An Extended Branch and Bound Search Algorithm for Finding Top-N Formal Concepts of Documents

Makoto HARAGUCHI and Yoshiaki OKUBO

Division of Computer Science
Graduate School of Information Science and Technology
Hokkaido University
N-14 W-9, Sapporo 060-0814, JAPAN
E-mail: { mh, yoshiaki }@ist.hokudai.ac.jp

Abstract. This paper presents a branch and bound search algorithm for finding only top N number of extents of formal concepts w.r.t. their evaluation, where the corresponding intents are under some quality control. The algorithm aims at finding potentially interesting documents of even lower evaluation values that belong to some highly evaluated formal concept. The experimental results show that it can effectively find such documents.

1 Introduction

One of the core tasks of Information Retrieval is to effectively find useful and important documents including Web pages. For this purpose, many retrieval engines compute ranks of documents, and show them in the order of their ranks [3, 2, 11]. Highly ranked documents are easily checked by users, while documents ranked lower are rarely examined. Any retrieval system based on document ranking has its own ranking scheme. So, even potentially interesting documents are sometimes ranked low and are therefore actually hidden and invisible to users.

The standard approach to cope with the above problem is to use the techniques of clustering [1, 4] by which we classify various documents into several clusters of similar documents. We pick up a few clusters that seems relevant, and then examine them in details to look for interesting documents. However, if the number of clusters is small, clusters tend to be larger ones involving even non-similar documents, and are hard to be examined. Conversely, if we have many number of clusters, it is also hard to check every cluster, although each cluster is smaller and involves only similar documents. Thus, it is not an easy task to have an adequate method for controlling the number of clusters.

For this reason, instead of dividing whole data set into several clusters by clustering, we have developed some strategy in [7, 5, 8] for finding only top N number of clusters of similar documents with respect to their evaluation values reflecting the ranks of documents in them. According to the strategy, the similarity is defined by the standard cosine measure for vector representation of documents and is used to draw edges among documents to form an undirected graph of documents. Then the algorithm has been designed as an extension of branch and bound maximum clique search algorithms [10, 9] to find top N (pseudo-) cliques as clusters of documents. We have already verified that the algorithm found some clusters in which lowly ranked documents appear in them together with highly ranked documents contributing toward raising the whole evaluation of clusters.

However, as is already pointed out in the area of conceptual clustering [12, 13], as long as the similarity of documents is derived from the cosine measure for vector representation, it is generally difficult to understand the meaning of clusters (cliques in this case) by means of feature terms. In our case of finding interesting documents with lower ranks, the detected lower ranked documents together with highly ranked documents in one cluster are in fact similar vectors. However, it is always difficult to judge if the former and the latter share the same meaning or not. In other words, the conceptual classes they belong may differ. In order to avoid such a conceptually indistinct argument, we make an informal constraint on the clusters to be obtained as follows:

> The notion of relevance or interestingness depends only on a conceptual class of documents, not dependent on particular instance document. Then the clusters we have to find must be concepts of documents that can be definable by means of feature terms.

As the primary data for document set is a document-term relationship, we adopt the notion of Formal Concept Analysis (FCA) [14, 12, 13]. As is well known, a formal concept consists of two closures, ψA called the extent of concept, a set of documents, and $A = \varphi\psi A$ called the intent of concept, a set of terms,

where A is a term set and (φ, ψ) is a Galois connection defined from the document-term relationship. Then, the extent ψA has a definition that $d \in \psi A$ iff $A \subseteq \varphi d$, where RHS means that a document d has terms in A. It is also well known that formal concepts can be computed by finding maximal *bipartite cliques* of a bipartite graph or equivalently by finding *closures* of documents or terms. Therefore, keeping the evaluation scheme for extents as clusters of documents, it can be a strategy to find only top-N extents by using some very fast enumeration algorithm, LCM [15] for instance, for finding all the closures.

The problem for such an approach is however that the number of possible extents is still large. Particularly, there exists e numerous number of extents of concepts whose corresponding intents are very smaller set of terms. Smaller intents we have, the extents tend to be larger sets of documents and to involve documents with less similarity. In other words, the quality of those extents becomes worse. For this reason, we revise our former algorithms [7, 5, 8] so that we try to find only top N extents w.r.t. the same evaluation schema for clusters, keeping the quality of extents by a given lower bound for the corresponding intents.

