19

Pinpoint Clustering of Web Pages and Mining Implicit Crossover Concepts

Makoto HARAGUCHI and Yoshiaki OKUBO
IST, Hokkaido University
JAPAN

1. Introduction

In this chapter, we present our Top-N methods for extracting clusters of documents which have originated from the article (Haraguchi, 2002). We first discuss a method for pinpoint clustering of Web pages by pseudo-clique search (Haraguchi & Okubo, 2006; Okubo et al., 2005) and then a method for finding implicit page groups (clusters) represented as formal concepts (Li et al., 2008).

A huge collection of documents including pages over the Web has been considered as an information source of knowledge. One of the core tasks of *Information Retrieval* (*IR*) is to effectively find useful and important documents from such a collection. For this purpose, many retrieval engines compute *ranks* of documents and show them in the order of their ranks (Page et al., 1999; Salton & McGill, 1983). 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 lower and are therefore actually hidden and invisible to users. In this sense, we might be missing many useful documents. If we can make such hidden significant documents visible, our chance to obtain valuable information and knowledge can be enhanced.

The standard approach to cope with this problem is to use the techniques of *clustering* (Gan et al., 2007) by which we classify various documents into several clusters of similar documents. We pick up a few clusters that seem to be 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 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.

This has motivated us to investigate a new clustering method, *Pinpoint Clustering*, by which we can efficiently extract *only nice clusters*. We have developed some strategy in (Haraguchi & Okubo, 2006; Okubo et al., 2005) for finding only Top-N number of clusters of similar documents with respect to their evaluation values reflecting the ranks of documents in them.

In the framework, the document similarity is evaluated with the help of *Singular Value Decomposition*(SVD) (Strang, 2003). We first extract semantic correlations among terms by applying SVD to the term-document matrix generated from a corpus with a specific topic. Then, given a set of ranked Web pages to be clustered, we evaluate potential similarities among

them based on the semantic correlations of terms, with the standard cosine measure for document vectors. Based on the similarities, we draw edges among similar documents to form a (weighted) undirected graph of documents. An algorithm has been designed as an extension of branch-and-bound *maximum clique search algorithms* (Fahle, 2002; Tomita & Seki, 2007) to find Top-*N pseudo-cliques* as clusters of documents. As is shown in Section 3, we verify that the algorithm can find clusters in which lowly ranked documents appear in them together with highly ranked documents contributing toward raising the whole evaluation of clusters.
However, it has already been pointed out in the area of *conceptual clustering* (Hotho et al., 2003; Hotho & Stumme, 2002) that 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 to may differ. In order to avoid such a conceptually indistinct argument, a method for finding Top-*N* clusters based on *formal concepts* in *Formal Concept Analysis* (*FCA*) (Ganter & Wille, 1999; Ganter et al., 2005) has been investigated (Haraguchi & Okubo, 2007; Li et al., 2008; Okubo & Haraguchi, 2006). Based on these our studies, we also discuss in this chapter a problem of mining *implicit Web page groups* from the data in the form of page-term relationship. In other words, our target page group is a relatively smaller set X of pages that has an intentional definition that "X is a set of pages that have every term in a feature term set A". Then a formal concept is a pair of X, called the *extent* of concept, and its term set A, called the *intent*.
Such an implicit concept will be useful in discovering "*Crossover Group of Pages*" for instance. Suppose we have several concepts with their extents of large numbers of pages so that they are visible by applying standard effective mining engines as (Han et al., 2007; Lakhal & Stumme, 2005; Uno et al., 2004; Wang et al., 2003) for instances. These pages are not necessarily connected by links, as we consider here a page-term relationship only. Suppose furthermore those groups are extensionally far away. There may be no overlapping. Even for such a case, there exists a possibility for two minor groups, each from each major group, of sharing common important feature terms. From a viewpoint of *FCA*, the union of the minor groups appears as a part of the concept defined from the common terms (see Figure 1). When the concept is minor with relatively smaller extent, the concept is worth examining to check if some invisible interconnection among the parent major groups occurs via the minor one. Those implicit concepts are also hard to be found by *clustering* (Gan et al., 2007). To detect implicit extents with smaller size, we are forced to have a large number of smaller clusters. It is actually unpractical for users to check them all. Without category labels to pages, or almost equivalently without using prior clustering, we show in Section 5.3 that our algorithm succeeds in finding several interesting implicit concepts beyond several distinct categories.
As is well known, each intent of concept just corresponds to a *closed itemset* of an association rule (Bastide et al., 2000). Many nice algorithms (Han et al., 2007; Lakhal & Stumme, 2005; Uno et al., 2004; Wang et al., 2003) for finding frequent closed itemsets have been developed successfully. However, since our targets are non-frequent, we cannot apply them at least directly.
A similar problem about potentially implicit page groups has been already conceived as *"implicitly defined communities"* (Zhang et al., 2006). The implicitly defined communities have too specific interests and are generally difficult to be identified via Web portals or centers in the

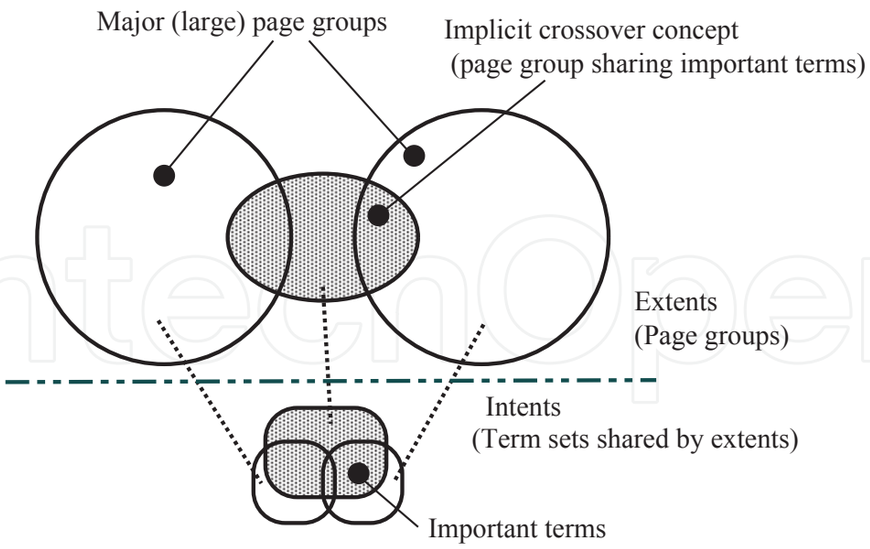

Fig. 1. Crossover Concept

bipartite graph (bigraph). Consequently the number of such communities is large. The situation will be worse when we consider a bigraph representing page-term relationships with a higher density. We are, therefore, required to have more effective miner for detecting implicit concepts under some constraints. In this sense, ours is an instance of *Constrained Mining* (Boulicaut & Jeudy, 2005).

For this purpose, we present in this chapter a revised version of the Top-N algorithm (Okubo & Haraguchi, 2006). Both of them try to enlarge extents as long as their intents are longer patterns to some extent. In other words, since too much smaller extents are out of our concerns, we maximize the extent size under the constraint about the corresponding intent's size. The algorithms are basically based on a depth-first and branch-and-bound search method (Tomita & Seki, 2007) with a pruning rule to cut off candidate concepts whenever their over-estimated evaluation values are less than the tentative Top-N values already detected.

In this chapter, to cope with large scale data and to reflect user's interests, we firstly improve the ability to enumerate possible solution concepts based on a *dynamic ordering technique*, and then introduce additional *space constraints*. A similar ordering strategy is also used in (Bayardo Jr., 1998; Burdick et al., 2001) to find longer itemsets using a *set enumeration tree*. In that case, however, no special expansion rule to avoid duplication is needed, while ours needs an expansion rule to skip duplications. Another important technique to improve the efficiency of pattern miners is a preprocessing method for concise representation (Wang et al., 2003) of dataset. However, our Top-N algorithm accesses only a part of whole data by the branch-and-bound pruning. For this reason, we here do a direct depth-first search without applying prior data analysis. A miner that searches for longer patterns (called *colossal patterns*) has been also proposed in (Zhu et al., 2007). It is based on some bias to avoid hopeless search for longer patterns, while we introduce some *space constraints* under which ours keeps the ability to enumerate every solution satisfying the constraints.

