d_i$$

where d_i is a binary variable, that takes 0 or 1, depending on whether the path p_i is selected to form the final metro map or not. There are also several constraints for the problem. First, we ensure that no more than K paths are selected, so $\sum_{i=1}^{|\mathcal{P}_q|} d_i \leq K$. Then, if two paths are similar, then only one path can be selected, so the second constraint is:

$$\forall i, j \in [1, |\mathcal{P}_q|], d_i + d_j \leq 1, \quad \mathit{if}\, sim(p_i, p_j) \geq \phi$$

Here, we use Jaccard similarity to measure the similarity between two paths by treating the path as a bag of learning objects:

$$\text{sim}(p_i, p_j) = \frac{p_i \cap p_j}{p_i \cup p_j}$$

ϕ is a parameter to tune the redundancy of the map. The constraint indicates that if the similarity between two paths is larger than ϕ, then at most one path would be selected. In our experiments, we set $\phi = 0.5$ empirically.

Table 2 Dataset statistics

Query q	$\#\mathcal{B}_q$	$\#c(\mathcal{B}_q)$	#lo	$\#\mathcal{P}_q$
Pathobiology	122	840	84	22,941
Operating system	122	600	64	5410
Sensor	63	347	44	54
Networking	139	723	75	6995
Building materials	67	365	37	1013
Mechanics	130	582	83	243
Software engineering	50	323	38	189
Graphics	28	172	20	30

Table 3 Comparison results for learning object acquirement

Ics	Paragraph2vec			Weighted word2vec		
	Pr	Rec	F1	Pr	Rec	F1
0.65	0.75	0.76	0.76	0.58	0.82	0.68
0.70	0.87	0.71	0.78	0.71	0.85	0.77
0.75	0.91	0.62	0.74	0.68	0.91	0.78
0.80	0.94	0.53	0.68	0.78	0.92	0.84
0.85	0.93	0.53	0.68	0.77	0.92	0.84
0.90	0.97	0.51	0.67	0.92	0.90	0.91
0.95	0.97	0.49	0.65	0.93	0.87	0.90

We solve the above optimization problem by using IBM CPLEX optimizer.[4] Finally, we can easily build the metro map $\mathcal{M}_q$ by assembling these selected K paths.

4 Experiments

To the best of our knowledge, no previous work has addressed the same task as we do in this paper, so there is no public dataset and existing evaluation metrics for evaluation.

In this section, we create an experimental dataset at first, and then evaluate learning object acquirement and metro map construction separately. In each evaluation, we both introduce the evaluation metrics and compared methods and then give the experimental results.

4.1 Datasets

In order to validate our approach, we create the experimental dataset by searching books in a digital library with some queries in different domains.

The statistics of the dataset is listed in Table 2, where $\#\mathcal{B}_q$ and $\#c(\mathcal{B}_q)$ are the total number of books and chapters in the dataset for query q, *#lo* and $\#\mathcal{P}_q$ is the number of learning objects and candidate learning paths for query q obtained from chapters of books in $\mathcal{B}_q$.

4.2 Evaluation on learning object acquirement

Since there is no public dataset for learning objects, we resort to human judgments to compare *paragraph2vec* and our *weighted word2vec* approach for learning object acquirement.

At first, we collect a set of chapters $\mathcal{B}_q$, and then cluster them into groups by *paragraph2vec* and *weighted word2vec*, respectively, which forms $\mathcal{G}_p$ and $\mathcal{G}_w$, and each group $g_i = \{t_{i_1}, t_{i_2}, \ldots, t_{i_M}\}$ contains a set of chapters. For each evaluation, we randomly select a chapter t from $\mathcal{B}_q$, and then obtain groups $g_t^p \in \mathcal{G}_p$ and $g_t^w \in \mathcal{G}_w$, which both contain the chapter t. Then, we ask an evaluator to judge whether titles in group g_t^p and g_t^w are similar to t, and obtain two subsets of titles $\hat{g}_t^p \subseteq g_t^p$ and $\hat{g}_t^w \subseteq g_t^w$ which are similar to t. Finally, we can get the precision, recall and F1 for *paragraph2vec* are: $\mathrm{pr}_p = \frac{|\hat{g}_t^p|}{|g_t^p|}$, $\mathrm{rec}_p = \frac{|\hat{g}_t^p|}{|\hat{g}_t^p \cup \hat{g}_t^w|}$, and $F1_p = \frac{2 \cdot \mathrm{pr}_p \cdot \mathrm{rec}_p}{\mathrm{pr}_p + \mathrm{rec}_p}$. Similarly, the precision, recall and F1 for *weighted word2vec* are: $\mathrm{pr}_w = \frac{|\hat{g}_t^w|}{|g_t^w|}$, $\mathrm{rec}_w = \frac{|\hat{g}_t^w|}{|\hat{g}_t^p \cup \hat{g}_t^w|}$, and $F1_w = \frac{2 \cdot \mathrm{pr}_w \cdot \mathrm{rec}_w}{\mathrm{pr}_w + \mathrm{rec}_w}$. In the experiment, we randomly select 10 chapters, and ask 5 evaluators to evaluate the clustering results by *paragraph2vec* and *weighted word2vec*. The average results are shown in Table 3.

From the table, we can see that *paragraph2vec* can obtain a higher precision, but the recall is intolerable small, which means it would lose many semantic-similar chapters in the learning objects. The reason is that *paragraph2vec* does not differentiate *topic word* and *aspect word*, so it would filter *Introduction to Indefinite Integral* and *Indefinite Integral Overviews* both for the cluster *Introduction to Definite Integral* as discussed in above section. With the increment of the bisection clustering parameter ics, the precision of *weighted word2vec* increases while still maintaining high recalls.