(FC1: **Evaluation**) Extents of formal concepts are evaluated by some monotone function. The evaluation becomes higher, as the extents grow as sets of documents, and as each document in them shows higher rank.
(FC2: **Graph Formation under Static Quality Control**) Two documents are judged similar if they share at least a given number of common terms. We draw an edge between any similar two documents, and form an weighted undirected graph of documents.
(PC3: **Extent Search under Dynamic Quality Control**) To enumerate only top N extents, closures of documents, our algorithm is again a branch and bound method, where
  (**Candidate Closures of Documents**) a list of N number of candidate top N closures is always kept.
  (**Standard Branch and Bound Pruning due to Monotone Evaluation**) any search node, a closure of documents, whose evaluation value is less than the minimum of those of candidates, we cut off the nodes below.
  (**Dynamic Quality Control**) any search node whose corresponding intent has less number of feature terms than a given lower bound, we also cut off the nodes below.

Clearly the two pruning rules are safe ones never missing any of top N extents satisfying the requirements. In addition, we can utilize another pruning rule based on a simple theoretical property of formal concepts. It is effective for reducing generation of useless formal concepts, that is, duplications of already generated ones.

Now let us go on to the details of our algorithm in Section 3. Our experimental results are presented in Section 4. Particularly in Section 5, we compare our algorithm with a very fast enumerator of closures from a computational viewpoint. Finally in Section 6, we discuss some future works.

2 Preliminaries

Let $\mathcal{O}$ be a set of *objects* (individuals) and $\mathcal{F}$ a set of *features* (attributes). A pair of $\mathcal{O}$ and $\mathcal{F}$, $< \mathcal{O}, \mathcal{F} >$ is called a *context*. Under a context $< \mathcal{O}, \mathcal{F} >$, each object in $\mathcal{O}$ is represented as the set of all features in $\mathcal{F}$ which the object has.

Given a context $< \mathcal{O}, \mathcal{F} >$, for a set of objects $O \subseteq \mathcal{O}$ and a set of features $F \subseteq \mathcal{F}$, we define two mappings $\varphi : 2^{\mathcal{O}} \to 2^{\mathcal{F}}$ and $\psi : 2^{\mathcal{F}} \to 2^{\mathcal{O}}$, respectively, as follows.

$$\varphi(O) = \{f \in \mathcal{F} \mid \forall o \in O, f \in o\} = \bigcap_{o \in O} o \quad \text{and} \quad \psi(F) = \{o \in \mathcal{O} \mid F \subseteq o\}.$$

The former computes the feature set shared by every object in O. The latter, on the other hand, returns the set of objects with F.

Based on the mappings, a *formal concept* (FC) under the context is defined as a pair of object set and feature set, (O, F), where $O \subseteq \mathcal{O}$, $F \subseteq \mathcal{F}$, $\varphi(O) = F$ and $\psi(F) = O$. Especially, O and F are called the *extent* and *intent* of the concept, respectively. From the definition, it is obvious that $\psi(\varphi(O)) = O$ and $\varphi(\psi(F)) = F$. That is, a formal concept is defined as a pair of *closed* sets under the mappings.

Let $\mathcal{T}$ be a set of *feature terms*. A document d can be represented as a set of feature terms in $\mathcal{T}$ appearing in the document, that is, $d \subseteq \mathcal{T}$. For a set of documents $\mathcal{D}$, therefore, we can consider a formal concept (D, T) under the context $< \mathcal{D}, \mathcal{T} >$. The extent D is regarded as a cluster of documents and the intent is the feature terms which are shared by the documents in D.

Let $G = (V, E)$ be an undirected graph, where V is a set of vertices and E ($\subseteq V \times V$) a set of edges. For a set of vertices $V' \subseteq V$, the graph G' defined as $G' = (V', E \cap (V' \times V'))$ is called a *subgraph of* G (*induced by* V'). If a subgraph $G' = (V', E')$ of G is complete, then G' is called a *clique* of G, where the clique is often denoted by simply V'.