We introduce the constraints of three kinds. The first one defines a starting extent that must include positive example pages. The second one requires for an extent not to cover any nega-

tive example pages. The positive and negative examples are also used in (Murata, 2003; 2000) to discover Web communities, given an Web bigraph consisting of centers and fans, where the communities are found by enlarging initial page groups guided by best-first search heuristics. Our Top-N method is also considered as an enlargement process. However, it is complete in the sense that it finds every solution page group under the constraints.
Although we allow to use negative examples, users seem not to be aware of target pages or concepts and their counterparts as well. For this reason, we introduce the third constraint in addition to positive and negative examples. The third one is for realizing searches with an upper bound concept whose intent is just a set of terms given by user. The constraint contributes for accelerating the search and for keeping the interestingness of the result to some extent, as we see in Section 5.3.
In a word, our constrained search can respond within 10 seconds for $10,000$ pages with $1,200$ terms, given an adequate set of constraints. Thus, the algorithm can run in an interactive mining environment for analyzing search results and for realizing implicit page groups connecting major groups. This will motive us to search Web from a different point of view represented by implicit concepts.
The remainder of this chapter is organized as follows. In the next section, we introduce some basic terminologies used throughout this chapter. Section 3 discusses a method for pinpoint clustering of Web pages by pseudo-clique search. An interesting cluster with higher and lower ranked pages is also presented. In Section 4, we turn our attention from clique-based clusters to formal concept-based clusters. In Section 5, we discuss our method for finding implicit groups of pages. We describe our problem specification and discuss an efficient algorithm for the problem. We show some concrete examples of interesting page groups including a crossover concept. Computational performance of our algorithm is also presented. In the final section, we conclude this chapter with a summary and an important future direction.

2. Preliminaries

We introduce in this section some terminologies used throughout this chapter.
A *simple graph* is denoted by $G = (V, E)$, where V is a set of *vertices* and $E \subseteq V \times V$ a set of (undirected) *edges*. For any vertices $v, v' \in V$, if $(v, v') \in E$, v is said to be *adjacent* to v'. If any pair of vertices $v, v' \in V$ $(v \neq v')$ are adjacent each other, then G is said to be *complete*. For a vertex $v \in V$, the set of vertices adjacent to v is denoted by $N_G(v)$, that is, $N_G(v) = \{v' \mid v' \in V \wedge (v, v') \in E\}$. The size of $N_G(v)$, $|N_G(v)|$, is called the *degree* of v in G. It is often referred to as $degree_G(v)$. If it is clear from the context, they are simply denoted by $N(v)$ and $degree(v)$, respectively. If each vertex $v \in V$ is assigned a positive weight, the graph is called a *weighted graph*. The weight of v is referred to as $w(v)$. For a vertex set $V' \subseteq V$, the weight of V', denoted by $w(V')$, is simply defined as the sum of individual weights, that is, $w(V') = \sum_{v \in V'} w(v)$. In this chapter, we are concerned with a weighted graph unless stated otherwise.
For a graph $G = (V, E)$, a complete subgraph of G is called a *clique* in G. We simply refer a clique as the set of vertices by which it is induced. For cliques C and D in G, if $C \subset D$, then D is said to be an *extension* of C. For a clique C in G, if there exists no extension of C, then C is said to be *maximal*. A maximal clique with the largest size is especially called a *maximum clique*.
Let $\mathcal{O}$ be a set of *objects* (or individuals) and $\mathcal{F}$ a set of *features* (or attributes). For a binary relation $R \subseteq \mathcal{O} \times \mathcal{F}$, a triple $< \mathcal{O}, \mathcal{F}, R >$ is called a *formal context*. If $(x, f) \in R$, we say that

the object o has the feature f. Then, for an object $o \in O$, the set of features associated with o is denoted by $F_R(o)$, that is, $F_R(o) = \{f \in \mathcal{F} \mid (o, f) \in R\}$.
Given a formal context $< \mathcal{O}, \mathcal{F}, R >$, for a set of objects $X \subseteq \mathcal{O}$ and a set of features $Y \subseteq \mathcal{F}$, we define two mappings $\varphi : 2^{\mathcal{O}} \to 2^{\mathcal{F}}$ and $\psi : 2^{\mathcal{F}} \to 2^{\mathcal{O}}$ as follows:

$$\begin{aligned} \varphi X &= \{f \in \mathcal{F} \mid \forall o \in X, f \in F_R(o)\} = \bigcap_{o \in X} F_R(o) \quad \text{and} \\ \psi Y &= \{o \in \mathcal{O} \mid Y \subseteq F_R(o)\}. \end{aligned}$$

That is, the former computes the set of features shared by every object in o. The latter, on the other hand, returns the set of objects with Y.
Based on these mappings, for a set of objects $X \subseteq \mathcal{O}$ and a set of features $Y \subseteq \mathcal{F}$, a pair of X and Y, (X, Y), is called a *formal concept* (or simply concept) under the formal context if and only if $\varphi X = Y$ and $\psi Y = X$, where X and Y are called the *extent* and the *intent* of the concept, respectively. From the definition, it is easy to see that $\psi\varphi X = X$ and $\varphi\psi Y = Y$. That is, a formal concept is defined as a pair of *closed* sets of objects and features under the mappings. Thus, the compound mappings, $\psi\varphi$ and $\varphi\psi$, define *closure operators*.
For a set of objects X, we can *uniquely* obtain a formal concept defined as $(\psi\varphi X, \varphi X)$. Dually, $(\psi Y, \varphi\psi Y)$ is a formal concept uniquely defined for a set of features Y.
Let (X, Y) and (X', Y') be formal concepts. If $X \subseteq X'$ (or $Y \supseteq Y'$), then we say (X, Y) *precedes* (X', Y') and denote it by $(X, Y) \preceq (X', Y')$. Under the ordering, the set of formal concepts in a formal context forms a lattice, called a *concept lattice*.

3. Pinpoint Clustering of Web Pages with Pseudo-Clique Search

In this section, we discuss a method of finding useful clusters of Web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages (Haraguchi & Okubo, 2006; Okubo et al., 2005). Since we are usually careless of pages with lower ranks, they are unconditionally discarded even if their contents are similar to some pages with high ranks. We try to extract such hidden pages together with significant higher-ranked pages as a cluster.
In order to obtain such clusters, we first extract semantic correlations among terms by applying *Singular Value Decomposition*(SVD) to the term-document matrix generated from a corpus w.r.t. a specific topic. Based on the correlations, we can evaluate potential similarities among Web pages from which we try to obtain clusters. The set of Web pages is represented as a weighted graph G based on the similarities and their ranks. Our clusters can be found as *pseudo-cliques* in G. We present an algorithm for finding Top-N weighted pseudo-cliques. Our experimental result shows that quite valuable clusters can be actually extracted according to our method.

3.1 Semantic Similarity among Web Pages

In order to find clusters of Web pages, we have to measure similarities among Web pages. For the task, we follow a technique in *Information Retrieval*(IR) (Salton & McGill, 1983).
Let $\mathcal{D}$ be a set of documents and $\mathcal{T}$ the set of terms appeared in $\mathcal{D}$. We first remove too frequent and too infrequent terms based on $\mathcal{T}$. The set of remaining terms, called *feature terms*, is denoted by $\mathcal{T}^*$. Supposing $|\mathcal{T}^*| = n$, each document $d_i \in \mathcal{D}$ can be represented as an n-dimensional document vector $\boldsymbol{d_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the term $t_j \in \mathcal{T}^*$ in the document d_i. Thus, $\mathcal{D}$ can be translated into a *term-document matrix* $(\boldsymbol{d_1}, \ldots, \boldsymbol{d}_{|\mathcal{D}|})$.