4.3 Evaluation on metro map construction

4.3.1 Evaluation metric

The quality of a path p can be measured by $Q(p) = I(p) \cdot F(p)$ when considering both the *informativeness* and *fluency* of the path, so the total quality of a metro map $\mathcal{M} = \{G, \Pi\}$ can be measured by $Q(\mathcal{M}) = \sum_{p \in \Pi} Q(p)$. In addition, the *redundancy* of the map can be measured by $R(\mathcal{M}) = \frac{\sum_{p_i, p_j \in \Pi, i<j} sim(p_i, p_j)}{|\Pi| \cdot (|\Pi| - 1)/2}$. Thus, inspired by F_1, the total score of a metro map can be measured by:

$$S(\mathcal{M}) = \frac{2 \cdot Q(\mathcal{M}) \cdot (1 - R(\mathcal{M}))}{Q(\mathcal{M}) + 1 - R(\mathcal{M})}$$

---

[4] http://www-01.ibm.com/software/commerce/optimization/cplexoptimizer/.

In addition, we can also calculated the total informativeness and fluency of the map by $I(\mathcal{M}) = \sum_{p \in \Pi} I(p)$ and $F(\mathcal{M}) = \sum_{p \in \Pi} F(p)$.

Besides, we also perform an evaluation with human judgments to validate the efficiency of our approach.

4.3.2 Compared methods

We compare our approach **MM4Books** with several state-of-the-art summarization approaches described briefly as follows:

- Random: selects paths from candidate paths $\mathcal{P}_q$ randomly.
- TopRank: selects the top ranked paths according to the score of path $s(p_i)$, where $p_i \in \mathcal{P}_q$.
- TextRank [21]: TR is a graph-based ranking algorithm of deciding the importance of a vertex within a graph based on a modified version of the PageRank algorithm taking edge weights into account. Here, paths in $\mathcal{P}_q$ are considered as the vertices in a graph, and $sim(p_i, p_j)$ is the weight of edge (p_i, p_j) and (p_j, p_i).
- DivRank [20]: DivRank is based on a reinforced random walk in a graph. The model can balance the prestige and the diversity of the top ranked vertices in a principled way. Here, the graph construction is similar to TextRank.
- Spectral Clustering (SC) [19]: SC makes use of the spectrum (eigenvalues) of the similarity matrix of the data to perform clustering. The similarity matrix $S \in R^{|\mathcal{P}_q| \times |\mathcal{P}_q|}$, and $S_{i,j} = \text{sim}(p_i, p_j)$. After clustering, two methods can be applied to select paths: $\mathbf{SC}_{\text{top}}$ and $\mathbf{SC}_{\text{centroid}}$, where $\mathbf{SC}_{\text{top}}$ is to select the top ranked path in each cluster according to the score of path $s(p_i)$, while SC_{centroid} is to select the central path in each cluster.
- DSDR [9]: DSDR can generate a summary which consist of those paths that can best reconstruct the other paths in $\mathcal{P}_q$. There are two variants: $\mathbf{DSDR}_{lin}$ by linear reconstruction, and $\mathbf{DSDR}_{non}$ by nonnegative linear reconstruction.

Besides, we also evaluate our approach with three variants: $\mathbf{MM4Books}_I$, $\mathbf{MM4Books}_F$ and $\mathbf{MM4Book}_Q$ by taking $s(p_i) = I(p_i)$, $s(p_i) = F(p_i)$ and $s(p_i) = I(p_i) \cdot F(p_i)$, respectively.

4.3.3 Experimental results

Table 4 shows the comparison results for the metro map construction with different number of learning paths.

We can see that our approach $\mathbf{MM4Books}_Q$ can achieve the best performance on $S(\mathcal{M})$ against other methods. That is to say, our method can take both the *quality* and *redundancy* of paths into account when constructing the metro map. **TextRank**, **TopRank** and $\mathbf{SC}_{\text{top}}$ tend to select paths with high scores but ignore the redundancy. In particular, for **TextRank**, it ranks the paths based on their prestige in a network, so similar paths are selected which increase the redundancy greatly. In the other hand, **DivRank** and **DSDR** can achieve a comparable *redundancy* $R(\mathcal{M})$, but they have slightly low-quality paths.

Table 4 Comparison results for the metro map construction with different number of learning paths

Method	$Q(\mathcal{M})$	$I(\mathcal{M})$	$F(\mathcal{M})$	$R(\mathcal{M})$	$S(\mathcal{M})$
(a) 5 paths are selected for the map construction					
Random	1.12	2.16	2.28	0.44	0.63
TopRank	2.78	4.06	3.36	0.62	0.62
TextRank	1.69	3.18	2.60	0.83	0.29
DivRank	0.96	1.95	2.38	0.33	0.70
SC_{top}	2.48	3.71	3.28	0.47	0.79
SC_{centroid}	0.94	1.90	2.30	0.45	0.58
$DSDR_{lin}$	1.50	2.94	2.48	0.41	0.77
$DSDR_{non}$	1.52	2.31	3.15	0.60	0.53
$\mathbf{MM4Books}_I$	1.72	3.62	2.29	0.34	0.91
$\mathbf{MM4Books}_F$	1.63	2.09	3.79	0.26	0.89
$\mathbf{MM4Books}_Q$	2.07	3.43	2.97	0.34	**0.95**
(b) 15 paths are selected for the map construction					
Random	3.08	6.19	6.54	0.42	0.88
TopRank	7.25	11.42	9.31	0.55	0.82
TextRank	5.05	9.38	7.80	0.73	0.50
DivRank	3.12	6.43	7.01	0.36	0.93
SC_{top}	6.24	10.32	8.88	0.47	0.92
SC_{centroid}	2.74	5.82	6.74	0.61	0.56
$DSDR_{lin}$	4.33	8.62	7.32	0.40	0.97
$DSDR_{non}$	4.41	7.16	8.90	0.54	0.73
$\mathbf{MM4Books}_I$	2.96	6.79	4.98	0.29	1.07
$\mathbf{MM4Books}_F$	3.02	4.77	7.41	0.26	1.08
$\mathbf{MM4Books}_Q$	3.46	6.37	6.16	0.28	**1.13**