3 Finding Document Clusters Based on Formal Concept Analysis

In order to provide clear meanings of clusters, we try to improve the previous method of pinpoint clustering with *Formal Concept Analysis* (FCA) [14]. Formal Concept Analysis is a theory of data analysis which identifies *conceptual structures among objects* (individuals).

3.1 Document Clusters as Formal Concepts

Let $\mathcal{T}$ be a set of feature terms and $\mathcal{D}$ a set of documents, where each document is represented as a set of feature terms in $\mathcal{T}$ appearing in the document.

Regarding $\mathcal{C} =< \mathcal{D}, \mathcal{T} >$ as a context in FCA, a formal concept (D, T) under $\mathcal{C}$ can be viewed as a cluster of documents. It should be noted here that we can clearly explain why documents in D are grouped together. Each document in D shares the set of feature terms T and any other document never contains T. In this sense, D can form a meaningful grouping (cluster) of documents. Thus, by restricting our clusters to being formal concepts under $\mathcal{C}$, we can explicitly consider the meanings based on their intents. We call this kind of clusters *Formal Concept-Based clusters* (FC-clusters in short).

The meaningfulness of FC-cluster is affected by both of its intent and extent. For example, a cluster with smaller intent might be unconvincing because the evidence for the grouping seems to be weak, though its extent tends to be larger. Conversely, although a cluster with larger intent might have more convincing evidence, its extent tends to be smaller. Thus, it is required to control the quality of FC-clusters in order to obtain useful ones. From these observations, we formalize FC clusters to be found as follows:

Quality Control : Constraint on Intents Quality of FC-cluster to be found is controlled by imposing a constraint on intent. As such a constraint, we will give a threshold δ for evaluation value of intent. For an FC-cluster, if the evaluation value of the intent is greater than or equal to δ, then the cluster is said to be *δ-valid*. As we will see later, the constraint can work statically and dynamically to prune useless search.

Preference in Extents: Among the δ-valid FC-clusters, we prefer ones with higher evaluation values of their extents. Especially, we try to extract clusters whose extents have Top-N evaluation values.

In order to evaluate intents and extents, we assume that each document d and feature term t have their (positive) weights which are referred to as $weight(d)$ and $weight(t)$, respectively. For example, each document is given a weight based on its rank assigned by an information retrieval system. Furthermore, a weight of feature term might be defined as the *inverted document frequency*(IDF) of the term. Then, an evaluation function might be defined as the *sum of weights* of individuals in each extent (intent). Evaluation by size of intent (extent) can be viewed as a special case of the function.

Thus, intents and extents of formal concepts can be evaluated from several viewpoints. We can actually define various evaluation functions for them. From the computational point of view, however, a function which behaves *monotonically* according to expansion of extents (intents) is strongly preferred. More concretely speaking, we prefer an evaluation function f such that for any set S and its superset S', $f(S) \leq f(S')$ holds. It is obvious that the above evaluation function based on the sum of weights behaves monotonically. The reason why such a function is preferable will become clear shortly.

Our problem of finding Top-N δ-valid FC-clusters is precisely defined as follows:

[Given] $\mathcal{T}$: a set of feature terms
$\mathcal{D}$: a set of documents each of which is represented as a subset of $\mathcal{T}$
w_d : an evaluation function for sets of documents
w_t : an evaluation function for sets of feature terms
δ : a threshold for the minimum evaluation value for intent for intent

[Find] the set of formal concepts $\{(T, D)\}$ such that the evaluation value of D, $w_d(D)$, is in the top N among δ-valid formal concepts under the context $< \mathcal{T}, \mathcal{D} >$

3.2 Algorithm for Finding Top-N δ-Valid FC-Clusters by Clique Search

Top-N δ-valid FC-clusters can be extracted by finding certain *cliques* in an weighted undirected graph.