For the term-document matrix, we apply *Singular Value Decomposition*(SVD) in order to extract correlations among feature terms (Moens, 2000).
An $m \times n$ matrix A can be decomposed by applying SVD as $A = U\Sigma V^T$, where U and V are $m \times m$ and $n \times n$ orthogonal matrices, respectively. Each column vector in U (V) is called a left (right) singular vector. Σ is an $m \times n$ matrix of the form

$$\Sigma = \left[\begin{array}{c|c} \begin{array}{ccc} \sigma_1 & & O \\ & \ddots & \\ O & & \sigma_r \end{array} & O_{r\times(n-r)} \\ \hline O_{(m-r)\times r} & O_{(m-r)\times(n-r)} \end{array}\right],$$

where $rank(A) = r$ ($r \leq \min\{m, n\}$) and σ_i is called a *singular value*. First r left singular vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_r}$ correspond to a orthonormal basis and define a new subspace of the original one in which column vectors of A exist, where the $m \times r$ matrix $(\boldsymbol{u_1}, \ldots, \boldsymbol{u_r})$ is denoted by U_r.
Let us assume the matrix A is a term-document matrix generated from a set of documents. Intuitively speaking, by applying SVD to A, we can capture *potential but not presently evident* correlations among the terms. Highly semantically correlated terms give a base vector $\boldsymbol{u_i}$ and define a dimension corresponding to a compound term. Such new base vectors define a new subspace based on compound terms. For documents not in A, therefore, if they are projected on the subspace, we can find similarity among them based on the semantic correlations among terms captured from the original documents in A.
In order to take such semantic similarities of Web pages into account, we prepare a *corpus* of documents written about some specific topic. Then by applying SVD to the term-document matrix generated from the corpus, we obtain a subspace reflecting semantic correlations among terms in the corpus. Let U_r be the orthonormal basis defining the subspace. In IR, we do not always use r left singular vectors. A part of them, that is, $U_k = (\boldsymbol{u_1}, \ldots, \boldsymbol{u_k})$ ($k < r$) is usually used for *approximation*. Such an approximation with U_k is called *Latent Semantic Indexing* (LSI) (Kowalski & Maybury, 2000).
Besides the corpus, with some keywords related to the corpus topic, we retrieve a set of Web pages $\mathcal{P}$ from which we try to obtain clusters. Using the same feature terms for the corpus, each document $p_i \in \mathcal{P}$ is represented as a vector $\boldsymbol{p_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the feature term t_j in p_i. Then each Web page $\boldsymbol{p_i}$ is projected on the subspace as

$$\boldsymbol{p_i^r} = U_r^T \boldsymbol{p_i}.$$

A similarity between Web pages p_i and p_j, denoted by $sim(p_i, p_j)$, is defined based on the standard *cosine measure*, that is,

$$sim(p_i, p_j) = \frac{\boldsymbol{p_i^r} \cdot \boldsymbol{p_j^r}}{\| \boldsymbol{p_i^r} \| \times \| \boldsymbol{p_j^r} \|}.$$

3.2 Finding Clusters of Web Pages by Top-N Pseudo-Clique Search

3.2.1 Graph Representation of Web Pages

Let $\mathcal{P}$ be a set of Web pages from which we try to extract clusters. In order to find our clusters, $\mathcal{P}$ is represented as an undirected weighted graph G.

Assume we computed the semantic similarities among pages in $\mathcal{P}$ according to the procedure just discussed above. Let δ be a similarity threshold. Each page $p_i \in \mathcal{P}$ corresponds to a vertex in G. For any Web pages $p_i, p_j \in \mathcal{P}$, if $sim(p_i, p_j) \geq \delta$, then they are connected by an edge. Furthermore, we assign a weight to each vertex (page) based on its rank, where a higher-ranked page is assigned a larger weight. The weight of a page p is referred to as $w(p)$.

3.2.2 Top-N Weighted Pseudo-Clique Problem

Our cluster of similar pages can be obtained as a weighted *pseudo-clique* in the graph G. In fact, we obtain only nice clusters by extracting maximal weighted pseudo-cliques whose evaluation values are in the top-N. Before giving the problem description, we first define the notion of pseudo-cliques.

Definition 1. (Pseudo-Clique)
Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a class of maximal cliques in a graph. $pseudo(\mathcal{C}) = \cup_{C_i \in \mathcal{C}} C_i$ is called a *pseudo-cliques* with the overlap degree $overlap(\mathcal{C})$ which is defined as $overlap(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \left\{ \left| \cap_{C_j \in \mathcal{C}} C_j \right| / |C_i| \right\}$, where $\bigcap_{C_j \in \mathcal{C}}$ is called the *core*. Moreover, its *size* and *weight* (evaluation value) are given by $|pseudo(\mathcal{C})|$ and $w(pseudo(\mathcal{C})) = \Sigma_{v \in pseudo(\mathcal{C})} w(v)$, respectively. Note here that the weight of pseudo-clique is not restricted to the sum of vertex weights. Any monotone weight under the set inclusion can be accepted. ■

Our problem of finding Top-N weighted pseudo-cliques is defined as follows.

Definition 2. (Top-N Weighted Maximal τ-Valid Pseudo-Clique Problem)
Let G be a graph and τ a threshold for overlap degree. The *Top-N Weighted Maximal τ Pseudo-Clique Problem* is to find any maximal pseudo-clique in G such that its overlap degree is greater than or equal to τ [1] and its weight is in the top N. ■

3.2.3 Computation of Top-N Weighted Pseudo-Cliques

Let $G = (V, E)$ be an weighted graph we are concerned with. In our search, for a clique Q in G, we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q.
Let $cand(Q)$ be the set of vertices v adjacent to any vertex in Q, that is, $cand(Q) = \{v \in V \mid \forall w \in Q \quad (v, w) \in E\}$. Then, we can easily observe that for any pair of cliques Q and Q' in G such that $Q \subseteq Q'$, $cand(Q) \supseteq cand(Q')$ and $w(Q) + w(cand(Q)) \geq w(Q') + w(cand(Q'))$ hold. Note here that the weight of a pseudo-clique with the core Q is *at most* $w(Q) + w(cand(Q))$. Therefore, a simple theoretical property can be easily observed.

Observation1 : Let Q be a clique. Assume we already have *tentative* Top-N maximal pseudo-cliques and the minimum weight of them is w_{min}. If $w(Q) + w(cand(Q)) < w_{min}$ holds, then for any Q' such that $Q' \supset Q$, there exists no pseudo-clique with the core Q' whose weight is in the top N.

Assume that a τ-valid pseudo-clique $\tilde{C}$ contains a clique Q as its core. $\tilde{C}$ can be obtained as the union of any maximal clique C such that $Q \subset C$ and $|Q|/|C| \geq \tau$. It should be noted here that for such a clique C, there exists a maximal clique D in $G(cand(Q))$ such that $Q \cup D = C$, where $G(cand(Q))$ is the subgraph induced by $cand(G)$. That is, finding any maximal clique D in $G(cand(Q))$ such that $|Q|/(|Q| + |D|) \geq \tau$ is sufficient to obtain the pseudo-clique $\tilde{C}$. Although one might claim that such a task is quite expensive from the computational point of view, we can observe some theoretical properties from which pruning rules can be derived.

[1] Such a pseudo-clique is said to be τ-*valid*.

Observation2 : For a clique Q in G, let us assume that we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q. For a clique D in $G(cand(Q))$, if $|D| > (\frac{1}{\tau} - 1) \cdot |Q|$, then any extension (superset) of D is useless for obtaining $\tilde{C}$.

Observation3 : For a clique Q, $Q \cup cand(Q)$ is a τ-valid maximal pseudo-clique with the core Q, if

- $(\frac{1}{\tau} - 1) \cdot |Q| \geq k$ holds, where k is an upper bound of the maximum clique size in $G(cand(Q))$ and
- for any $v \in cand(Q)$, its degree in $G(cand(Q))$ is less than $|cand(Q)| - 1$.