The parameter ϕ influences the quality and redundancy of the metro map. Intuitively, when ϕ decreases, the redun-

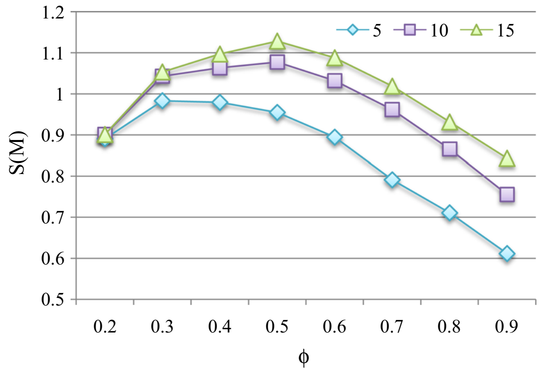
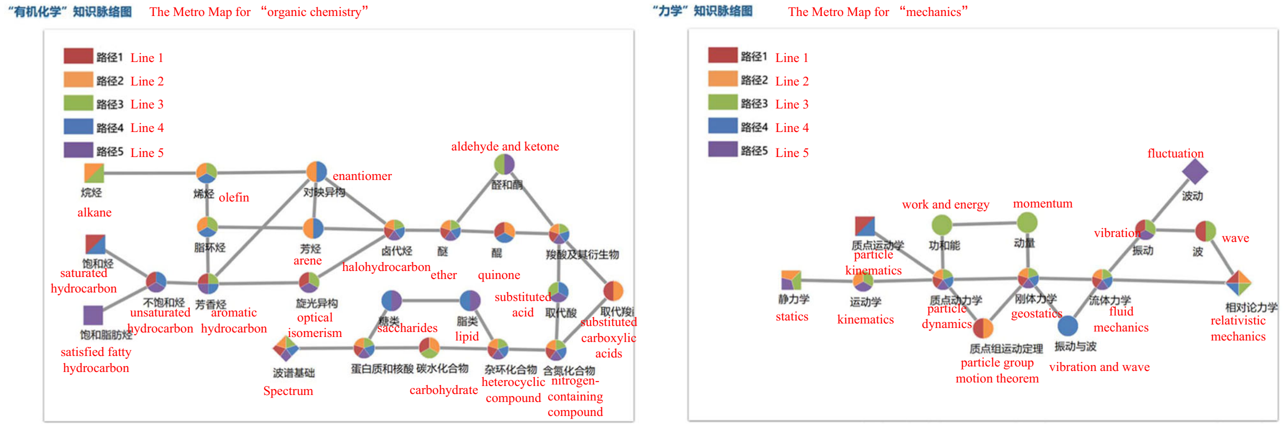
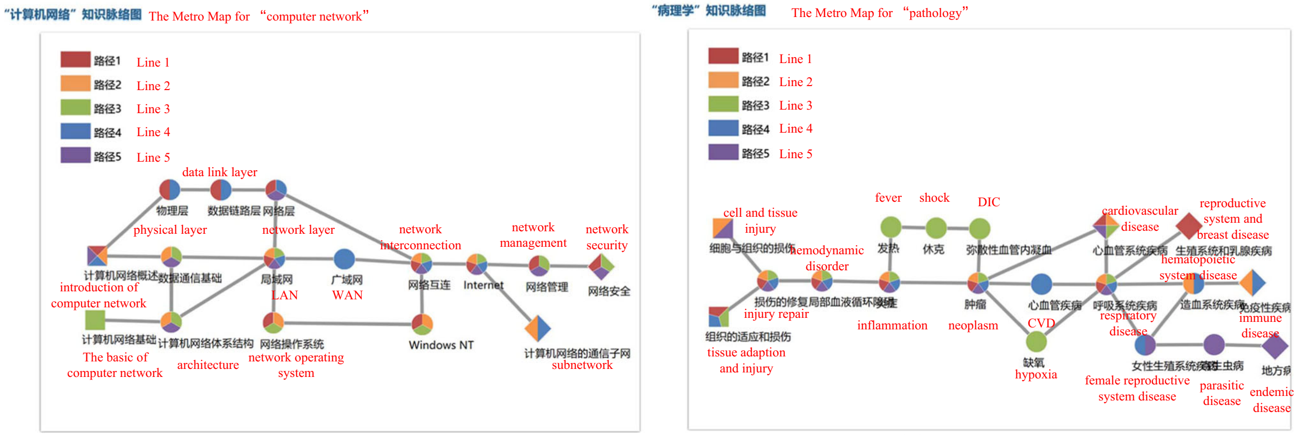


Fig. 3 $S(\mathcal{M})$ changes with different ϕ


"有机化学"知识脉络图 The Metro Map for "organic chemistry"
路径1 Line 1
路径2 Line 2
路径3 Line 3
路径4 Line 4
路径5 Line 5
烷烃 alkane
烯烃 olefin
对映异构 enantiomer
醛和酮 aldehyde and ketone
饱和烃 saturated hydrocarbon
脂环烃
芳烃 arene
卤代烃 halohydrocarbon
醚 ether
醌 quinone
羧酸及其衍生物
不饱和烃 unsaturated hydrocarbon
芳香烃 aromatic hydrocarbon
旋光异构 optical isomerism
取代酸 substituted acid
取代羧酸 substituted carboxylic acids
饱和脂肪烃 satisfied fatty hydrocarbon
糖类 saccharides
脂类 lipid
波谱基础 Spectrum
蛋白质和核酸
碳水化合物 carbohydrate
杂环化合物 heterocyclic compound
含氮化合物 nitrogen-containing compound
"力学"知识脉络图 The Metro Map for "mechanics"
路径1 Line 1
路径2 Line 2
路径3 Line 3
路径4 Line 4
路径5 Line 5
fluctuation
波动
work and energy
momentum
质点运动学 particle kinematics
功和能
动量
vibration
振动
波 wave
静力学 statics
运动学 kinematics
质点动力学 particle dynamics
刚体力学 geostatics
流体力学 fluid mechanics
相对论力学 relativistic mechanics
质点组运动定理 particle group motion theorem
振动与波 vibration and wave

(a) Metro Map for "Organic Chemistry" **(b)** Metro Map for "mechanics"



(c) Metro Map for "Computer Network" **(d)** Metro Map for "pathology"

Fig. 4 Example of metro maps from MM4Books system with different domains: **a** *Chemistry*, **b** *Mechanics*, **c** *Computer Network*, and **d** *Pathology*. In each map, five learning paths are selected to build the metro map, where each path is a sequence of learning objects. Each node represents one learning object, which has a representative title for browsing. Since the books in our library are Chinese, we translate the titles into English for understanding in the Figure. The objects which have more than one color means they are involved into several learning paths. For example, *arene* in the *Chemistry Metro Map* is rendered by two colors, because it involved into *Line 2* and *Line 4*

dancy would reduce, but it also brings down the total quality of paths, since two high-quality but similar paths cannot be selected together. Figure 3 shows the total score $S(\mathcal{M})$ changes with different ϕ, where all three lines, which indicate three different number of paths are selected to construct the metro map, have the similar trajectories. So when $\phi = 0.5$, our method can obtain the best performance.

5 MM4Books system and manual evaluation

5.1 MM4Books system

In order to prove that users can really benefit from the proposed approach when they learn knowledge, we implemented the **MM4Books** system[5] for users by collecting electronic textbooks from our digital library,[6] which contains more than 2.5 million books. When users submit a query for learning, our system will generate and exhibit the metro map automatically.

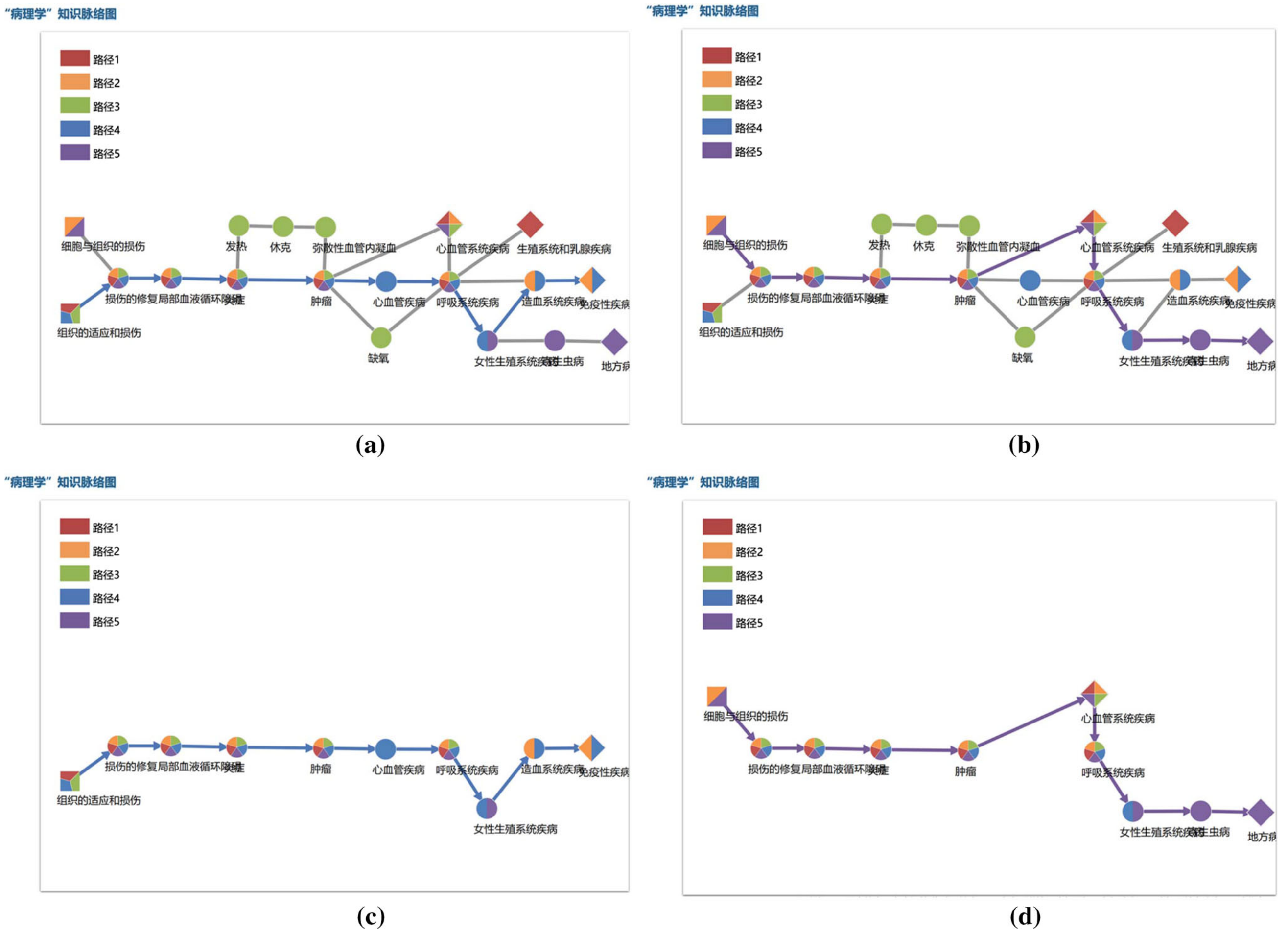
Figure 4 shows some metro map examples generated by our system, which involve different domains such as *Chemistry*, *Mechanics*, *Computer Network* and *Pathology*. In the examples, we select five learning paths to build each metro map, and render the learning object with different color to differentiate the paths. Thus, the learning objects involved into several paths are rendered by several colors. For example, *arene* in the *Chemistry Metro Map* (Fig. 4a) is rendered by two colors, because it involved into *Line 2* and *Line 4*.

When users move his/her mouse on one line, the corresponding learning path could be highlighted as shown in Fig. 5a, b, where *Line 4* and *Line 5* are selected. In order

---

[5] http://www.ckcest.zju.edu.cn/kv.

[6] http://www.cadal.zju.edu.cn.



(a) (b) (c) (d)

Fig. 5 Interaction with the Metro Map. Users can move his/her mouse on a line to highlight it as shown in **a**, **b**, or demonstrate it individually as shown in **c**, **d**

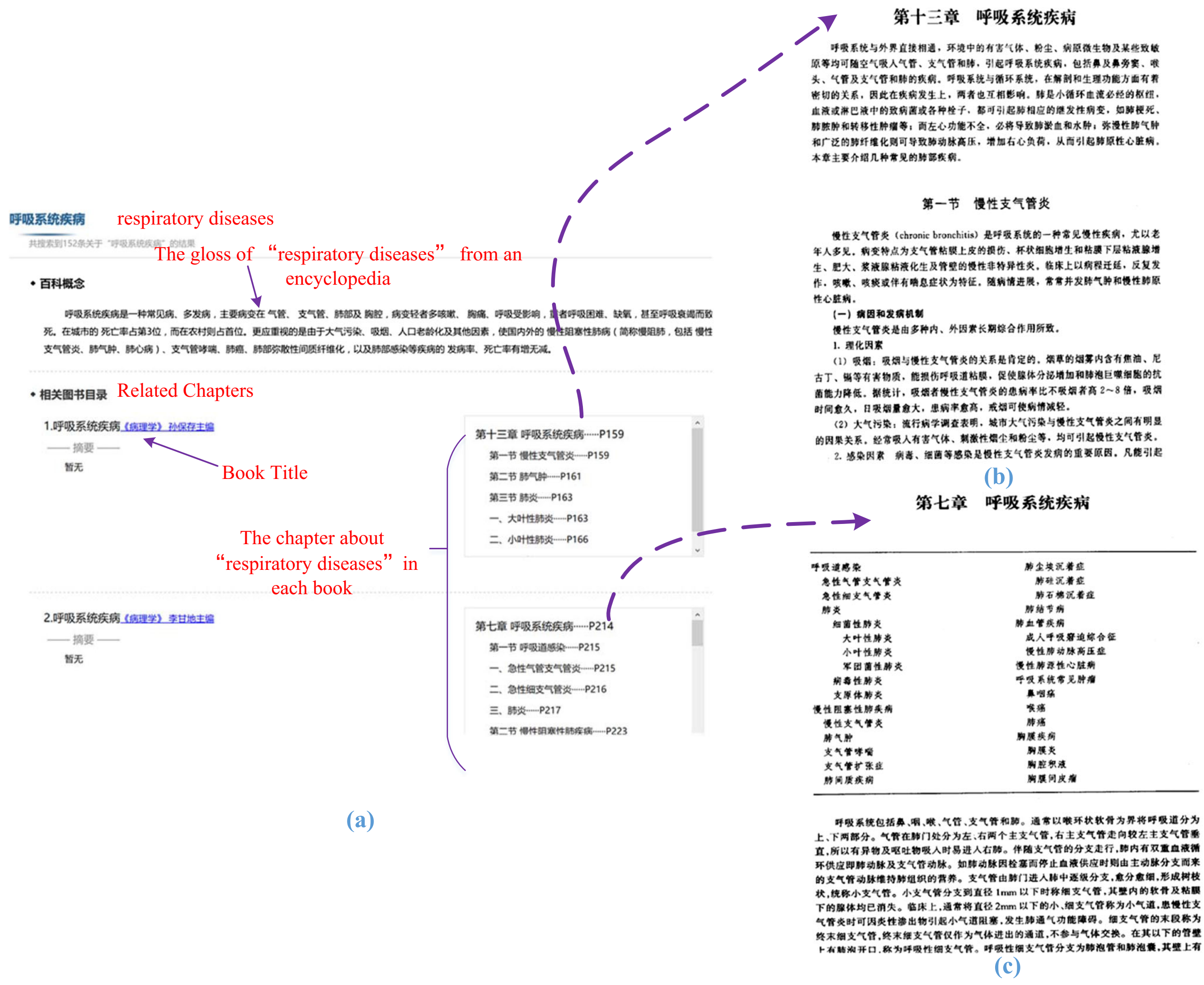
to reduce the interference from other learning paths, each line can be demonstrated individually as shown in Fig. 5c, d. According to Fig. 5c, d, we can see that when learning knowledge about *Pathology*, two learning paths could be followed. We can firstly follow the path: *cell and tissue injury* or (*tissue adaption and injury*) → *injury repair* → *hemodynamic disorder* → ⋯ → *respiratory disease* to learn some basic knowledge, and then learn *parasitic disease* and *endemic disease* or *hematopoietic system disease* and *immune disease* according to different learning paths.

When a user wants to learn one certain knowledge, he/she can just click the corresponding learning object to show the detail as shown in Fig. 6a. For example, Fig. 6 shows the detail about *respiratory disease*, which contains the gloss from an encyclopedia, and a collection of related chapters from different books. Each chapter is shown together with its belonging book, and the contextual chapters in that book, which is convenient for users to read books. Then, the user can learn the knowledge by reading these chapters (Fig. 6b, c).

5.2 Manual evaluation

In this section, we perform a human judgment to evaluate our approach.

We generate the metro maps for each query in different domains by all comparing approaches on the dataset in Table 2, where each metro map contains 5 learning paths as in the MM4Books system (Fig. 4). Then, for each domain, we ask 10 undergraduate students who major in that domain to rate the metro maps for the following two items: (1) map's informativeness about whether the map cover more important learning objects. (2) map's coverage about whether the map has less redundancy among learning paths and covers more learning objects. The ratings range from 1 (lowest) to 10 (highest). During the evaluation, we randomize the maps to avoid any bias. Table 5 shows the results obtained by manual evaluation. From the table, we can see that our approach can generate a more informative and high-coverage metro map than other approaches. In addition, similar to Table 4, the metro maps generated by **TextRank** indeed are low-


respiratory diseases
The gloss of "respiratory diseases" from an encyclopedia
Related Chapters
Book Title
The chapter about "respiratory diseases" in each book
(a)
(b)
(c)

呼吸系统疾病

共搜索到152条关于“呼吸系统疾病”的结果

• 百科概念

呼吸系统疾病是一种常见病、多发病，主要病变在 气管、 支气管、肺部及 胸腔，病变轻者多咳嗽、 胸痛、呼吸受影响，重者呼吸困难、缺氧，甚至呼吸衰竭而致死。在城市的 死亡率占第3位，而在农村则占首位。更应重视的是由于大气污染、吸烟、人口老龄化及其他因素，使国内外的 慢性阻塞性肺病（简称慢阻肺，包括 慢性支气管炎、肺气肿、肺心病）、支气管哮喘、肺癌、肺部弥散性间质纤维化，以及肺部感染等疾病的 发病率、死亡率有增无减。

• 相关图书目录

1.呼吸系统疾病《病理学》 孙保存主编

—— 摘要 ——

暂无

第十三章 呼吸系统疾病……P159
第一节 慢性支气管炎……P159
第二节 肺气肿……P161
第三节 肺炎……P163
一、大叶性肺炎……P163
二、小叶性肺炎……P166

2.呼吸系统疾病《病理学》 李甘地主编

—— 摘要 ——

暂无

第七章 呼吸系统疾病……P214
第一节 呼吸道感染……P215
一、急性气管支气管炎……P215
二、急性细支气管炎……P216
三、肺炎……P217
第二节 慢性阻塞性肺疾病……P223

第十三章　呼吸系统疾病

呼吸系统与外界直接相通，环境中的有害气体、粉尘、病原微生物及某些致敏原等均可随空气吸人气管、支气管和肺，引起呼吸系统疾病，包括鼻及鼻旁窦、喉头、气管及支气管和肺的疾病。呼吸系统与循环系统，在解剖和生理功能方面有着密切的关系，因此在疾病发生上，两者也互相影响。肺是小循环血流必经的枢纽，血液或淋巴液中的致病菌或各种栓子，都可引起肺相应的继发性病变，如肺梗死、肺脓肿和转移性肿瘤等；而左心功能不全，必将导致肺淤血和水肿；弥漫性肺气肿和广泛的肺纤维化则可导致肺动脉高压，增加右心负荷，从而引起肺原性心脏病。本章主要介绍几种常见的肺部疾病。

第一节　慢性支气管炎

慢性支气管炎（chronic bronchitis）是呼吸系统的一种常见慢性疾病，尤以老年人多见。病变特点为支气管粘膜上皮的损伤、杯状细胞增生和粘膜下层粘液腺增生、肥大、浆液腺粘液化生及管壁的慢性非特异性炎。临床上以病程迁延，反复发作，咳嗽、咳痰或伴有喘息症状为特征。随病情进展，常常并发肺气肿和慢性肺原性心脏病。

（一）病因和发病机制

慢性支气管炎是由多种内、外因素长期综合作用所致。

1. 理化因素

（1）吸烟：吸烟与慢性支气管炎的关系是肯定的。烟草的烟雾内含有焦油、尼古丁、镉等有害物质，能损伤呼吸道粘膜，促使腺体分泌增加和肺泡巨噬细胞的抗菌能力降低。据统计，吸烟者慢性支气管炎的患病率比不吸烟者高 2～8 倍，吸烟时间愈久，日吸烟量愈大，患病率愈高，戒烟可使病情减轻。

（2）大气污染：流行病学调查表明，城市大气污染与慢性支气管炎之间有明显的因果关系。经常吸入有害气体、刺激性烟尘和粉尘等，均可引起慢性支气管炎。

2. 感染因素　病毒、细菌等感染是慢性支气管炎发病的重要原因。凡能引起

第七章　呼吸系统疾病

呼吸道感染
急性气管支气管炎
急性细支气管炎
肺炎
细菌性肺炎
大叶性肺炎
小叶性肺炎
军团菌性肺炎
病毒性肺炎
支原体肺炎
慢性阻塞性肺疾病
慢性支气管炎
肺气肿
支气管哮喘
支气管扩张症
肺间质疾病
肺尘埃沉着症
肺硅沉着症
肺石棉沉着症
肺结节病
肺血管疾病
成人呼吸窘迫综合征
慢性肺动脉高压症
慢性肺源性心脏病
呼吸系统常见肿瘤
鼻咽癌
喉癌
肺癌
胸膜疾病
胸膜炎
胸腔积液
胸膜间皮瘤

呼吸系统包括鼻、咽、喉、气管、支气管和肺。通常以喉环状软骨为界将呼吸道分为上、下两部分。气管在肺门处分为左、右两个主支气管,右主支气管走向较左主支气管垂直,所以有异物及呕吐物吸入时易进入右肺。伴随支气管的分支走行,肺内有双重血液循环供应即肺动脉及支气管动脉。如肺动脉因栓塞而停止血液供应时则由主动脉分支而来的支气管动脉维持肺组织的营养。支气管由肺门进入肺中逐级分支,愈分愈细,形成树枝状,统称小支气管。小支气管分支到直径 1mm 以下时称细支气管,其壁内的软骨及粘膜下的腺体均已消失。临床上,通常将直径 2mm 以下的小、细支气管称为小气道,患慢性支气管炎时可因炎性渗出物引起小气道阻塞,发生肺通气功能障碍。细支气管的末段称为终末细支气管,终末细支气管仅作为气体进出的通道,不参与气体交换。在其以下的管壁上有肺泡开口,称为呼吸性细支气管。呼吸性细支气管分支为肺泡管和肺泡囊,其壁上有

Fig. 6 Example of a learning object *respiratory disease*, which contains the gloss from an encyclopedia, and a collection of related chapters from different books as shown in **a**. Users can click each chapter to show its content for knowledge learning as shown in **b**, **c**

Table 5 Manual evaluation by 10 evaluators on *Informativeness*(**Inf**) and *Coverage*(**Cov**) of metro map

Method	Inf	Cov	Method	Inf	Cov
TopRank	7.14	6.15	TextRank	7.50	6.38
DivRank	6.95	6.75	SC_{top}	7.12	6.58
SC_{centroid}	7.23	6.68	$DSDR_{\text{lin}}$	7.51	7.19
$DSDR_{\text{non}}$	7.65	6.93	**MM4Books**$_Q$	7.71	7.67

coverage, and the performance of **DSDR**$_{lin}$ is close to our approach.

6 Conclusion

In this paper, we proposed an approach to generate metro maps for efficient knowledge learning by summarizing massive electronic textbooks. To the best of our knowledge, it is the first work to address this task. We acquired the learning objects by clustering the semantically similar chapters via an unsupervised clustering method, and then built the metro map by applying an ILP-based technique to select a collect of high informative and fluent but low redundant learning paths from a learning graph. Experiments show that our proposed approach outperforms all the state-of-the-art baseline approaches, and the human evaluation on the **MM4Books** system also proves that users can really benefit from the proposed approach for knowledge learning.

There are still some limitations of our approach. First, the method for learning object acquirement is simple and heuristic. Second, the human evaluation is relatively simple, which just takes *informativeness* and *coverage* into accounts. In the future, we plan to introduce knowledge graph to promote the learning object acquirement and try to apply graph convolutional network on the learning graph to build metro maps. In addition, we would like to propose more metrics based on educational theory for metro maps, and make further experiments on human evaluation to make sure that our metro map can indeed boost the learning efficiency.

Acknowledgements This work is supported by the Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST), the Fundamental Research Funds for the Central Universities (No. 2017FZA5016), and MOE-Engineering Research Center of Digital Library.

References

1. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Enriching textbooks with images. In: CIKM (2011)
2. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Data mining for improving textbooks. ACM SIGKDD Explor. Newsl. **13**(2), 7–19 (2012)
3. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Study navigator: an algorithmically generated aid for learning from electronic textbooks. In: EDM (2014)
4. Chen, Z., Zhang, X., Boedihardjo, A.P., Dai, J., Lu, C.T.: Multimodal storytelling via generative adversarial imitation learning. In: IJCAI (2017)
5. Csomai, A., Mihalcea, R.: Linking educational materials to encyclopedic knowledge. In: AIED (2007)
6. Dou, W., Yu, L., Wang, X., Ma, Z., Ribarsky, W.: Hierarchicaltopics: visually exploring large text collections using topic hierarchies. IEEE Trans. Vis. Comput. Graph. **19**, 2002–2011 (2013)
7. Filippova, K.: Multi-sentence compression: finding shortest paths in word graphs. In: COLING (2010)
8. Gillies, J., Quijada, J.J.: Opportunity to learn: a high impact strategy for improving educational outcomes in developing countries. Working Paper. Academy for Educational Development (2008)
9. He, Z., Chen, C., Bu, J., Wang, C., Zhang, L., Cai, D., He, X.: Document summarization based on data reconstruction. In: AAAI (2012)
10. Hu, B., Lu, Z., Li, H., Chen, Q.: (2014a) Convolutional neural network architectures for matching natural language sentences. In: NIPS
11. Hu, P., Huang, M., Zhu, X.: Exploring the interactions of storylines from informative news events. J. Comput. Sci. Technol. **29**, 502–518 (2014b)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: ACL (2014)
13. Kenter, T., de Rijke, M.: Short text similarity with word embeddings. In: CIKM (2015)
14. Kokkodis, M., Kannan, A., Kenthapadi, K.: Assigning educational videos at appropriate locations in textbooks. In: EDM (2014)
15. Larranaga, M., Conde, A., Calvo, I., Elorriaga, J.A., Arruarte, A.: Automatic generation of the domain module from electronic textbooks: method and validation. IEEE Trans. Knowl. Data Eng. **26**(1), 69–82 (2014)
16. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)
17. Liang, C., Wang, S., Wu, Z., Williams, K., Pursel, B., Brautigam, B., Saul, S., Williams, H., Bowen, K., Giles, C.L.: Bbookx: an automatic book creation framework. In: Proceedings of the 2015 ACM Symposium on Document Engineering, pp 121–124. ACM (2015)
18. Lu, Z., Li, H.: A deep architecture for matching short texts. In: NIPS (2013)
19. von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
20. Mei, Q., Guo, J., Radev, D.R.: Divrank: the interplay of prestige and diversity in information networks. In: KDD (2010)
21. Mihalcea, R., Tarau, P.: Textrank: bringing order into texts. In: EMNLP (2004)
22. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: AAAI (2016)
23. Shahaf, D., Guestrin, C., Horvitz, E.: Metro maps of science. In: KDD (2012a)
24. Shahaf, D., Guestrin, C., Horvitz, E.: Trains of thought: generating information maps. In: WWW (2012b)
25. Sigurdsson, G.A., Chen, X., Gupta, A.: Learning visual storylines with skipping recurrent neural networks. In: ECCV (2016)
26. Tang, S., Wu, F., Li, S., Lu, W., Zhang, Z., Zhuang, Y.: Sketch the storyline with charcoal: a non-parametric approach. In: IJCAI (2015)
27. Tran, T.A., Niederée, C., Kanhabua, N., Gadiraju, U., Anand, A.: Balancing novelty and salience: adaptive learning to rank entities for timeline summarization of high-impact events. In: CIKM (2015)
28. Wang, D., Li, T., Ogihara, M.: Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In: AAAI (2012)
29. Wang, L., Cardie, C., Marchetti, G.: Socially-informed timeline generation for complex events. In: HLT-NAACL (2015a)
30. Wang, S., Liang, C., Wu, Z., Williams, K., Pursel, B., Brautigam, B., Saul, S., Williams, H., Bowen, K., Giles, C.L.: Concept hierarchy extraction from textbooks. In: Proceedings of the 2015 ACM Symposium on Document Engineering, pp. 147–156. ACM (2015b)
31. Wang, S., Ororbia, A., Wu, Z., Williams, K., Liang, C., Pursel, B., Giles, C.L.: (2016) Using prerequisites to extract concept maps from textbooks. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 317–326. ACM
32. Wang, Z., Shou, L., Chen, K., Chen, G., Mehrotra, S.: On summarization and timeline generation for evolutionary tweet streams. IEEE Trans. Knowl. Data Eng. **27**, 1301–1315 (2015c)
33. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. CoRR arXiv:1702.03814 (2017)
34. Wu, Y., Wu, W., Li, Z., Zhou, M.: Response selection with topic clues for retrieval-based chatbots. arXiv:160500090 (2016)
35. Wu, Z., Li, Z., Mitra, P., Giles, C.L.: Can back-of-the-book indexes be automatically created? In: CIKM (2013)
36. Yang, S., Lu, W., Yang, D., Li, X., Wu, C., Wei, B.: Keyphraseds: automatic generation of survey by exploiting keyphrase information. Neurocomputing **224**, 58–70 (2017)
37. Yu, S., Li, X., Zhao, X., Zhang, Z., Wu, F.: Tracking news article evolution by dense subgraph learning. Neurocomputing **168**, 1076–1084 (2015)
38. Zhang, L., Li, L., Li, T., Zhang, Q.: Patentline: analyzing technology evolution on multi-view patent graphs. In: SIGIR (2014)
39. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: CIKM (2002)
40. Zhou, D., Xu, H., He, Y.: An unsupervised Bayesian modelling approach for storyline detection on news articles. In: EMNLP (2015)

41. Zhou, D., Xu, H., Dai, X.Y., He, Y.: Unsupervised storyline extraction from news articles. In: IJCAI (2016)
42. Zhu, X., Ming, Z., Zhu, X., Chua, T.S.: Topic hierarchy construction for the organization of multi-source user generated contents. In: SIGIR (2013)
43. Zhu, X., Ming, Z., Hao, Y., Zhu, X., Chua, T.S.: Customized organization of social media contents using focused topic hierarchy. In: CIKM (2014)