Graph Construction

Given a context $\mathcal{C} =< \mathcal{T}, \mathcal{D} >$, an evaluation function for term sets w_t and a threshold δ for the quality of intents, we first construct an weighted undirected graph $G = (\mathcal{D}, V)$, where V is defined as

$$V = \{(d_i, d_j) \mid d_i, d_j \in \mathcal{D}(i \neq j) \wedge w_t(d_i \cap d_j) \geq \delta\}.$$

That is, if a pair of documents share a set of feature terms whose evaluation value is greater than or equal to δ, then they are connected by an edge. For example, let us assume a set of documents

$$\mathcal{D} = \{d_1 = \{a, b, c, d\}, d_2 = \{a, b, f\}, d_3 = \{b, c, f\}, d_4 = \{b, d, e\}, d_5 = \{a, b, e, g\}\},$$

where each document d_i is given its weight $1/i$, refered to as $w(d_i)$. Let w_d and w_t be evaluation functions defined as $w_d(D) = \sum_{d_i \in D} w(d_i)$ for a document set D and $w_t(T) = |T|$ for a term set T, respectively. Under the setting of $\delta = 2$, we have a weighted undirected graph

$$G = (\mathcal{D}, \{(d_1, d_2), (d_1, d_3), (d_1, d_5), (d_2, d_3), (d_2, d_5), (d_4, d_5)\}),$$

where the weight of each vertex d_i is $1/i$.

From the definition of the graph G, any extent of *delta*-valid FC can be found as a clique in G. Therefore, Top-N δ-valid FCs can be obtained by searching cliques in G. It should be noted here that any clique does not always corresponds to a formal concept even if it is a maximal clique in the graph. In the above example, $\{d_1, d_2, d_3\}$ is a maximal clique, but it is not an extent of formal concept. Each d_i in the clique shares the term b, that is, $\varphi(\{d_1, d_2, d_3\}) = \{b\}$. However, since $\phi(\{b\}) = \{d_1, d_2, d_3, d_5\}$ holds, $\{d_1, d_2, d_3\}$ is not a closed set. This means that there exists no FC whose extent is the clique. It is remarkable that although $(\{b\}, \{d_1, d_2, d_3, d_5\})$ is an FC, it is not δ-valid and the extent is out of our clique search. In our graph construction, thus, many useless cliques can be eliminated *statically* based on the threshold δ.

Search Strategy

From the graph G, we try to extract δ-valid FCs whose extents have Top-N evaluation values.

Our algorithm finds Top-N formal concepts with *branch-and-bound depth-first search strategy*. During our search, we maintain a list of formal concepts which stores Top-N FCs among ones already found. That is, the list keeps *tentative* Top-N clusters.

Basically speaking, for each clique (a set of documents) Q, $\varphi(Q)$ is computed and then evaluated. If $w_t(\varphi(Q)) \geq \delta$, the set of feature terms $\varphi(Q)$ can become the intent of a δ-valid FC with its extent $\psi(\varphi(Q))$. Then, the tentative Top-N list is adequately updated for the obtained FC $(\psi(\varphi(Q)), \varphi(Q))$. The procedure is iterated in depth-first manner until no Q remains to be examined. That is, starting with the initial Q of the empty set, Q is expanded step by step.

As has been mentioned above, in our algorithm, we assume that as an intent becomes larger under set inclusion, our evaluation function for intents behaves monotonically. Especially, we prefer a monotonically decreasing function. As long as we evaluate intents with such a monotonic function, we can utilize a simple pruning rule. That is, for a clique Q, if $w_t(\varphi(Q)) < \delta$ holds, then we can never obtain δ-valid formal concepts from any extension of Q, because its corresponding intent becomes smaller. Therefore, we can immediately stop expanding Q and backtrack. Thus, the quality of intents is *dynamically* controlled in our search.

Moreover, we can also enjoy a pruning based on tentative Top-N clusters. From a theoretical property of cliques, if $w_d(Q) + w_d(cand(Q))$ is less than the minimum evaluation value of extents in the tentative Top-N list, we do not have to examine any extension of Q, where $cand(Q)$ is the set of vertices adjacent to any vertex in Q. More concretely speaking, $w_d(Q) + w_d(cand(Q))$ gives an upper bound of extent values obtained by expanding Q. If the upper bound is less than the tentative minimum value, any extension of Q is no longer useful and can be pruned safely.

In addition to the prunings, our current algorithm also utilizes another pruning rule based on a theoretical property of formal concepts. It is quite effective for reducing generation of useless FCs, that is, duplications of already generated ones.