Upper bounds for the maximum clique size have been widely utilized in efficient depth-first branch-and-bound algorithms for finding maximum cliques (Fahle, 2002; Tomita & Seki, 2007). The literature (Fahle, 2002) has argued that the (*vertex*) *chromatic number* χ can provide the tightest upper bound. However, since identifying χ is an NP-complete problem, approximations of χ are usually computed (Fahle, 2002; Tomita & Seki, 2007).
Based on the above observations, Top-N τ-valid weighted pseudo-cliques can be extracted with a *depth-first hybrid search*. For each core candidate Q, its surroundings are explored by finding maximal cliques in $G(cand(Q))$. In the search for core candidates, we can enjoy a pruning based on Observation1. In the surroundings search, a pruning based on Observation2 can be applied. Furthermore, for some core candidates, our surroundings search can be skipped based on Observation3. More precise description of our algorithm is found in (Haraguchi & Okubo, 2006).

3.3 Experimental Results

In this section, we present our experimental results. The main purpose of this experimentation is to confirm that we can actually obtain a useful cluster of Web pages consisting of higher-ranked pages and any other similar (or related) pages with lower ranks. Our system has been implemented in C language and run on a PC with Xeon-2.40 GHz CPU and 512MB memory.

3.3.1 Datasets and Graph Construction

In order to capture semantic correlations among terms, we have prepared a Japanese corpus constructed from 100 Web pages written about "Hokkaido". These pages have been manually selected and only visible texts on them have been manually gathered. After an application of *Morphological Analysis*, we have obtained 2,224 nouns appeared in the corpus. Nouns with frequencies more than 1,000 and less than 2 have been removed from them. The remaining 211 nouns were regarded as feature terms. Applying SVD to the term-document matrix constructed from the corpus, we have obtained a new 98-dimensional subspace.
Besides the corpus, we have retrieved 829 (Japanese) Web pages by Google with the keywords "Hokkaido" and "Sightseeing". We have tried to extract significant clusters from these pages. Each Web page has been first represented as a document vector w.r.t. the original feature terms and then projected on the 98-dimensional subspace in order to capture potential similarities among pages. For any pair of pages, then, we have evaluated the similarity between them based on the cosine measure. Under the setting of $\delta = 0.95$, we have constructed a weighted graph G from the pages. That is, if the angle between two pages is less than or equal to about 18.2 degree, then they are connected by an edge. The numbers of vertices and edges are 829 and 798, respectively. Each page (vertex) d has been assigned a weight defined as $w(d) = 1/rank(d)^2$. As has been stated in the previous section, although we can define

Page Rank	Subject
11^{th}	Index page for travel information maintained by a local travel agency in Hokkaido (especially, for travels in Hokkaido)
382^{th}	Index page for travel information maintained by a famous newspaper company (for domestic and overseas travels)
416^{th}	An article on a private BBS for travels
797^{th}	Information about smorgasbords enjoyable at a hotel in Hokkaido
798^{th}	Information about smorgasbords enjoyable at another hotel in Hokkaido
826^{th}	Page for hotel awards in a famous travel site

Table 1. The 11^{th} significant cluster

various weights according to ranks of pages, we have currently adopted the reciprocal of the rank squared. The reason why we prefer this measure is as follows:

- It is sensitive to difference of ranks in higher range of ranks.
- On the other hand, in lower range, page weights are hardly affected by difference of ranks.

From the characteristics, a clique containing higher-ranked pages is likely to be extracted even if its size is relatively small. Since we can often expect higher-ranked pages are significant, such a phenomenon would be desirable. On the other hand, we are usually careless of lower-ranked pages. In other words, difference of weights among lower-ranked pages would be unimportant for us. In this sense, a likelihood of extracting pseudo-cliques should not be sensitively affected by weights of pages with lower ranks. The above measure would be reasonable from this viewpoint as well.

3.3.2 Example of Extracted Interesting Cluster

We have tried to extract Top-15 weighted 0.8-pseudo cliques in the graph constructed above. Among the extracted clusters (pseudo-cliques), the authors especially consider that the 11^{th} cluster is quite interesting.

The cluster consists of 6 Web pages. Table 1 shows their ranks assigned by Google and subjects. In the authors' opinion, their contents are considered to be very similar in the sense that all of them give us some information about accommodations in Hokkaido, especially information about hotels and foods. The 11^{th} and 382^{th} pages are index pages for travel information and we can make reservations for many hotels via the pages. The 416^{th} page is an article in a private BBS site for travels. The article reports on a private travel in Hokkaido and provides an actual and valuable information about a hotel and enjoyable foods in "Furano" [2]. The 797^{th} and 798^{th} personal pages give us the names of two hotels serving smorgasbords in Hokkaido. The 826^{th} page tells us several hotels which were the most popular or were most frequently reserved in 2004.

Thus, the pages in the 11^{th} cluster are closely related each other and give us quite valuable information. When we try to make travel plans for sightseeing in Hokkaido, we would often care about hotels and foods as important factors. In such a case, the cluster will be surely helpful for us.

Needless to say, we can find clusters of Web pages by *exact* clique search. In that case, however, the above 11^{th} cluster can never be obtained. The cluster as a pseudo-clique consists of two

[2] "Furano" is one of the most famous sightseeing areas in Hokkaido.

exact maximal cliques: $\{11^{th}, 382^{nd}, 797^{th}, 798^{th}, 826^{th}\}$ and $\{382^{nd}, 416^{th}, 797^{th}, 798^{th}, 826^{th}\}$. In the exact case, the former can be ranked as 11^{th}, whereas the latter cluster as 343^{rd}. It should be noted that the 416^{th} page will be invisible unless we specify a large N for Top-N. However, it would be impractical to specify such a large N because many clusters are undesirably extracted. Although 416^{th} page has valuable contents as mentioned above, we will lose a chance to browse it.
In case of pseudo-clique search, the 343^{rd} exact cluster can be absorbed into the 11^{th} cluster to form a pseudo-clique. In other word, the 343^{rd} cluster can be drastically raised its rank. As the result, 416^{th} page can become visible by just specifying a reasonable N.
Thus, our chance to get significant lower-ranked pages can be enhanced with the help of pseudo-cliques. This is a remarkable advantage brought by pseudo-cliques.

3.3.3 Computational Performance of Pseudo-Clique Search

Our experimental result also shows that the pruning rules presented in the previous section are very effective. The number of cores actually examined was 69,981 and our pruning based on the tentative minimum weight were invoked at 40,801 nodes of them. Moreover, the maximal clique searches were skipped at 31 nodes. Thus, the pruning rules can be applied very frequently in our search. As the result, the total computation time was just 0.847 second.
As we have experienced, an IR system often retrieves over hundreds of thousands of Web pages. Therefore our graph constructed from gathered Web pages would have a large number of vertices in more practical situation. In general, however, our graph tends to be quite sparse. Therefore, it is expected that our algorithm can still work well even in such a practical case.
From the experimental result, the authors consider that our pseudo-clique search would be a promising approach to finding significant clusters of Web pages.

4. From Clique-Based Clusters to Formal Concept-Based Clusters

As has been shown just above, we can extract an interesting cluster of Web pages with pseudo-clique search. In the area of *conceptual clustering* (Hotho et al., 2003; Hotho & Stumme, 2002), however, it has been pointed out that as long as the similarity of documents is based on 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 to may differ. In order to avoid such a conceptually indistinct argument, we have made 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 documents. 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 a document set is a document-term relationship, we have adopted the notion of *Formal Concept Analysis* (FCA) (Ganter & Wille, 1999; Ganter et al., 2005). Thus, if some higher-ranked documents and lower-ranked ones share a set of terms, they could form the extent of a formal concept, that is, a conceptual cluster of documents.
It is 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* (Uno et al., 2004) 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 exist a numerous number of extents of concepts whose corresponding intents are very smaller set of terms. For 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 the reason, we have tried to find only Top-N extents w.r.t. the same evaluation schema for clusters, keeping the quality of their intents (Haraguchi & Okubo, 2007; Okubo & Haraguchi, 2006). The method is summarized as follows:

Evaluation on Extents

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.

