information

MDPI

Article

A Web Page Clustering Method Based on Formal Concept Analysis

Zuping Zhang *, Jing Zhao and Xiping Yan

School of Information Science and Engineering, Central South University, Changsha 410000, China; ajingzhao@foxmail.com (J.Z.); 954160193@foxmail.com (X.Y.)

* Correspondence: zpzhang@csu.edu.cn; Tel.: +86-731-88879628

Received: 21 August 2018; Accepted: 2 September 2018; Published: 6 September 2018

Abstract: Web page clustering is an important technology for sorting network resources. By extraction and clustering based on the similarity of the Web page, a large amount of information on a Web page can be organized effectively. In this paper, after describing the extraction of Web feature words, calculation methods for the weighting of feature words are studied deeply. Taking Web pages as objects and Web feature words as attributes, a formal context is constructed for using formal concept analysis. An algorithm for constructing a concept lattice based on cross data links was proposed and was successfully applied. This method can be used to cluster the Web pages using the concept lattice hierarchy. Experimental results indicate that the proposed algorithm is better than previous competitors with regard to time consumption and the clustering effect.

Keywords: formal concept analysis; feature weight; cross linked list; concept lattice; Web page clustering

1. Introduction

In the era of big data, due to the massive amount of data in the network, the difficulty of getting the correct information is greatly increased. More and more people are beginning to realize the importance of network resource standardization and reorganization. Web clustering technology has attracted a lot of attention as an important part of network resource management. It is mainly used in the fields of data extraction, knowledge discovery, data mining, and information fusion [1]. Similar Web pages are grouped into one category by extracting and clustering the similarity of these Web pages. Because of the large capacity and dynamic nature of the network, Web page clustering is more difficult and complicated than ordinary text clustering [2]. At present, most studies on Web page clustering have been based on text, link structure or hybrid clustering. Oren Etzioni and Oren Zamir [3] explored the clustering of Web pages, and found that users are often forced to screen the list of files returned by the search engine, which makes it hard to retrieve information. They then proposed a new algorithm to construct suffix trees based on the Web abstract by studying the clustering of Web pages; the clustering was more effective when using this algorithm. In addition, Mecca et al. [4] proposed a new Web page clustering algorithm in 2006, which not only directly analyzes the results returned by search engines, but also obtains the entire Web page by analyzing the fragment content of the results.

The Formal Concept Analysis (FCA) represents the relationship between objects and attributes, and is used widely in data mining, e-commerce, personalized navigation, text clustering, semantic Web, search engines, and so on. Sun introduced an information management system for data mining by integrating ontology and FCA in e-commerce [5]. Hitzler applied FCA in a semantic web [6]. Shen et al. proposed an intelligent search engine in which the information retrieval model is based on the formal context of FCA and is integrated with a browsing mechanism for such a system based on the concept lattice [7]. Freeman and White suggested that network data could take the form of a square-actor in an actor-binary adjacency matrix, where each row and each column in the matrix represents a social actor.

They used a Galois Lattice to represent network data [8]. Although there are many algorithms for FCA clustering with larger feature sets and larger data sets, the results are not satisfactory. Clustering based on FCA still has some limitations. For example, there are too many concepts; the scale of the concept lattice is too large; the computation is large and complex; and the process of generating a complete Hasse diagram is difficult.

The purpose of this paper is to study a web page clustering method based on formal concept analysis. The contributions of this paper are as follows:

- Propose a bi-directional maximum matching word segmentation method based on a Chinese word segmentation method and propose a word weighting scheme based on term frequency.
- Propose a ListFCA algorithm based on the study of different types of concept lattice construction algorithms. The ListFCA uses a more efficient method to traverse the header list based on cross data link when constructing concept lattices, which makes it easier to generate the Hasse graph corresponding to the concept lattice.

The rest of the paper is organized as follows. The related works and the classic algorithm for constructing the concept lattice is discussed in Section 2. In Section 3, we preprocess the Web page and propose a more efficient header list traversal method for the construction of concept lattices. The experiments are shown in Section 4 to verify the efficiency of the ListFCA. The conclusion of the paper is in Section 5.

2. Related Work

2.1. The Background of FCA

In the 1980s, Wille proposed FCA and introduced its mathematical definition and related theories [9]. In this method, the binary relationship between objects and attributes is analyzed, the formal context is constructed, and finally a concept lattice is constructed based on the formal context. Due to the effective description of data and data relationships, FCA has become a powerful tool in rule extraction and data analysis.

FCA is a formalized description of concepts (i.e., an ontology). The intension of a concept is the set of attributes belonging to the concept, and the extension of a concept is the set of objects that contain the concept [10]. In FCA, we use all the attributes and objects to construct formal contexts, and these contexts are usually in the form of a cross table. In this cross table, objects are represented by rows, and attributes are represented by columns, while the intersection of the mth row and the nth column represents the mth object with the nth attribute [11,12]. Clustering based on the FCA method contains four steps. First, the data sets are pretreated and a feature item is extracted from a data element, which is represented by a feature item. Then, a formal context is constructed by taking data elements as objects and feature items as attributes. Next, the concept of formal contexts is extracted, and the concept lattice of the formal contexts is constructed. Finally, a Hasse diagram with a high degree of visualization can be generated.