The pruning rule is based on the following simple property:

Table 1. FC-Based Clusters

Cluster ID.	Extent (Page IDs)	Intent
FC_1	194 203 205 210	Adam Archiv Back Carr39; Nation Psepho Top middot summari
FC_2	20 21 66 709	Administr Bush COVERAGE Coverag Elector FULL Full New RELATED Reform Yahoo amp
FC_3	246 280 405 600 608	05 Lanka Sri Tamil TamilNet accur concern featur focus inform issu new peopl provid reliabl servic
FC_4	176 205 444	2001 Adam Archiv Carr39; Nation Psepho provinc summari
FC_5	70 326 479	Ukrainian alleg controversi exampl fraud includ irregular massiv

Let Q be a set of documents. For any documents α and β, if $\alpha \in \psi(\varphi(Q \cup \{\beta\}))$ and $\beta \in \psi(\varphi(Q \cup \{\alpha\}))$, then $\psi(\varphi(Q \cup \{\alpha\})) = \psi(\varphi(Q \cup \{\beta\}))$.

For a clique Q, let W_Q be the set of vertices already used to expand Q. Assume we try to expand Q with a vertex α. From the above property, if there exists a vertex $w \in W_Q$ such that $\alpha \in \psi(\varphi(Q \cup \{w\}))$ and $w \in \psi(\varphi(Q \cup \{\alpha\}))$, then we do not have to expand Q with α. Expanding Q with α will generate a duplication of an extent already generated. Therefore, the search branch can be pruned safely.

4 Experimental Result

In this section, we present our experimental result.

We have conducted an experimentation to observe characteristics of FC-clusters.

A set of web pages $\mathcal{P}$ to be clustered has been retrieved by using *Google Web API* [1] with keywords "Presidential" and "Election". The number of retrieved pages is 968. For each page, its *summary* and *snippet* extracted by Google Web API are gathered as a document. After the stemming process, we have obtained 3600-terms in the pages (documents) and extracted 947 of them as feature terms [2] . Therefore, each page is represented as a 947-dimensional document vector.

Each web page p is assigned a linear weight $w(p) = |\mathcal{P}| - rank(p) + 1$, where $rank(p)$ is the rank assigned by Google. Each extent (a set of pages) is evaluated by the sum of the page weights.

The weight of feature term t is given as *inverted document frequency* of t, that is, $w(t) = \log(|\mathcal{P}|/df(t))$, where $df(t)$ is the number of pages in $\mathcal{P}$ containing the term t. Each intent is evaluated by the sum of term weights in the intent.

Under the setting of $\delta = 33.0$, we constructed a weighted undirected graph. Roughly speaking, since the average of term weights is approximately 5.0, a pair of web pages sharing about $6 - 7$ feature terms are connected by an edge.

We tried to extract Top-10 δ-valid FC-clusters from the graph. The computation time was just 0.52 second. Some of the obtained clusters are shown in Table 1.

For comparison, we extracted several clusters based on the standard cosine measure for vector representation of documents. As one of them, we can obtain a cluster D consisting of the pages with the ranks $176, 191, 193, 194, 203, 204, 205, 210$ and 465. Note here that FC_1 in Table 1 is a subset of D. Needless to say, it is difficult to understand the meaning of D by means of feature terms. On the other hand, we can clearly understand $F1$ based on the intent. Thus, we can conceptually understand each cluster by referring to the intent. It would be easy to judge whether the cluster is interesting for us or not.

The cluster FC_2 shows that we can obtain 709-th page with (relatively) higher-ranked pages with the ranks 20, 21 and 66. Such a lower-ranked page would not be browsed in many cases. However, our cluster tells us the page might be significant if the concept (intent) would be interesting for us. Thus, our chance to find significant lower-ranked pages can be enhanced.

The cluster FC_3 shows a remarkable characteristic of our method. It consists of web pages concerned with the presidential election by *Tamil people*. The pages would not be so popular and their ranks are really lower. Therefore, almost people will miss them. However, our method can make such clusters visible. More precisely speaking, such an effect can be controlled by our definition of document weights.

In the experimentation, each document is given a linear weight based on its rank. In other words, such a linearly weighting gives some degree of importance to documents with not higher ranks. Therefore, a cluster consisting of only middle-ranked pages might be extracted as in Top-N, if its size is relatively

[1] `http://www.google.com/apis/`

[2] All terms with the frequencies above 100 and below 3 have been removed.

larger. On the other hand, if we assign a weight to each document d such as $w(d) = 1/rank(d)^2$, only clusters with higher-ranked pages will be extracted as in Top-N.