Graph Formation under Static Quality Control on Intents

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 a weighted undirected graph of documents, where each document is assigned a weight based on its rank. It should be noted here that any extent with enough quality of intent is always a clique in the graph.

Extent Search under Dynamic Quality Control

To enumerate only Top-N extents (that is, closures of documents), our algorithm adopts again a branch-and-bound method, where

Candidate Closures of Documents: a list of candidate top-N closures is always kept,

Branch-and-Bound Pruning due to Monotone Evaluation: for any search node, a closure of documents, whose evaluation value can never become larger than the minimum of those candidates, we cut off the nodes below, and

Dynamic Quality Control: for 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 in the sense that we never miss any of Top-N extents satisfying the requirements.
In the graph formation process, we can exclude document pairs *in advance* which are never included in any extent with enough quality of intent. Furthermore, a theoretical property of cliques can provide us several *upper-bounds* of evaluation values for extents. For example, we can obtain a tight upper-bound with a *sequential approximate coloring* (Fahle, 2002; Tomita & Seki, 2007). Based on the bounds, we can prune many useless extents which are never in Top-N. Thus, the clique search-based approach enables us to efficiently find Top-N extents.

5. Finding Implicit Groups of Web Pages as Constrained Top-N Formal Concepts

In this section, we present a method for finding relatively smaller therefore more implicit groups of Web pages as formal concepts and discuss an effective depth-first mining algorithm for them (Li et al., 2008). The algorithm is based on a dynamic ordering method depending on each search node and some search tree expansion rules. Moreover it is designed so as to find Top-N implicit concepts subject to the size restriction and some space constraints reflecting user's interests.

5.1 Problem Specification

For a given formal context $< \mathcal{O}, \mathcal{F}, R >$, we suppose $\mathcal{O}$ and $\mathcal{F}$ represent the set of pages (documents) and a set of their feature terms, respectively. Then, the set of terms possessed by every page in $X \subseteq \mathcal{O}$ is denoted as φX. Conversely, ψA is a set of pages with every term in $A \subseteq \mathcal{F}$. The actual construction of φ and ψ from Web pages is described in Section 5.3.
The only fact remarked here is that φX and ψA are an intent and an extent for any set $X \subseteq \mathcal{O}$ and $A \subseteq \mathcal{F}$, respectively. Since a formal concept is defined as a pair of extent X and its corresponding intent φX, we identify the concept with its extent (or its intent).
We suppose in addition a pair of monotone evaluation functions $eval_{\mathcal{O}}$ and $eval_{\mathcal{F}}$ such that $eval_{\mathcal{O}}(X_1) \leq eval_{\mathcal{O}}(X_2)$ whenever $X_1 \subseteq X_2$ and $eval_{\mathcal{F}}(A_1) \leq eval_{\mathcal{F}}(A_2)$ if $A_1 \subseteq A_2$. Their most simple forms are set sizes which we assume simply in this chapter. Another forms of *eval* can be found in (Haraguchi & Okubo, 2006) including rank information of Web pages.
Now, our problem of finding implicit concepts is described as follows:

Definition 3. (Top-N Implicit Concept Problem)
For a formal context $< \mathcal{O}, \mathcal{F}, R >$,

Objective: Enumerate every solution extent X with top N evaluation value $eval_{\mathcal{O}}(X)$, where they must be subject to the followings:

Length Constraint (required): Given $\delta > 0$, $eval_{\mathcal{F}}(\varphi X) \geq \delta$ for excluding larger X .

Space Constraints (option): X must satisfy

(POS) $S^+ \subseteq X$ for an example page set S^+,

(NEG) $S^- \cap X = \phi$ for a negative page set S^-, and

(SUB) $X \subseteq \psi K$ for a relevant term set K. ■

5.2 Efficient Computation of Implicit Concepts

5.2.1 Basic Search Strategy

Given a formal context $\mathcal{C} =< \mathcal{O}, \mathcal{F}, \mathcal{R} >$, for each formal concept under $\mathcal{C}$, there always exists a set of objects $X \subseteq \mathcal{O}$ such that $\psi\varphi X$ and φX correspond to the extent and the intent of the concept, respectively. Therefore, by applying the mappings φ and ψ to each set of objects $X \subseteq \mathcal{O}$, we can completely obtain all of the concepts under $\mathcal{C}$.
From the monotonicity of the evaluation function $eval_{\mathcal{F}}$, a simple theoretical property can be observed. Let X_i and X_j be sets of objects in $\mathcal{O}$ such that $X_i \subseteq X_j$. Then, $eval_{\mathcal{F}}(\varphi X_i) \geq eval_{\mathcal{F}}(\varphi X_j)$. As a direct consequence, a pruning rule is available in our search. That is, for a set of objects $X \subseteq \mathcal{O}$, if $eval_{\mathcal{F}}(\varphi X) < \delta$, then there is no need to examine any superset of X. Therefore, our search for finding target concepts can be performed in *depth-first manner* with the simple pruning.
During our search, we maintain a list which stores Top-N concepts already found. That is, the list keeps *tentative* Top-N concepts. For a set of objects $X \subseteq \mathcal{O}$, we check whether $eval_{\mathcal{F}}(\varphi X) \geq \delta$ holds or not. If it holds, then $(\psi\varphi X, \varphi X)$ becomes a concept satisfying the length constraint under δ and the tentative Top-N list is adequately updated for the concept. Then a child of the extent φX, $\varphi X \cup \{x\}$, is generated by expanding the extent with an object $x \in \mathcal{O} \backslash \varphi X$ and the same procedure is recursibly performed for the child. If $eval_{\mathcal{F}}(\varphi X) < \delta$, we can immediately backtrack to examine another search branches. Starting with the initial X of the empty set, the procedure is iterated in depth-first manner until no X remains to be examined.

When common terms of Z appear as terms shared by W (that is, $\varphi Z \subseteq \varphi W$), we here say that Z *implies* W and write as $Z \rightarrow W$. Then, the extent of a concept is defined as a set X such that $X = \{x \mid X \rightarrow \{x\}\}$. That is, the extent is closed under (object) implication, and is called a *closure* (or *closed set*). Similarly, intent A of terms is similarly defined using (attribute) implication (Ganter & Wille, 1999).
The constraint (POS) is requiring $I = \{z \mid S^+ \rightarrow \{z\}\} \subseteq X$. Hence S^+ defines the starting extent I in our depth-first search. The constraint (SUB) assigns an upper bound closure ψK, and is equivalent to $K \subseteq \varphi X$ meaning that X must have every term in K which users show their interests. By (POS) and (SUB), a sublattice with I and ψK as the least and the greatest closures, respectively, is formed. When (POS) is not presented, S^+ is just the bottom extent of whole concept lattice. Similarly, we treat other constraint types in the same manner when they are not explicitly presented.

5.2.2 Dynamic Ordering in Expansion Process

Although we are allowed to restrict the search space by the constraints, it is a key to have an effective enumeration method of concepts when the optional constraints are not presented explicitly or when the data in the form of page-term relationship scales up. For this reason, we introduce a *dynamic ordering of candidates* and a search tree expansion rule customized to it.

Definition 4. (Candidate Page)
Let X be a present extent consistent with the given constraints. Then, a page $x \notin X$ is called a *candidate* at X if the enlarged extent, $\psi\varphi(X \cup \{x\}) = \{z \mid X \cup \{x\} \rightarrow \{z\}\}$, still satisfies the constraints. ■

Some candidate z at X cannot be a candidate at $\psi\varphi(X \cup \{x\})$ if $\{w \mid X \cup \{x, z\} \rightarrow \{w\}\}$ violates the constraints. Thus the sequence of candidate sets is monotonically decreasing as we add new candidates to the closure extents.

Dynamic Candidate Ordering: For a present extent X and its candidate x, x is a branch to form the next extent. We arrange candidates x in the increasing order of the sizes of term sets $\varphi(X \cup \{x\})$. The ordering is locally fixed at each X. So we denote it as $\prec_X$.