2.2. Related Algorithms for Constructing Concept Lattice

Among FCA-related applications, the core content is constructing a formal context and then generating the corresponding concept lattice. The algorithms for constructing the concept lattice are mainly divided into three types: batch construction algorithm, incremental construction algorithm and parallel construction algorithm [13].

2.2.1. The Batch Algorithm

The Batch algorithm is a kind of early construction algorithm. Based on the different ways of constructing the lattice, they can be divided into three categories, including top-down, bottom-up, and enumeration [14]. In the top-down method, the top node is constructed first, and then it proceeds

downwards layer by layer. The Bordat algorithm [15] is one of the classic algorithms. In the bottom-up method, the bottom node is constructed first, and then it expands upward, such as in the Chein algorithm [16]. The enumeration method is based on a given data set, and all nodes are enumerated according to a certain order, such as Ganter algorithm [10]. The father-child relationships between concepts and all the concept lattices can be obtained in the batch algorithm. However, it is difficult to generate Hasse diagram.

We introduce the Bordat algorithm [15] as a proposition. At first, the top node concept is constructed which has empty intension or has only one element, and its extension is all sets of object in formal context. Secondly, all the sub nodes of the top node are calculated, each of their direct sub nodes is also calculated one by one. Finally, these steps are repeated until there are no sub nodes.

2.2.2. The Incremental Algorithm

In the Incremental algorithm, the lattices are computed by adding objects or attributes of a given context one by one. Godin algorithm is one of the classic algorithms. The incremental algorithm can avoid redundant nodes and work with dynamic datasets. It has the ability to construct concept lattices for dynamic formal context. Because of the superior temporal performance, most of the concept lattice systems are based on this method.

We introduce the Godin algorithm as a proposition [17]. Godin algorithm scans every node in the original concept lattice and estimates which category the node belonged, then the corresponding operation is taken according to different categories.

2.2.3. The Parallel Algorithm

The Parallel algorithm falls into two categories. The first one involves the partition of the formal context, so that the sub context can be processed in parallel when constructing concept lattice. The other one involves improving serial algorithm, so that it can be parallelized. In the Parallel algorithm, the complex parent-child relationships between concepts can be obtained, and the concept of dynamic formal context can be extracted. It has great advantages in dealing with complex and huge data sets. However, in most parallel construction algorithms it is difficult to produce a complete lattice structure, and Hasse diagram cannot be generated. Therefore, it is not suitable for the study in this paper.

3. The ListFCA Algorithm

3.1. Web Data Cleaning

Before constructing the concept lattice, the Web page needs to be preprocessed. The Web page generally contains a large number of HTML tags in the source file. The HTML tags can provide a lot of information and structure information of the Web pages, but there is much useless information in the HTML tags.

3.1.1. Disturbance Cleaning

The organization and format of Web content are different from the plain text. The text content of Web page does not exist alone in the Web source file [18]. Instead, the text content is nested in the HTML tags. The different contents can be added with different tags so that it can be displayed in different styles. Through the analysis of the Web page contents, we find that some contents have no relation to the content of Web page itself [19]. Those contents are useless for clustering even increases the complexity of the algorithm and becomes interference. The interference generally covers most of the content except the text, such as various pictures, advertisements, borders and so on. More of the interference come from the source codes, include some page rule information, page style information, page function information etc. Removing the interference is necessary. Different contents are usually wrapped in different tags in the source file, which is very helpful for the identification of

interference [20]. The interference exists in the interference item tags, and the information in the tags should be filtered out.

Through several Web page analyses, we have developed the interference item tags shown in Table 1 [21].

Table 1. Tag of Interference Item.

Tag
<ALT>
<IMG>
<TABLE>
<STYLE>
<APPLET>
<FORM>
<META>
<SCRIPT>
<LINK>

3.1.2. Tag Weighting

The tags appear in pairs in Web pages. After removing the interference in the Web page, the unfiltered tags are different in content. Through the analysis of several Web instances, the contents of different tags have different levels of importance. Thus, the different weights should be given to the tags. In this way, the contents of the tag have the same weight as the tag, and the feature words are contained in the corresponding contents [22]. For example, the <TITLE> tags have higher weight values while the tags such as <H4> are not important and have lower weight values. In this section, a higher frequency tag is developed in the Web page and the relative value of the corresponding weight is shown in Table 2 [23].

Table 2. Weighting of Page Tag.

Tag	Weight	Tag	Weight
<TITLE>	5	<H2>	3
<EM>	4	<H3>	2
<STRONG>	4	<H4>	2
<B>	3	<DT>	3
<BIG>	3	<A>	3
<H1>	4	<UL><OL>	3

Web page preprocessing is an important foundation work in Web clustering. A Web page is represented as a set of feature words through preprocessing and is very useful for simplifying Web page clustering. We need to segment the Web pages, calculate the word weight and extract the feature in preprocessing.

In this paper, we used the bi-direction maximum matching word segmentation method [24]. The rules are as follows:

The algorithm has an ability of eliminating ambiguities, and it can also effectively solve the flaws of the segmentation algorithm based on dictionary in ambiguity recognition, which ensures the speed of word segmentation.

- If the number of the result words in forward or the reverse maximum matching is different, then take the fewer words as the correct result;
- If the number of the result words is the same, there is no ambiguity, then return to any one as a result.

After segmenting Web page into words and removing stop words, the term frequency (TF) was used as the primary basis to calculate the weights. The formula for calculating term frequency is as follows:

$$tf_k = \sum_{j \in J} \beta_j \times tf_{kj} \tag{1}$$

where tf_k indicates the term frequency of the kth character in a Web page; J represents a collection of label categories; β_j represents the weight of j class label; tf_{kj} represents the term frequency of the feature word k in the j class label.

The term weight uses normalization method [25]. The formula is as follows:

$$w(t_k) = \frac{tf_k}{\sum_{n=1}^{N} tf_n} \tag{2}$$

where N represents the number of term frequencies; $\sum_{n=1}^{N} tf_n$ represents the total term frequency of a Web page; $w(t_k)$ represents the kth term weight in this Web page.

Since the word weight has been calculated, we could sort the feature words by weight in descending order in feature extraction and select the top m feature words with maximum weight values as the dimension of the Web feature vector to represent the Web page.

3.2. Web Formal Context Construction

According to the related knowledge of formal context, we could define the formal contexts of multiple Web pages:

Definition 1. *For a triple $K = (G, M, I,)$, where $G = \{g_1, g_2, g_3, \ldots g_i\}$ denotes a set of objects; $M = \{m_1, m_2, m_3, \ldots m_j\}$ denotes a set of attributes; and $I \in G \times M$ is a binary relation between these two sets; where the symbol $\times$ denotes Cartesian product. Then the triple K is defined as the formal context of the Web page* [26,27].

Based on the definition of the formal context of the Web page given above as well as the formal context definition in formal concept analysis, set P is the collection of Web pages, set M is the collection of feature words that can represents Web pages. For any Web page $p \in P$ and any feature words $m \in M$, if Web page p contains feature m, then $(p,\ m) \in R$. For example, we set the formal context as $K = (P, M, R)$, $P = (1, 2, 3, 4)$ as the collection of Web page, $M = (m_1, m_2, m_3, m_4, m_5)$ as the collection of feature words.

Table 3 shows a simple Web formal context. In the table, a single Web page is represented by a row and a feature word is represented by a column [28]. The "×" at the intersection between rows and columns indicates that the Web page of this line has the feature word of this column. The Hasse diagram corresponding to the formal context can be obtained as shown in Figure 1. we number the concepts in the concept lattice of Figure 1 as ① to ⑧.

Table 3. Weighting of Page Tag.

	m_1	m_2	m_3	m_4	m_5
1		×	×		×
2	×	×	×		×
3			×	×	
4	×				×

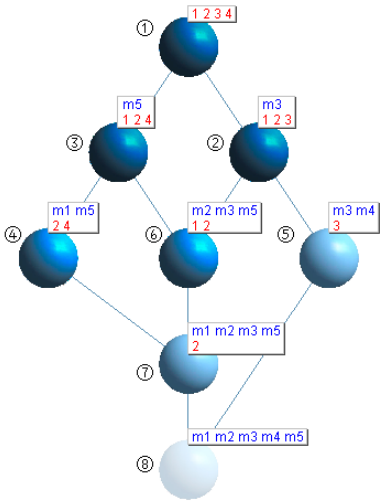

Figure 1. The Hasse graph corresponding to Table 3.

In Web page clustering studies, in order to emphasize the commonalities of the Web pages, the feature words with high frequency in Web documents should be highlighted. In addition, when the number of pages is large, if a feature word appears only on one page, the word should be considered a special case and not be chosen as an attribute for clustering. When building the formal context of Web pages, we should first filter out the words that appear only on one page (the frequency of the page $df_k = 1$; when the number of pages continue to increase, we can consider enlarging filtration threshold df_k). Thus, the purpose of simplifying the concept lattice can be achieved.

3.3. The Construction Process of the Cross Data Link

As is shown below in Figure 2, a node consists of four fields: row, col, down, and right. The row field is used to store the superconcept. The col field is used to store the subconcept. The down field points to the next node of this column. The down field of the last concept node in each column is empty, which indicates the end of the column. In addition, the right field points to the next concept node in the line, the right field of last concept node in each line is also empty, indicating the end of the line.

row	col
down	right

Figure 2. Node structure in the cross data link.

The construction process of the cross data link is as follows:

- Step 1. The concept preprocessing: We iterate through the concepts, and sort them in ascending order based on their number of intensions. If the number of intensions is the same, we sort the concepts by lexicographic order.
- Step 2. Creating the link header: We create a new node based on the definition of the node structure and empty the row field and col field of the node.
- Step 3. Creating the row headers and the column headers: All the col field in row header nodes and all the row fields in column header nodes are emptied. For each row, the row field of ith header node stores the ith concept and for each column, the col field of ith header node stores the ith concept. The down field of the $i-1$th row header node points to the ith row header node, and the last down field is empty. The right field of $i-1$th column header node points to the ith line node, and the last right field is empty.

- Step 4. Constructing nodes based on subconcept-superconcept relations: If the superconcept is numbered as x and the subconcept is numbered as y, we can insert a new node in the xth row and the yth list. Then the superconcept is stored in the row field of the new node, and the subconcept is stored in the col field. The down field of the new node points to the next node in this column and the right field points to the next node in this row. If it does not have a next node, we empty its down field or right field.

As an example of the concept in the formal context in Table 3, a cross data link is created with the above method (the "0" indicates that the row field or the col field is empty).

- Step 1. Ordering the eight concepts in Figure 1 based on the number of intentions as ① to ⑧.
- Step 2. Creating the total header and setting the row field and the col field to "0".
- Step 3. Creating the row headers and list heads; all col field of the row headers are set to "0", the row field stores concepts from ① to ⑧, the down field points to the next node, and the last one is empty. All the row fields of the column headers are set to "0", the col field stores concepts from ① to ⑧, the right field points to the next node, and the last one is also empty.
- Step 4. Inserting the concepts in Figure 1. Concepts ② and ③ are direct subconcepts of the concept ①. A node is inserted at the first row and the second column. Concept ① is stored in the row field and concept ② is stored in the col field in this node. Then a node is inserted in the first row and the third column. Concept ① is stored in the row field and concept ③ is stored in the col field in this node. The right field and the down field point to the next node. This process is repeated for all the other nodes.

Through the above four steps, we can get the cross data link, as shown in Figure 3.

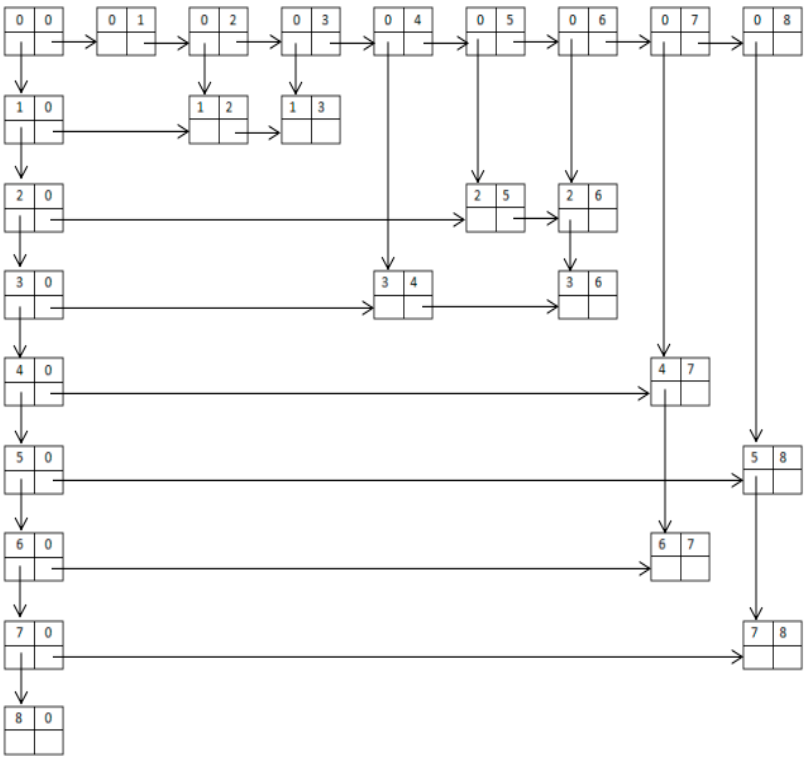

Figure 3. The cross data link corresponding to Table 3.

3.4. Construction Algorithm of the Concept Lattice Based on a Cross Data Link

We introduce an algorithm for constructing the concept lattice based on a cross data link and take the added object as an example. We iterate through the column header, judge the type of concept in the concept lattice, and then perform different operations.

Definition 2. *For a formal context $K = (P, M, R)$, the corresponding concept lattice is $L(K)$, if a new concept p is added, based on the relationship between the concept in $L(K)$ and the attributes $\{p\}'$ of the new object, the types of the concepts stored in the col field of the column header will be divided into three categories:*

1. *For a concept $(A,\ B) \in L(K)$, if $B \subseteq \{p\}'$, the concept is an updating concept, and the extension of this concept will be updated.*
2. *For a concept $(A,\ B) \in L(K)$, if $B \neq \varnothing$ and $B \cap \{p\}' = \varnothing$, the concept is an invariant concept, and no operation is required.*
3. *For a concept $(A,\ B) \in L(K)$, if $B \cap \{p\}' = H \neq \varnothing$ and the intension of any concept in $L(K)$ not equal to H and for any superconcept $(A_2,\ B_2)$ of concept $(A_1,\ B_1)$, $B_2 \cap \{p\}' \neq H$ is established, the concept is a generation concept, and the new concept will be created.*

If the concept is a generation concept, the following operations are performed:

- Step 1. Insert a new header node at the end of the row header and the column header. The row field of the new row header node and the col field of the new column header node are set to be the new concept.
- Step 2. Insert the new node at this column and the new line. The col field of the new node is consistent with the col field of this column, and the new concept is stored in the row field in the new node.
- Step 3. We need to determine the relationship between the direct superconcept set A in the col field in the new node and the direct superconcept set B in the new concept. The results are divided into two cases: one is where set A has an intersection set C with set B. In this situation, we delete the nodes in this column where the row field concept belongs to set C. Another situation is where set A and set B have no intersection, in which case no operation is required.
- Step 4. We simply insert new nodes based on set B; the number of new nodes is equal to the size of set B. The row field in the new nodes stores each concept in set B, and the col field in the new nodes stores the new concepts.

In accordance with the above algorithm, we add a new object "5" in Figure 1, with attributes m_4 and m_5. This is represented by $\{5, (m_4, m_5)\}$. We first iterate through the column header storing the concepts from ① to ⑧. The concepts are divided into 3 types in accordance with Definition 2.

1. Concepts ②, ④, ⑥, ⑦ are invariant concepts. No operation is required.
2. Concepts ①, ③ are updating concepts. We add the new object "5" to the extension of these concepts.
3. Concepts ⑤, ⑧ are generation concepts. We create the new concepts ⑨ and ⑩ and, taking concept ⑨ as an example, the following operations are performed:

    - Step 1. Insert a new header node at the end of row header and column header.
    - Step 2. Insert a new node at the 5th column and the 9th row. Concept ⑤ is stored in the col field of the new node and concept ⑨ is stored in the row field of the new node.
    - Step 3. The relationship between the direct superconcept of concept ⑤ and concept ⑨ have no intersection.
    - Step 4. Insert a new node at the 9th column and the 1st row, because the direct superconcept of concept ⑨ is concept ①. Concept ⑨ is sotred in the col field of the new node and concept ⑤ is stored in the row field of the new node.

The new cross data link is shown in Figure 4, the new nodes are indicated by bold boxes.

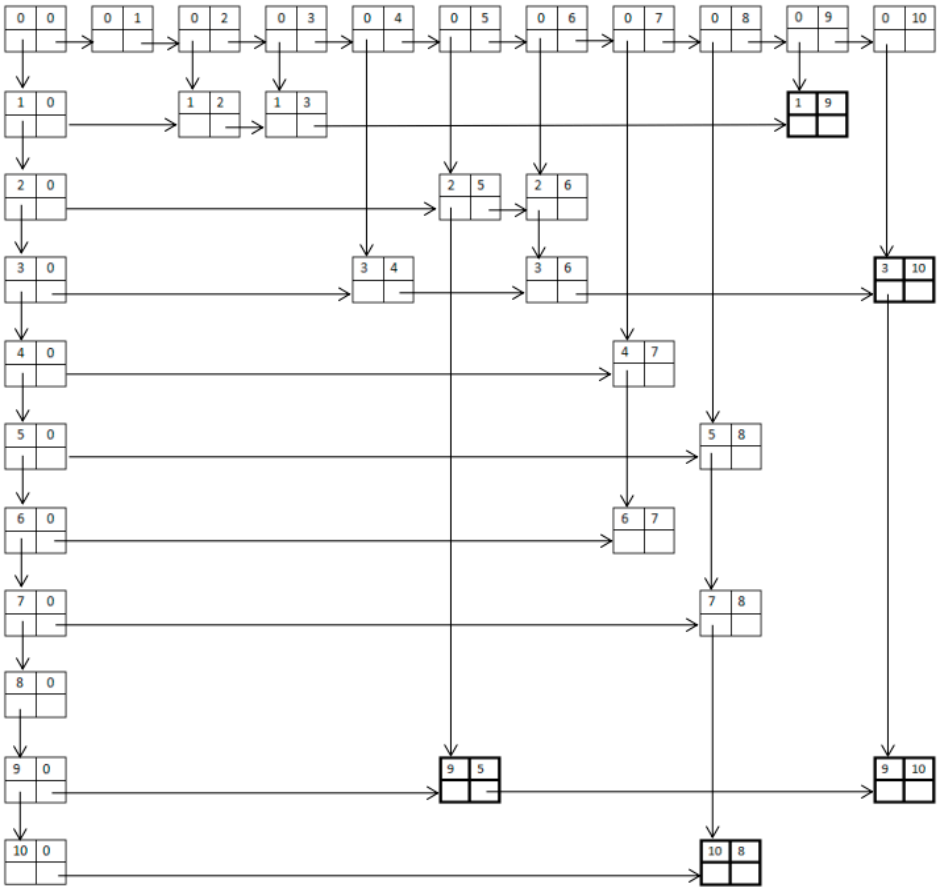

Figure 4. The cross data link after inserting the object.

The corresponding Hasse graph is shown in Figure 5:

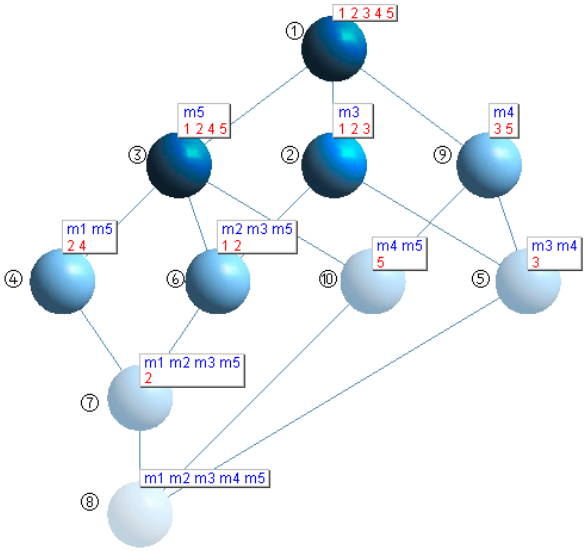

Figure 5. The Hasse graph after inserting the object.

The concept lattice construction algorithm based on the cross data link proposed in this section does not need to re-establish the concept lattice when adding objects. It only needs to operate on nodes and connections, and the connections are the direct relationships between subconcepts

and superconcepts. The algorithm considers the case of multiple direct superconcepts and has a wider applicability.

4. Experiments

4.1. The Evaluation of the Clustering Effect

Among the clustering methods, some are based on distance, some are based on density, etc. Due to the difference between the methods, there is no uniform evaluation index. Evaluation methods for clustering include methods such as the Standard IR method [29], the Merge-then-cluster method [29] and the user evaluation method. The method used in this paper contains three indexes for evaluating the clustering results, and is adopted based on the Standard IR method.

4.1.1. Cluster Label Readability

The cluster label readability refers to the proportion of readable labels in all labels, the higher the proportion is, the better the readability is. The formula is as follows:

$$CLR = \frac{u}{l} \tag{3}$$

where CLR represents cluster label readability; u represents the number of labels which have better readability; l represents the total number of labels.

4.1.2. Cluster Content Coverage

Cluster content coverage refers to the proportion of all initial documents in the clustering results. The formula is as follows:

$$CCC = \frac{o}{s} \tag{4}$$

where CCC represents content coverage; o represents the number of documents in the clustering result; s represents the number of initial documents.

4.1.3. Cluster Overlap

Cluster overlap refers to the proportion of the total number of documents that is repeated in the clustering results. The formula is as follows:

$$CO = \frac{c - s}{c} \tag{5}$$

where CO represents cluster overlap; c represents the total number of documents in the cluster labels.

4.2. Efficiency Verification

We analyzed the time efficiency of the algorithm through experiments. First of all, in order to obtain the experimental data sets, we used Java to generate a random "0 1" matrix of scale 300×20, We set the probability as 30% (i.e., the "1" in the proportion of matrix), and divide 400 object sets into 10 groups. To evaluate the time efficiency, we carried out experiments using the ListFCA algorithm as proposed in this paper, and the improved linked list algorithm and the Godin algorithm based on the references. Then we recorded the time consumption after each group calculation; the results are shown in Figure 6.

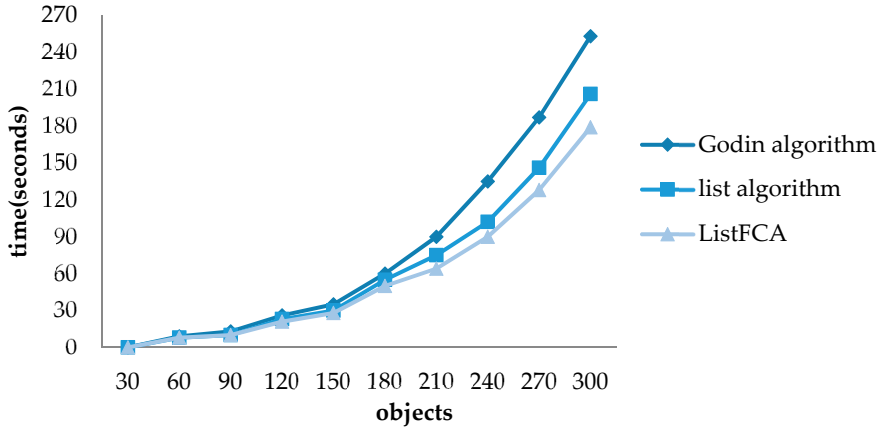

Figure 6. Result comparison when the probability is 30%.

We set the probability to 40%, with the other parameters remaining unchanged, and then recorded the time consumption; the results are shown in Figure 7.

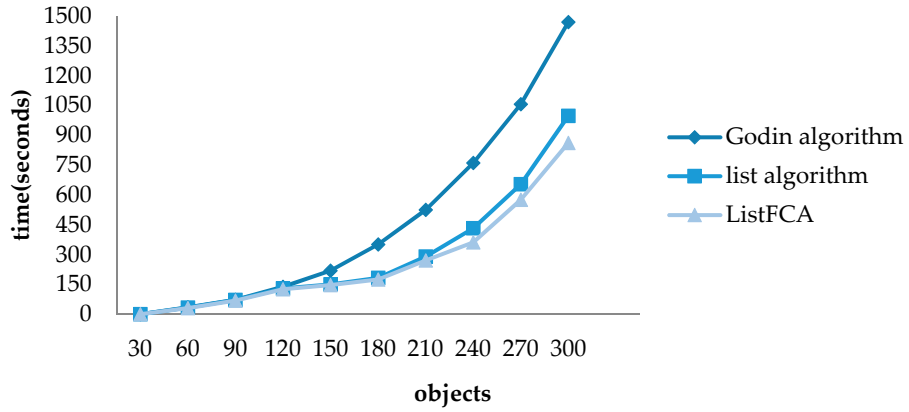

Figure 7. Result comparison when the probability is 40%.

4.3. Experiment and Analysis

To control the scale of the experiment, we first used a search engine to cluster the Web data of the search results for a keyword in the experiment in order to enhance the correlation of experimental Web data. "computer", "physics" and "environment" were used as the keywords, and 40 pages were selected for clustering. Then, we selected the first 5 feature items representing the Web page based on their weight during preprocessing of the Web page. The attributes of Web frequency equal to 1 were eliminated in the formal contexts. Taking the "computer" keyword as an example, we finally obtained the Hasse diagram shown in Figure 8.

In the above result, every node in the Hasse graph can be seen as a class of clustering result. They are repeatedly divided into different classes because of different combinations of attributes. Web page clustering based on formal concept analysis is very effective for querying simple or rich and ambiguous information. To show that the clustering method is effective and feasible, we compared the results of ClusterFCA [30] with a typical clustering method based on formal concept analysis. Similarly, "computer", "physics" and "environment" were used as the keywords of the search, and 40 pages were selected for clustering; we intercepted the second layers and the third layers of the index value to compare; the results are shown in Figure 9.

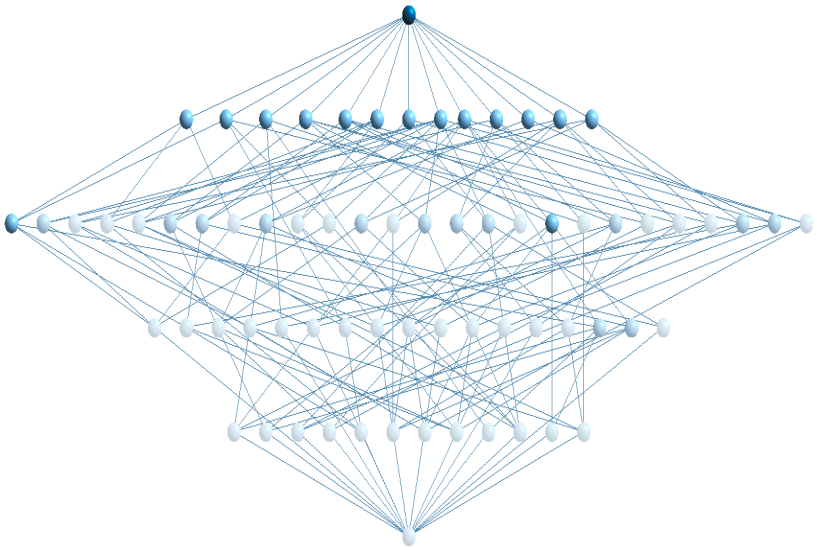

Figure 8. Hasse graph of the concept lattice corresponding to the 40 Web pages.

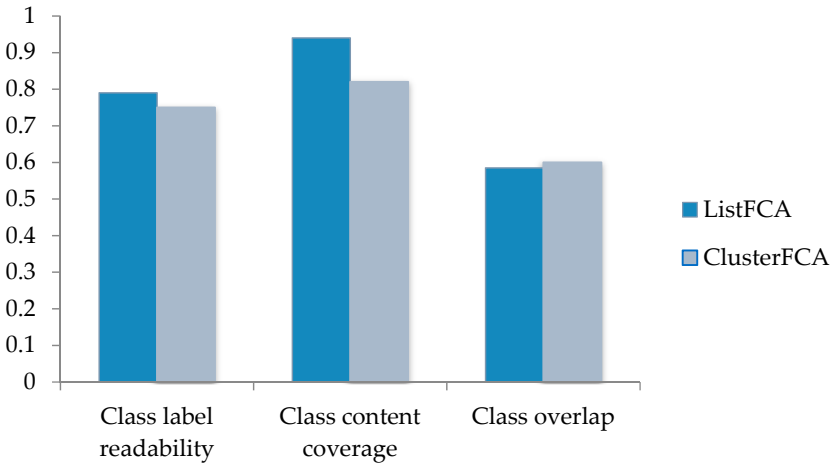

Figure 9. Comparison of evaluation between two methods.

From the comparison of the index values of the two methods, we can conclude that the readability of the cluster labels in the method introduced in this paper is higher than that of the other method. The cluster overlap is slightly lower than the ClusterFCA method, and the cluster coverage ratio of the ListFCA is much higher than the ClusterFCA method. Based on the above comparison, the ListFCA is proven to be more effective, and it has advantages in cluster content coverage.

5. Conclusions and Future

In this paper, we firstly studied a method of Web page clustering based on formal concept analysis. Then, we researched the preprocessing methods of Web pages and proposed a bi-directional maximum matching word segmentation method. Thirdly, through the study of different types of weighting algorithms, we found a new computing scheme based on term frequency. Finally, by studying the construction of concept lattices, we proposed a new concept lattice construction algorithm based on cross data link and verified its effectiveness. Using this algorithm, the Hasse graph corresponding to the concept lattice can be easily generated. The method of Web page clustering based on formal concept analysis has a high degree of visualization, rich content, and is characterized by overlapping clustering.

This paper has achieved a more efficient clustering performance through clustering Web pages based on FCA. However, the following properties still need to be further explored:

- Although the structure of Web pages is different from ordinary text, some research results of text classification and text clustering can also be used in Web page clustering. We should consider applying existing research results in other fields to Web clustering.
- The concept lattice is the core content of formal concept analysis, and there is still much room for improvement in the study of its construction algorithm. We can optimize the concept lattice construction algorithm based on a cross linked list in order to reduce the traversing times while maintaining the concept lattice with an increase or decrease in the number of objects.

Author Contributions: J.Z. and X.Y. performed the experiment and data collection. Z.Z. was involved in theoretical investigation and optimization.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 61379109, M1321007), Science and Technology Plan of Hunan Province (Grant No. 2014GK2018, 2016JC2011).

Acknowledgments: The authors would to thank the reviewers for their valuable suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, X.; Ding, C.H.; Zha, H.; Simon, H.D. Automatic topic identification using webpage clustering. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 46–54.
2. Dau, F.; Ducrou, J.; Eklund, P. Concept similarity and related categories in searchsleuth. In Proceedings of the 16th International Conference on Conceptual Structures, Toulouse, France, 7–11 July 2008; pp. 255–268.
3. Zamir, O.; Etzioni, O. Web document clustering: A feasibility demonstration. In Proceedings of the 21st Annual ACM/SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998; pp. 46–54.
4. Mecca, G.; Raunich, S.; Pappalardo, A. A new algorithm for clustering search results. *Data Knowl. Eng.* **2007**, *32*, 504–522. [CrossRef]
5. Sun, X. Construction data mining information management system based on FCA and ontology. In *Advances in Electronic Engineering, Communication and Management*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 19–24.
6. Hitzler, P. What's happening in semantic web: And what FCA could have to do with it. In Proceedings of the International Conference on Formal Concept Analysis, Nicosia, Cyprus, 2–6 May 2011; pp. 18–23.
7. Shen, X.; Xu, Y.; Yu, J.; Zhang, K. Intelligent search engine based on formal concept analysis. In Proceedings of the 2007 IEEE International Conference on Granular Computing (GRC 2007), San Jose, CA, USA, 2–4 November 2007; p. 669.
8. Freeman, L.C.; White, D.R. Using Galois lattices to represent network data. *Sociol. Methodol.* **1993**, *23*, 127–146. [CrossRef]
9. Wille, R. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets*; Springer: Dordrecht, The Netherlands, 1999; pp. 445–470.
10. Ganter, B.; Wille, R. *Formal Concept Analysis: Mathematical Foundations*; Springer: Berlin/Heidelberg, Germany, 2012.
11. Valtchev, P.; Missaoui, R.; Godin, R. Formal concept analysis for knowledge discovery and data mining: The new challenges. In Proceedings of the Second International Conference on Formal Concept Analysis, Sydney, Australia, 23–26 February 2004; pp. 352–371.
12. Poelmans, J.; Ignatov, D.I.; Viaene, S.; Dedene, G.; Kuznetsov, S.O. Text mining scientific papers: A survey on FCA-based information retrieval research. In Proceedings of the 12th Industrial Conference on Data Mining, Berlin, Germany, 13–20 July 2012; pp. 273–287.
13. Kuznetsov, S.O.; Obiedkov, S.A. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.* **2002**, *14*, 189–216. [CrossRef]
14. Kovács, L. Efficiency analysis of concept lattice construction algorithms. *Procedia Manuf.* **2018**, *22*, 11–18. [CrossRef]

15. Godin, R.; Missaoui, R.; Alaoui, H. Incremental concept formation algorithms based on Galois (concept) lattices. *Comput. Intell.* **1995**, *11*, 246–267. [CrossRef]
16. Chen, Q.Y. Improvement on Bordat algorithm for constructing concept lattice. *Comput. Eng. Appl.* **2010**, *46*, 33–35. (In Chinese)
17. Yang, S.T. A webpage classification algorithm concerning webpage design characteristics. *Int. J. Electron. Bus. Manag.* **2012**, *10*, 73.
18. Wang, Y.Z.; Chen, X.; Dai, Y.F. Web information extraction based on webpage clustering. *J. Chin. Comput. Syst.* **2018**, *39*, 111–115. (In Chinese)
19. Tan, K.W.; Han, H.; Elmasri, R. Web data cleansing and preparation for ontology extraction using WordNet. In Proceedings of the International Conference on Web Information Systems Engineering, Hongkong, China, 19–21 June 2000; pp. 11–18.
20. Bu, Z.; Zhang, C.; Xia, Z.; Wang, J. An FAR-SW based approach for webpage information extraction. *Inf. Syst. Front.* **2014**, *16*, 771–785. [CrossRef]
21. Cai, D.; Yu, S.; Wen, J.R.; Ma, W.Y. Extracting content structure for web pages based on visual representation. In Proceedings of the 5th Asia-Pacific Web Conference and Application, Xian, China, 23–25 April 2003; pp. 406–417.
22. Kwon, O.W.; Lee, J.H. Web page classification based on k-nearest neighbor approach. In Proceedings of the 5th international workshop on Information Retrieval with Asian Languages, Hong Kong, China, 30 September–1 October 2000; pp. 9–15.
23. Huang, C.N.; Gao, J.; Li, M.; Chang, A. Chinese Word Segmentation. U.S. Patent 10/662,602, 31 March 2005.
24. Cohen, W.W.; Singer, Y. Context-sensitive learning methods for text categorization. *ACM Trans. Inf. Syst. (TOIS)* **1999**, *17*, 141–173. [CrossRef]
25. Kuznetsov, S.O.; Obiedkov, S.A. Algorithms for the construction of concept lattices and their diagram graphs. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Barcelona, Span, 20–24 September 2010; pp. 289–300.
26. Zou, L.; Zhang, Z.; Long, J.; Zhang, H. A fast incremental algorithm for deleting objects from a concept lattice. *Knowl. Based Syst.* **2015**, *89*, 411–419. [CrossRef]
27. Croitoru, M.; Ferré, S.; Lukose, D. Conceptual Structures: From Information to Intelligence. *Lect. Note Comput. Sci.* **2010**, *9*, 26–30.
28. Frakes, W.B.; Baeza-Yates, R. *Information Retrieval: Data Structures and Algorithms*; Prentice Hall: Englewood Cliffs, NJ, USA, 1992.
29. Zamir, O.E. Clustering Web Documents: A Phrase-Method for Grouping Search Engine Results. Ph.D. Thesis, University of Washington, Seattle, WA, USA, 1999.
30. Wang, J. Research of Web Search Result Clustering Based on Formal Concept Analysis. Ph.D. Thesis, Xihua University, Chengdu, China, 2008.