5 Discussion

In the previous sections, we have discussed a method for Top-N pinpoint clustering of documents. However, it is not limited only for documents. Since our method is a general framework, we can easily apply it for others which can be represented as *relational data*.

Formal Concept Analysis is closely related to the problem of *closed itemset mining* (e.g. [15]). More precisely speaking, there is an exact correspondence between a formal concept and a closed itemset. For a context $\mathcal{C} =< \mathcal{O}, \mathcal{F} >$ in FCA, $\mathcal{O}$ and $\mathcal{F}$ can be viewed as a set of transactions and a set of items, respectively, in a problem of itemset mining. In this case, the intent of a formal concept is equivalent to a closed itemset. Thus, finding formal concepts is equivalent to mining closed itemsets.

Uno *et al.* have designed an efficient algorithm, named *LCM*, for *enumerating* all frequent closed itemsets [15]. It is recognized as the fastest algorithm in the world for the problem. From the above correspondence, *LCM* can efficiently enumerate all formal concepts. Therefore, one might claim that we would be able to efficiently extract Top-N FC-clusters by simply enumerating all FCs and then sorting δ-valid ones in descending order of extent values. Our preliminary experimentations have shown that the claim is really true under several problem settings. However, the authors emphasize the following remarkable characteristics and advantage of our algorithm:

- *LCM* finds all closed itemsets whose frequencies are greater than or equal to a given threshold. From a theoretical property of itemsets, an itemset with larger size tends to have a lower frequency. Therefore, if we require a higher value of δ to retain a certain quality of formal concepts, extents of preferable formal concepts will become smaller. This means that when we try to extract such formal concepts with the help of *LCM*, we have to provide a lower threshold of frequency. However, as the threshold becomes lower, the number of frequent closed itemsets becomes larger. In such a case, therefore, the computation necessarily takes longer time, because *LCM* is an enumeration algorithm for such formal concepts. As the result, we need much computation time for finding Top-N FC-clusters with *LCM*.
- On the other hand, our algorithm takes both quality and preference of formal concepts into account and can prune many useless formal concepts during search. Especially, in case of higher δ, the computation of Top-N FC-clusters is much faster than one with *LCM*.

The above has been really observed in our preliminary experimentation. In the experimentation, the set of documents to be clustered consists of 5619 articles on politics in newspapers, where the number of feature terms is 2793. The documents and the feature terms are simply assigned uniform weights. That is, intents and extents are evaluated by their sizes. In case of $\delta = 30$, our algorithm takes just 1.21 second for finding Top-5 FC-clusters. On the other hand, in order to obtain the same clusters with *LCM*, we have to give a frequency threshold 5. In that case, *LCM* enumerates about 30 million formal concepts taking 198 second. Furthermore, we have to extract Top-5 30-valid clusters from them. Thus, our algorithm is quite efficient in case of higher δ.

6 Concluding Remarks

In this paper, we discussed a method for pinpoint clustering of documents based on Formal Concept Analysis. Our cluster can consist of similar higher-ranked and lower-ranked pages. Although we are usually careless of pages with lower ranks, they can be *explicitly* extracted together with significant higher-ranked pages. As the result, our clusters can provide new valuable information for users.

Our clusters can be explicitly provided with more convincing meanings, with the help of FCA. By restricting our clusters (cliques) to formal concepts, we can consider their clear conceptual meanings as their intents (the set of shared feature terms). We designed an algorithm for finding Top-N FC-clusters. It can be viewed as an extended algorithm of our previous one. The extended algorithm utilizes new pruning rules based on a theoretical property of formal concepts. In our experimentation, we confirmed that meaningful clusters can be really extracted according to our method. Furthermore, we verified that our algorithm can efficiently finds Top-N clusters compared with *LCM* in case of higher δ. From the observations, we expect that our method based on Formal Concept Analysis would be a promising approach to finding meaningful clusters of documents.

In general, we can observe some correlation among feature terms. By analyzing such a correlation among terms, we can reduce the number of feature terms to be taken into account. As the result, we would be able to consider meaningful intents with low-redundancy. The method of *Latent Semantic Indexing* (LSI) [11] would be useful for this purpose. Improving our current method from this viewpoint is an important future work.

It is known that formal concepts we really observe are quite sensitive to *noise* or *exceptions* in a context (data set) we are concerned with. In general, existence of such noise or exceptions will increase the number of possible FCs. Especially, we often obtain many FCs which are *slightly different*. Therefore, one might reasonably claim that some *data cleaning* process would be indispensable.

As another approach, *approximating* FCs is also effective. For example, if several FCs are almost the same, an approximate FC can be obtained by grouping them together. The notion of pseudo-cliques [8] will become a basis of this kind of approximation. Furthermore, an effective method for approximating closed itemsets has been investigated in the literature [6]. As is well known, since closed itemsets exactly correspond to intents of FCs, an approximation of our valid FCs can be proposed by extending the method in [6].

Needless to say, our method is not only for document clustering. We can apply our method to another kind of data in which each object to be clustered can be represented as a set of attributes. Applying the method to other practical data will be an interesting work.

References

1. M.W. Berry (Ed.), Survey of Text Mining : Clustering, Classification, and Retrieval, Springer, 2004.
2. L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", http://dbpubs.stanford.edu/pub/1999-66, 1999.
3. G. Salton and M. J. McGill, Introduction to modern information retrieval, Mcgraw-Hill, 1983.
4. A. Vakali, J. Pokorný and T. Dalamagas, "An Overview of Web Data Clustering Practices", Proceedings of the 9th International Conference on Extending Database Technology - EDBT'04, Springer-LNCS 3268, pp. 597 - 606, 2004.
5. M. Haraguchi and Y. Okubo, "A Method for Pinpoint Clustering of Web Pages with Pseudo-Clique Search", Federation over the Web, International Workshop, Dagstuhl Castle, Germany, May 1 - 6, 2005, Revised Selected Papers, Springer-LNAI 3847, pp. 59 - 78, 2006.
6. K. Kanda, M. Haraguchi and Y. Okubo: "Constructing Approximate Informative Basis of Association Rules", Proceedings of the 4th International Conference on Discovery Science - DS'01, Springer-LNAI 2226, pp. 141 - 154, 2001.
7. Y. Okubo and M. Haraguchi, "Creating Abstract Concepts for Classification by Finding Top-N Maximal Weighted Cliques", Proceedings of the 6th International Conference on Discovery Science - DS'03, Springer-LNAI 2843, pp. 418 - 425, 2003.
8. Y. Okubo, M. Haraguchi and B. Shi, "Finding Significant Web Pages with Lower Ranks by Pseudo-Clique Search", Proceedings of the 8th International Conference on Discovery Science - DS'05, Springer-LNAI 3735, pp. 346 - 353, 2005.
9. E. Tomita and T. Seki, "An Efficient Branch and Bound Algorithm for Finding a Maximum Clique", Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science - DMTCS'03, Springer-LNCS 2731, pp. 278 - 289, 2003.
10. T. Fahle, "Simple and Fast: Improving a Branch and Bound Algorithm for Maximum Clique", Proceedings of the 10th European Symposium on Algorithms - ESA'02, Springer-LNCS 2461, pp. 485 - 498, 2002.
11. K. Kita, K. Tsuda and M. Shishibori, "Information Retrieval Algorithms", Kyoritsu Shuppan, 2002 (in Japanese).
12. A.Hotho, G. Stumme, Conceptual Clustering of Text Clusters, In Proc. of the Machine Learning Workshop (FGML'02), 37–45, 2002.
13. A.Hotho, S. Staab, G. Stumme, Explaining text clustering results using semantic structures, Principles of Data Mining and Knowledge Discovery, 7th European Conference (PKDD 2003), 2003
14. B. Ganter and R. Wille, "Formal Concept Analysis: Mathematical Foundations", Springer, 1999.
15. T. Uno, M. Kiyomi and H. Arimura, "LCM ver. 2: Efficient Mining Algorithm for Frequent/Closed/Maximal Itemsets", IEEE ICDM'04 Workshop FIMI'04, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-126/, 2004.