When the candidate x is actually chosen at X, another y s.t. $X, x \rightarrow y$ is included together with x into the next closure. As x has smaller term set at X, it has more chances to imply such additional y. This helps us to form larger next closures earlier.

5.2.3 Prunings with Right and Left Candidates

Now, based on the dynamic ordering strategy, we expand our search tree. The root node is $\{z \mid S^+ \rightarrow \{z\}\}$. The procedure expands tree nodes in the depth-first manner by selecting a candidate at each node according to the dynamic ordering. The sequence of chosen candidates $c_1, ..., c_k$ represents the path from the root to the extent $\{z \mid S^+ \cup \{c_1, ..., c_k\} \rightarrow \{z\}\}$. Thus a path with S^+ is just a generator (Lakhal & Stumme, 2005) of the extent. Unlike a set enumeration tree, some control to avoid duplicated generations of the same extents is needed, as there exist several generators for the same extents. For this reason, we classify candidates into two types. One is called a *right candidate* used for expansion. The other is called a *left candidate* used for checking the duplication. Suppose we have a series of extents $X_k = \{z \mid S^+ \cup \{c_1, ..., c_k\} \rightarrow \{z\}\}$, where c_k is a chosen candidate at X_{k-1} to form X_k. That is, $X_k = \{z \mid S^+ \cup X_{k-1} \cup \{c_k\} \rightarrow \{z\}\}$. Then a candidate r at X_k is called a left candidate, given a chosen candidate c_{k+1} at X_k to form X_{k+1}, if $r \in \{c_1, \cdots, c_k\}$ or $r \prec_{X_{k+1}} c_{k+1}$.

With the help of right and left candidates, we can enjoy the following prunings in our search process.

Inverse Implication Pruning: For a present extent X and its right candidate r, if $X \cup \{r\} \to \ell$ holds for some left candidate ℓ at X, we need not take the branch by r.

Branch-and-Bound Pruning: For a present X and a right candidate r, we skip the branch by r whenever the evaluation value of $(X_r = \{w \mid X \cup \{r\} \to \{w\}\}) \cup \{\text{right candidate at } X_r\}$ by $eval_{\mathcal{O}}$ is less than the minimum of the current top N values. When the number of values stored is less than N, this rule is void.

The algorithm repeats the tree expansion on a path in a depth-first manner, using the above pruning rules, and goes back to its parent node to try another right candidate at the parent node, whenever the remaining right candidate set becomes empty.

5.3 Experimental Results

We present here our experimental results. Our system has been implemented in JAVA and run on a PC with Dual-Core AMD Opteron processor 2222 SE and 16GB main memory.

5.3.1 Dataset

In our experimentation, we have tried to extract Top-N clusters from a dataset called *BankSearch*.

The dataset *BankSearch* has been released as a benchmark for Web document clustering (Sinka & Corne, 2002). It consists of Web documents (HTML sources) in 11 categories, "*Commercial Banks*", "*Building Societies*", "*Insurance Agencies*", "*Java*", "*C/C++*", "*Visual Basic*", "*Astronomy*","*Biology*", "*Soccer*", "*Motor Sport*" and "*Sport*". The total number of documents is 11,000 (1,000 documents for each category).

As a preprocess, we have first converted each HTML source into a plane text by removing HTML tags. From the text documents, adjectives and adverbs in `WordNet` (Fellbaum, 1998) have been eliminated. Furthermore, we have removed a set of stopwords as well. After *Stemming Process* with Porter stemmer (Porter, 1980), we have selected 1,223 words as feature terms by removing too frequent and too infrequent ones. That is, each document can be represented as a 1223-dimensional vector. It should be emphasized here that the category informations never appears in the documents as features explicitly.

5.3.2 Extracted Clusters

We present here some clusters we have actually extracted based on our method.

Given a Web page,

`http://www.vbsquare.com/files/association/`,

as a positive example, we have tried to find Top-3 concepts under $\delta = 50$. As an example, a concept

```
({http://www.vbsquare.com/files/association/,
  http://www.vbsquare.com/registry/tip471.html,
  http://www.vb-helper.com/links.htm,
                  ⋮
  http://www.vbsquare.com/databases/dbclass/,
  http://www.vbsquare.com/databases/learndb/,
```

```
        http://www.vbsquare.com/mouse/context/ },
      { API, component, resource, ...tips, VB, graphic } )
```

consisting of 35-pages has been extracted. All of the pages are related to resource links, tutorials and stories on *Visual Basic*. They belong to the same category assigned in (Sinka & Corne, 2002). It should be noted here that our method never uses the information about the categories explicitly. Our clusters are extracted based on only terms appearing in Web pages. Thus, without the category information, our method can extract clusters which are consistent with the known categories.
Given two Web pages,

`http://www.citibank.com/uk/portal/consumer/helpdesk/tc/tc1.htm` and
`http://vbtechniques.com/useragreement.asp`,

and two terms, `claim` and `Internet`, as positive examples and relevant terms, respectively, we have tried to find Top-1 concepts under $\delta = 50$, then obtained a concept

```
( { http://www.citibank.com/uk/portal/
              consumer/helpdesk/tc/tc1.htm,
    http://vbtechniques.com/useragreement.asp,
    http://www.hrbs.co.uk/cashisatandcapply.htm,
                    ⋮
    http://www.hrbs.co.uk/panthertandconline.htm,
    http://www.hrbs.co.uk/rewardsixtandcapply.htm,
    http://www.lloyds.com/un/en/
              termsandconditions/category/article/ },
  { claim, Internet, accept, ...
         law, condition, reason, right, term, transfer } )
```

consisting of 22-pages. These pages are concerned with contracts and terms of agreement. Furthermore, since they belong to different categories, "*Commercial Banks*", "*Visual Basic*", "*Building Society*" and "*Insurance Agency*", we consider that it is a concrete example of crossover concepts actually obtained with our method.
Thus, our Top-N method has an ability to flexibly extract various concepts reflecting our interests represented as positive example and relevant terms.

5.3.3 Computational Performance

Finding Formal Concepts by Closed Itemset Miners:
As has been mentioned previously, formal concepts can be obtained by any closed itemset miner, e.g. `LCM` (Uno et al., 2004). Such a system is, however, not always helpful for finding our Top-N formal concepts satisfying some constraints. More concretely speaking, in order to find our Top-N formal concepts, a closed itemset miner must first enumerate frequent closed itemsets including our targets and then choose the targets from them. However, the miner often enumerates a huge number of frequent closed itemsets, taking long computation time.
Figure 2 shows the computation time by `LCM` and the number of frequent closed itemsets under various minimum support thresholds (*minsup*) for the *BankSearch* dataset, regarding each feature term as an item. The figure tells us that for lower *minsup* values, extracting Top-N concepts with `LCM` would be impractical from the viewpoint of its computation time

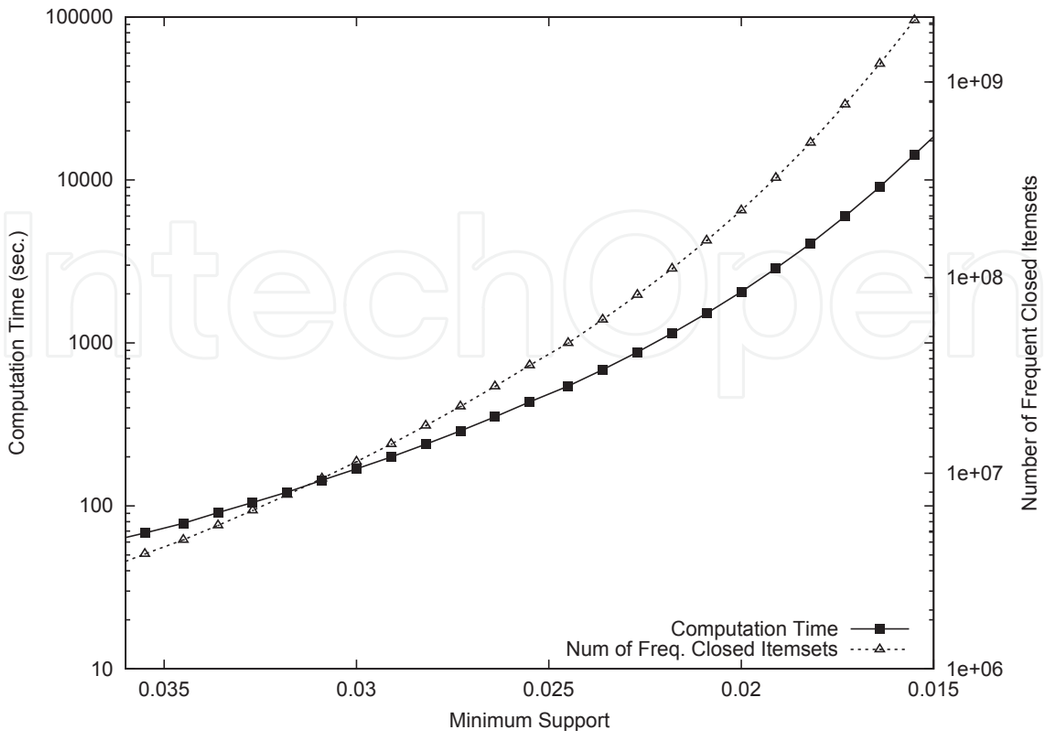

Fig. 2. Computation Time by LCM and Number of Frequent Closed Itemsets

and output size. For example, the setting of $minsup = 0.015$ forces us to extract all concepts consisting of at least 165 documents. Therefore, any smaller concepts (say, below a hundred) can never be obtained with the help of closed itemset miners in practice. More concretely speaking, the extent of each concept just presented above consists of 35-pages and 22-pages, respectively. In order to obtain the former concept with a *minsup*-based closed itemset miner like LCM, therefore, we have to set $minsup = \frac{35}{11000} = 0.003$. For the latter, $minsup = \frac{35}{11000} = 0.002$. Needless to say, our targets are out of range for which such a miner can compute. Thus, our Top-N method can extract targets actually intractable for *minsup*-based itemset miners. This is a remarkable advantage of our Top-N method.

Effectiveness of Positive Examples, Relevant Terms and Dynamic Ordering:
Since positive examples and relevant terms restrict the search space, our computational cost can be reduced. In addition, our dynamic ordering on candidate expansions also achieves improvement in computation time. For the same positive examples and relevant terms, their effectiveness is verified in Figure 3. In the figure, we can easily observe that they are quite effective in improving our computational efficiency. We can enjoy significant improvement with them. Although the positive examples can solely provide a great reduction of computation time, the relevant terms bring us further drastic improvement. Particularly, for lower δ-values, the ratio of computation time with only examples to those with both examples and relevant terms is above 100. It is highly expected that the larger our dataset becomes, the greater difference we will observe. Thus, our method would be promising even for large-scale datasets.

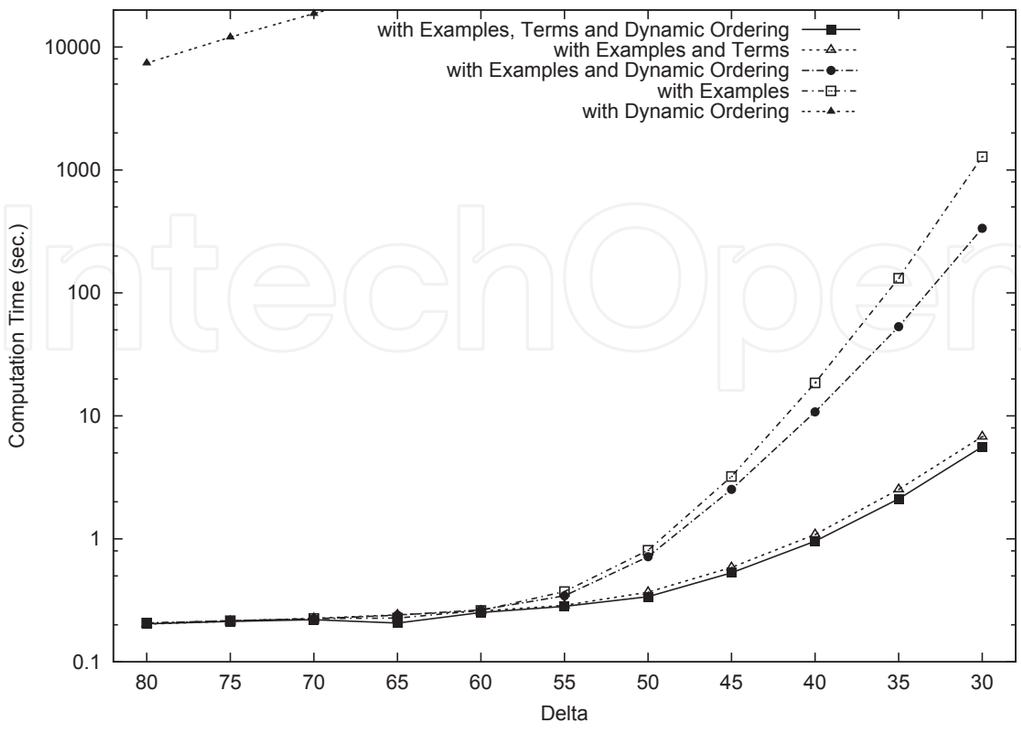

Fig. 3. Computation Time with Positive Examples, Relevant Terms and Dynamic Ordering

6. Conclusion

In this chapter, we presented our Top-N methods for extracting clusters of Web pages, especially, a method for pinpoint clustering of Web pages by pseudo-clique search and a method for finding implicit page groups represented as formal concepts.

In our pinpoint clustering, we first extract semantic correlations among terms by applying SVD to the term-document matrix generated from a corpus w.r.t. a specific topic. Based on the correlations, we can evaluate potential similarities among Web pages from which we try to obtain clusters. The set of Web pages is represented as a weighted graph G based on the similarities and their ranks. Then our clusters are extracted as *pseudo-cliques* in G. Our experimental results showed that a valuable cluster can be actually extracted according to our method.

Turning our attention from clique-based clusters to formal concept-based clusters in order to make our clusters more meaningful, we discussed an effective depth-first mining algorithm for finding relatively smaller therefore more implicit groups of Web pages as formal concepts. The algorithm is based on a dynamic ordering method depending on each search node and some search tree expansion rules. Moreover it was designed so as to find Top-N implicit concepts subject to the size restriction and some space constraints reflecting user's interests. Our experimental results showed that our Top-N algorithm succeeds in finding less frequent (crossover) concepts under some space constraints.

In order to have more effective method under more vague constraints, we are planning to define the notion of crossover concepts more directly and to design more efficient and accurate procedure under the help of clustering of pages allowing outliers (Gan et al., 2007).

7. References

Bastide, Y.; Pasquier, N.; Taouil, R.; Stumme, G. & Lakhal, L. (2000). Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets, *Proceedings of the 1st International Conference on Computational Logic - CL'00*, LNCS 1861, pp. 972–986, ISBN 3540677976, London, U. K., July, 2000, Springer, Berlin.

Bayardo Jr., R. J. (1998). Efficiently Mining Long Patterns from Databases, *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 85–93, ISBN 0-89791-995-5, Washington, D. C., U. S. A., June, 1998, ACM Press, New York.

Burdick, D.; Calimlim, M. & Gehrke, J. (2001). MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases, *Proceedings of the 17th International Conference on Data Engineering - ICDE'01*, pp. 443–452, ISBN 0-7695-1001-9, Heidelberg, Germany, April, 2001, IEEE Computer Society Press, Washington, D. C.

Boulicaut, J. & Jeudy, B. (2005). Constraint-Based Data Mining, *The Data Mining and Knowledge Discovery Handbook 2005*, pp. 399–416, Springer, ISBN 978-0387244358, Berlin.

Fahle, T. (2002). Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique, *Proceedings of the 10th European Symposium on Algorithms - ESA'02*, pp. 485 – 498, LNCS 2461, ISBN 978-3540441809, Rome, Italy, September, 2002, Springer, Berlin.

Fellbaum, C. (1998). *WordNet - An Electronic Lexical Database*, Fellbaum, C. (Ed.), The MIT Press, ISBN 978-0262061971, Cambridge.

Gan, G.; Ma, C. & Wu J. (2007). *Data Clustering – Theory, Algorithms, and Applications*, SIAM, ISBN 978-0898716238, Philadelphia.

Ganter, B. & Wille, R. (1999). *Formal Concept Analysis - Mathematical Foundations*, Springer, ISBN 978-3540627715, Berlin.

Ganter, B.; Stumme, G. & Wille, R. (2005). *Formal Concept Analysis – Foundations and Applications*, Ganter, B., Stumme, G. and Wille, R. (Eds.), LNAI 3626, Springer, ISBN 978-3540278917, Berlin.

Han, J.; Cheng, H.; Xin, D. & Yan, X. (2007). Frequent pattern mining - current status and future directions, *Data Mining and Knowledge Discovery*, Vol. 15, No. 1, pp. 55–86, ISSN 1384-5810.

Haraguchi, M. & Okubo, Y. (2007). An Extended Branch-and-Bound Search Algorithm for Finding Top-*N* Formal Concepts of Documents, *New Frontiers in Artificial Intelligence*, JSAI 2006 Conference and Workshops, Tokyo, Japan, June 5-9, 2006, Revised Selected Papers, pp. 276 – 288, LNCS 4384, Springer, ISBN 978-3540699019, Berlin.

Haraguchi, M. & Okubo, Y. (2006). 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, pp. 59 – 78, LNAI 3847, Springer, ISBN 978-3540310181, Berlin.

Haraguchi, M. (2002). Concept Learning Based on Optimal Clique Searches, *JSAI SIG Report*, pp. 63 – 66, SIG-FAI-A202-11, Fukuoka, Japan, December, 2002, JSAI, Tokyo (in Japanese).

Hotho, A.; Staab, S. & Stumme, G. (2003). Explaining Text Clustering Results Using Semantic Structures, *Proceedings of the 7th European Conference on Principles of Data Mining and*

Knowledge Discovery - PKDD'03, pp. 22 – 26, LNAI 2838, ISBN 3-540-20085-1, Cavtat-Dubrovnik, Croatia, September, 2003, Springer, Berlin.

Hotho, A. & Stumme, G. (2002). Conceptual Clustering of Text Clusters, *Proceedings of the Machine Learning Workshop - FGML'02*, pp. 37 – 45, Hannover, SIG of German Informatics Society.

Kowalski, G. J. & Maybury, M. T. (2000). *Information Storage and Retrieval Systems: Theory and Implementation (Second Edition)*, Kluwer Academic Publishers, ISBN 978-0792379249, Dordrecht.

Lakhal, L. & Stumme, G. (2005). Efficient mining of association rules based on formal concept analysis, *Formal Concept Analysis – Foundations and Applications*, Ganter, B., Stumme, G. and Wille, R. (Eds.), pp. 180–195, LNAI 3626, Springer, ISBN 978-3540278917, Berlin.

Li, A.; Haraguchi, M. & Okubo, Y. (2008). Implicit Groups of Web Pages as Constrained Top-*N* Concepts, *Proceeding of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, pp. 190 – 194, Sydney, Australia, December, 2008, IEEE Computer Society Press, Washington, D. C.

Moens, M-F. (2000). *Automatic Indexing and Abstracting of Document Texts*, Kluwer Academic Publishers, ISBN 978-0792377931, Dordrecht.

Murata, T. (2003). Discovery of Web Communities from Positive and Negative Examples, *Proceedings of the 6th International Conference on Discovery Science - DS'03*, LNAI 2843, pp. 369 – 376, ISBN 978-3540202936, Sapporo, Japan, October, 2003, Springer, Berlin.

Murata, T. (2000). Discovery of Web Communities based on the Co-Occurrence of Reference, *Proceedings of the 3rd International Conference on Discovery Science - DS'00*, LNCS 1967, pp. 65 – 75, ISBN 978-3540413523, Kyoto, Japan, December, 2000, Springer, Berlin.

Okubo, Y. & Haraguchi, M. (2006). Finding Conceptual Document Clusters with Improved Top-N Formal Concept Search, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence - WI'06*, pp. 347 – 351, ISBN 0-7695-2747-7, Hong Kong, China, December, 2006, IEEE Computer Society Press, Washington, D. C.

Okubo, Y.; Haraguchi, M. & Shi, B. (2005). Finding Significant Web Pages with Lower Ranks by Pseudo-Clique Search, *Proceedings of the 8th International Conference on Discovery Science - DS'05*, pp. 346 – 353, LNAI 3735, ISBN 978-3540292302, Singapore, October, 2005, Springer, Berlin.

Page, L.; Brin, S.; Motwani, R. & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web*, http://dbpubs.stanford.edu/pub/1999-66.

Porter, M. F. (1980). An Algorithm for Suffix Stripping, *Program: Electronic Library and Information Systems*, Vol. 14, No. 3, pp. 130 – 137, ISSN 0033-0337.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*, Mcgraw-Hill College, ISBN 978-0070544840, Boston.

Sinka, M. P. & Corne, D. W. (2002). A Large Benchmark Dataset for Web Document Clustering, *Soft Computing Systems: Design, Management and Applications*, Series of Frontiers in Artificial Intelligence and Applications, Vol. 87, pp. 881 – 890, `http://www.pedal.reading.ac.uk/banksearchdataset/`, ISBN 978-1586032975.

Strang, G. (2003). *Introduction to Linear Algebra*, 3rd Edition, Wellesley-Cambridge Press, ISBN 978-0961408893, Massachusetts.

Tomita, E. & Seki, T. (2007). An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique with Computational Experiments, *Journal of Global Optimization*, Vol. 37, No. 1, pp. 95 – 111, ISSN 0925-5001.

Uno, T; Kiyomi, M. & Arimura, H. (2004). LCM ver. 2: Efficient Mining Algorithm for Frequent/Closed/Maximal Itemsets, *Proceedings of IEEE ICDM'04 Workshop on Frequent Itemset Mining Implementations - FIMI'04*, http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-126/.

Wang, J.; Han, J. & Pei, P. (2003). CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'03*, pp. 236 – 245, ISBN 978-1581137378, Washington, D. C., U. S. A., August, 2003, ACM Press, New York.

Zhang, Y.; Yu, J. & Hou, J. (2006). *Web Communities - Analysis and Construction*, Springer, ISBN 978-3540277378, Berlin.

Zhu, F,; Yan, X.; Han, H.; Yu, P. S. & Cheng, H. (2007). Mining Colossal Frequent Patterns by Core Pattern Fusion, *Proceedings of the 23rd International Conference on Data Engineering - ICDE'07*, pp. 706 – 715, ISBN 1-4244-0803-2, Istanbul, Turkey, April, 2007, IEEE Computer Society Press, Washingon, D. C.

Web Intelligence and Intelligent Agents

Edited by Zeeshan-Ul-Hassan Usmani

ISBN 978-953-7619-85-5
Hard cover, 486 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents a unique and diversified collection of research work ranging from controlling the activities in virtual world to optimization of productivity in games, from collaborative recommendations to populate an open computational environment with autonomous hypothetical reasoning, and from dynamic health portal to measuring information quality, correctness, and readability from the web.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Makoto Haraguchi and Yoshiaki Okubo (2010). Pinpoint Clustering of Web Pages and Mining Implicit Crossover Concepts, Web Intelligence and Intelligent Agents, Zeeshan-Ul-Hassan Usmani (Ed.), ISBN: 978-953-7619-85-5, InTech, Available from: http://www.intechopen.com/books/web-intelligence-and-intelligent-agents/pinpoint-clustering-of-web-pages-and-mining-implicit-crossover-concepts

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen