A thesis for the degree of Doctor of Philosophy

Extracting Hierarchical Structure of Web Video Groups for Realizing Advanced Web Video Retrieval

高度なWeb映像検索を実現する Web映像集合の階層構造抽出に関する研究

Ryosuke Harakawa

Graduate School of Information Science and Technology

Hokkaido University

March, 2016

Abstract

Due to the widespread use of video hosting services such as YouTube, more and more Web videos, *i.e.*, video materials on the Web, are being uploaded and accessed by many users. From such a social background, it is necessary to develop a method that enables effective retrieval of desired Web videos. However, there is a limitation that existing search engines require users to input suitable keywords as a query to accurately retrieve the desired Web videos.

To overcome this limitation, this thesis proposes a method for extracting the hierarchical structure of Web video groups on the basis of the relevance between heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos. By utilizing the relevance between heterogeneous features, advanced Web video retrieval becomes feasible. Specifically, the proposed method calculates latent features on the basis of the correlation between visual, audio and textual features. Then the combination use of the latent features and features of link relationships enables extraction of the hierarchical structure of Web video groups. Furthermore, this thesis proposes a method for improving the proposed method, which enables application of the proposed method to many Web videos. The improved method can efficiently calculate latent features of Web videos on the basis of a clustering scheme. Furthermore, a graph, which can handle many Web videos by a small number of nodes, is constructed and efficient extraction of the hierarchical structure becomes feasible. Moreover, this thesis proposes a method for further improving accuracy and efficiency of extraction of the hierarchical structure. Specifically, this method utilizes only local features of link relationships and enables accurate and efficient extraction of the hierarchical structure. Experimental results for Web videos collected by using YouTube have confirmed that the proposed methods in this thesis enable retrieval of desired Web videos even if users cannot input suitable keywords as a query.

Acknowledgments

First, I would like to sincerely thank my supervisor, Prof. Miki Haseyama. This thesis would not have been possible without the invaluable guidance and encouragement she has given me over the four years I spent at the School of Engineering and the Graduate School of Information Science and Technology, Hokkaido University.

I would like to thank Prof. Tsuyoshi Yamamoto, Prof. Yuji Sakamoto and Prof. Kenji Araki for providing insightful comments and suggestions about the research I performed in the Graduate School of Information Science and Technology, Hokkaido University.

I would also like to sincerely thank Assistant Prof. Takahiro Ogawa for the countless hours of assistance and fruitful discussion over the course of performing the work described in this thesis.

Furthermore, I would like to sincerely thank Assistant Prof. Sho Takahashi for his constant encouragement and advice about research and academic life.

Also, I am very grateful to Postdoctoral Fellow Yasutaka Hatakeyama for his tremendous support from the beginning of the research on this thesis.

Finally, I would like to sincerely thank everyone at the Laboratory of Media Dynamics, Graduate School of Information Science and Technology, Hokkaido University for their invaluable support and assistance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The background and purpose of this study and the organization of this thesis are presented in this chapter.

1.1 Background

With the widespread use of video hosting services, more and more Web videos[1] are being uploaded and many users retrieve them to access desired Web videos, *i.e.*, Web videos with topics that users try to find [1, 2]. YouTube[2], the most popular video hosting service in the world, has over a billion users and billions of views are generated every day[3]. From such a social background, it is necessary to develop a method that enables effective retrieval of desired Web videos.

To meet this necessity, many methods based on a query-response model have been proposed [3–5]. In the query-response model, users input a query such as keywords, sketches and videos into search engines, and Web videos corresponding to the input query are returned as ranked lists to users. Recently, the progress of machine learning techniques has led to the development of semantic understanding including object recognition and event detection [6–9], and users can now accurately retrieve Web videos associated with the input query. However, if the user cannot input a suitable query into a search engine, the desired Web videos cannot be obtained unless the user searches low ranked results since the desired Web videos are not

[1] In this thesis, we call video materials on the Web "Web videos".
[2] http://www.youtube.com
[3] http://www.youtube.com/yt/press/statistics.html, last accessed: November 2015.

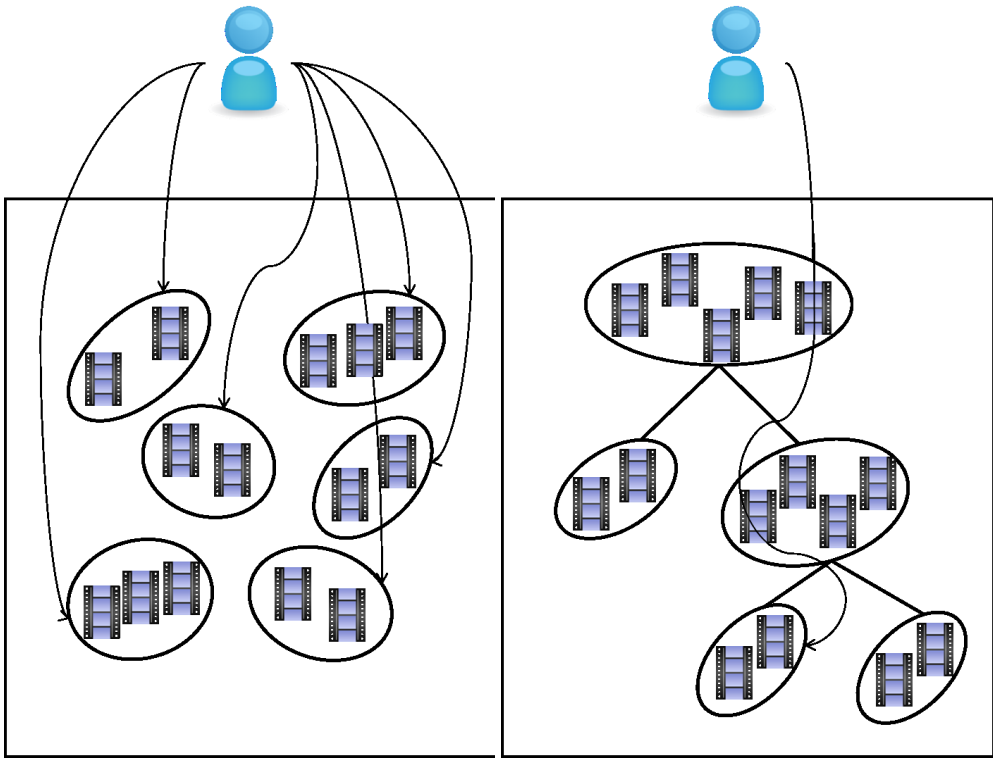
provided in high ranked results [3]. Therefore, it is necessary to develop an alternative method that enables accurate retrieval of desired Web videos even in such a case.

To meet this necessity, clustering-based retrieval methods, which use Web video groups, have been proposed [10–14]. In this thesis, Web video groups are defined as Web video sets with similar topics. Since these methods enable users to effectively browse many Web videos via the Web video groups, they can help users retrieve desired Web videos even if users cannot input a suitable query. It should be noted that topics contained in each Web video group have hierarchical relationships to each other and there exists high level topics that represent abstract contents and low level topics that represent specific contents. However, these clustering-based methods present Web video groups to users without focusing on the hierarchical relationships between topics. Therefore, as topics contained in the presented Web video group become more varied, it becomes more difficult to retrieve desired Web videos since users need to find Web video groups containing topics at the desired hierarchical levels from many Web video groups.

To solve this problem, hierarchical clustering-based retrieval methods that can provide results including topics at various hierarchical levels have been proposed [15–18]. As shown in Fig. 1.1, the hierarchical structure that includes many topics can navigate users to the desired Web videos. Although these methods can overcome the above difficulty, a single modality, *i.e.*, textual features or visual features, is utilized. Thus, these conventional methods have limitations in their performance due to the lack of useful information obtained from the other modalities. Since Web videos have heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos, the use of relevance between these heterogeneous features can overcome the limitation.

1.2 Proposition in this Thesis

To overcome the aforementioned limitation, this thesis proposes a method for extracting the hierarchical structure of Web video groups to realize retrieval of the desired Web videos. In this thesis, the hierarchical structure denotes the property of Web video groups being divided into sub-groups. The contributions of this thesis are as follows.



(a) (b)

Figure 1.1: Schemes to provide Web video groups. (a) Non-hierarchical scheme: If topics contained in the Web video groups and those contained in users' desired Web videos do not belong to the same hierarchical levels, retrieval of the desired Web videos becomes difficult. (b) Hierarchical scheme: The hierarchical structure, which contains many topics, can navigate users to the desired Web videos.

(Contribution-i)

The hierarchical structure of Web video groups is successfully extracted by utilizing the relevance between heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos.

(Contribution-ii)

Even if users cannot input suitable keywords as a query, accurate retrieval of the desired Web videos is realized by presentation of the obtained hierarchical structure to users.

1.3 Organization of this Thesis

The remainder of this thesis is organized as follows.

In Chapter 2, related work of Web video retrieval is presented and problems to be solved are clarified.

A method to extract the hierarchical structure of Web video groups by utilizing the relevance between heterogeneous features is presented in Chapter 3. Specifically, the proposed method first applies canonical correlation analysis (CCA) [19] to visual, audio and textual features of Web videos and calculates latent features of Web videos by using the results of CCA. By using the obtained latent features and features of link relationships between Web videos, a graph, whose edges represent similarities between latent features of Web videos, is constructed. From the graph, the proposed method extracts strongly connected components (SCCs) [20] and calculates modularity [21]. Furthermore, the sub-graphs and their inclusion relationships become clear by calculating edge betweenness [22] from each edge of the graph, and thus the hierarchical structure of Web video groups can be extracted.

In Chapter 4, a method that enables acceleration of the proposed method in Chapter 3 is presented. The proposed method applies CCA to a small number of vectors selected via a clustering scheme [23] and then constructs a graph for which nodes contain multiple Web videos. Since many Web videos can be handled by a small number of nodes, efficient extraction of the hierarchical structure is realized.

In Chapter 5, a method that enables improvement of the accuracy and efficiency of the method shown in Chapter 4 is presented. Specifically, significant acceleration of extraction of the hierarchical structure of Web video groups becomes feasible by applying a graph analysis algorithm that focuses only on local structure [24] to a graph that represents similarities between latent features of Web videos. The methods presented in Chapters 3 and 4 require screening of Web videos when they target many Web videos; therefore, there is a problem that the accuracy of extraction of the hierarchical structure may be degraded by the screening. Since the proposed method in Chapter 5 makes the screening unnecessary, the proposed method can solve this problem and enables improvement of the accuracy of extraction of the

hierarchical structure as well.

Finally, conclusions of this thesis and future directions of this study are discussed in Chapter 6.

Chapter 2

Related Work

2.1 Introduction

This chapter presents related work of this thesis. In this thesis, I propose methods that enable retrieval of desired Web videos even if users cannot input suitable keywords as a query by deriving clustering schemes via heterogeneous features. Therefore, in this chapter, I first present related work of Web video retrieval based on a query-response model (Section 2.2). Furthermore, I present related work of flat clustering-based Web video retrieval (Section 2.3.1) and hierarchical clustering-based Web video retrieval (Section 2.3.2). It should be noted that several conventional methods explained below do not focus only on Web videos since this chapter reviews schemes that can be applied to Web videos.

2.2 Related Work of Web Video Retrieval Based on a Query-Response Model

This section presents related work of Web video retrieval based on a query-response model. The progress of machine learning techniques has led to the development of semantic understanding for retrieving Web videos associated with the input query. In particular, recent progress of object recognition and event detection, which are the fundamental techniques for retrieval, has attracted wide attention. Below, several researches on them are shown.

Reference [6]

In this paper, a regularized multi-modality deep learning algorithm is proposed for video

event detection. By encoding the relationships between visual and audio modalities in the same video, multimodal features are utilized. Experimental results show that this conventional method significantly outperforms the other methods.

Reference [7]

This paper proposes a multimodal feature fusion method for robust event detection in Web videos. Specifically, this conventional method enables detection of events by combining many features including visual and audio features, and high-level features obtained from object detector responses, automatic speech recognition, and video text recognition.

Reference [8]

This paper presents a method to utilize the attributes at video level, *i.e.*, semantic label of external videos for event detection. Since many other works exploit the attributes at image level, complex event detection in videos is difficult due to the limited capability in characterizing the dynamic properties of videos. On the other hand, this paper proposes an approach to overcome this difficulty and shows the effectiveness by the experiments on a real-world large-scale dataset.

Reference [9]

This paper proposes a method that enables successful event detection based on the hypergraph theory to bridge semantic gap [25], *i.e.*, the difference between the low-level features and high-level interpretation of humans. Experiments on event detection and analysis from surveillance videos confirm the effectiveness of this conventional method.

These methods enable accurate retrieval of Web videos associated with the input query; however, accurate retrieval of desired Web videos becomes difficult if users cannot input a suitable query.

2.3 Related Work of Web Video Retrieval Based on Clustering Schemes

In this section, I present several researches on Web video retrieval based on clustering schemes, which aim to overcome the aforementioned difficulty. Since the browsing environment, which enables grasping the overview of many Web videos, can be provided on the basis of the clustering results, users can easily find the desired Web videos from many Web videos. It should be noted that such a browsing environment may be provided on the basis of classification schemes, which assign topics to Web videos, as well. However, there is a problem that classification schemes cannot assign topics that do not exist in a pre-defined topic set. Since new topics are frequently generated on the Web, clustering schemes, *i.e.*, unsupervised schemes, should be adopted for Web video retrieval. Therefore, in this section, I present Web video retrieval methods based on clustering schemes rather than classification schemes. In Fig. 2.1, the overview of related work explained below and the proposed method in this thesis.

2.3.1 Flat Clustering-based Web Video Retrieval

In this section, I explain related work of flat clustering-based Web video retrieval. To perform clustering of Web videos and realize its application to retrieval, methods using multimodal features and metadata such as browsing history have been proposed. Several methods are presented below.

Reference [10]

In this paper, in order to retrieve near duplicate Web videos, a method for performing clustering via visual features is proposed. Experiments for Web videos have verified that this conventional method can realize efficient and accurate retrieval in terms of retrieval time and recall and precision, respectively.

Reference [11]

In this paper, a method to perform clustering of Web videos by using textual features of title and description and users' browsing history is presented. This conventional method realizes large-scale clustering using 50000 Web videos as seeds on the basis of

parallel computation via MapReduce [26]. Furthermore, this paper presents a scheme for naming the obtained clusters by using Freebase topics [27] in order to assist users to find the desired Web videos.

Reference [12]

This paper assumes that users can only input short and ambiguous queries although they try to find Web videos with specific topics. Under this assumption, this paper proposes a method that enables retrieval of desired Web videos. Usually, existing search engines present Web videos with varied topics as the retrieval results if users can only input short and ambiguous queries.On the other hand, this conventional method realizes effective browsing of Web videos with varied topics on the basis of clustering via visual and textual features; therefore, it becomes feasible to retrieve the desired Web videos.In the experiments, this conventional method adopts normalized cuts [28] and affinity propagation [29] as clustering schemes and each performance is verified.

Reference [13]

This paper proposes a method that performs clustering based on multimodal features. In particular, this conventional method enables clustering of similar Web videos by utilizing visual, audio and textual features and features of link relationships between Web videos. Users can retrieve the desired Web videos accurately by selecting clusters including the desired Web videos from the extracted clusters.

Reference [14]

This is a method based on clustering via latent semantic indexing (LSI) that uses only textual features of Web videos. This conventional method utilizes information of YouTube Playlists, which is a type of users' viewing behavior. By the playlist-based clustering, users can grasp the outline of search result videos in a new light.

Although these conventional methods can perform successful clustering by using multimodal features and metadata, they present the obtained clusters to users without focusing on the hierarchical relationships between topics. Therefore, as topics contained in the presented clusters

become more varied, it becomes more difficult to retrieve desired Web videos since users need to find clusters containing topics at the desired hierarchical levels from many clusters.

2.3.2 Hierarchical Clustering-based Web Video Retrieval

In this section, conventional methods for performing hierarchical clustering-based Web video retrieval, which are useful for solving the above problem, are described. Recently, hierarchical clustering-based methods using unimodal features, which aim to effective browsing of many Web videos with varied topics, have been proposed as follows.

Reference [15]

This paper proposes a method to realize retrieval of the desired Web videos by presentation of the hierarchical topic structure obtained via textual features of Web videos and WordNet [30].It is assumed that users input complex queries concerning political and social events or issues, *e.g.*, 9/11 attack. Even in such a case, this conventional method allows users to discover desired topics by presentation of the hierarchical topic structure rather than ranked list results.

Reference [16]

This paper proposes a method to present clusters with the hierarchical structure whose upper layers contain similar color shots and lower layers contain shots with similar motion features. In the experiments, it is shown that this conventional method realizes effective retrieval of sports video shots.

Reference [18]

This paper proposes a method to retrieve similar video shots by using visual features on the basis of a hierarchical approach. Specifically, this paper presents “the similarity pyramid”, which enables effective browsing of shots with similar visual features. The similarity pyramid enables users to browse the video database at various levels of detail.

Although the above methods overcome the limitation in flat clustering-based methods by utilizing the hierarchical relationships between topics, they use only a single modality. Therefore,

these conventional methods have limitation in their performance due to the lack of useful information obtained from the other modalities.

2.4 Problems to Be Solved in this Thesis

In this section, problems to be solved are clarified. As explained in Section 2.2, the development of semantic understanding enables accurate retrieval of Web videos associated with the input query. However, accurate retrieval of desired Web videos becomes difficult if users cannot input a suitable query. Although flat clustering-based Web video retrieval methods shown in Section 2.3.1 aim to overcome this difficulty, it becomes more difficult to retrieve the desired Web videos as topics contained in the presented Web video groups become more varied. On the other hand, hierarchical clustering-based Web video retrieval methods described in Section 2.3.2 can solve this problem. However, these conventional methods have the limitation in their performance since they use only a single modality. Here, the use of relevance between heterogeneous features obtained from Web videos can overcome this limitation. Therefore, I should realize a new scheme for successfully extracting the hierarchical structure of Web video groups by utilizing the relevance between heterogeneous features to realize advanced Web video retrieval.

The remainder of this thesis is organized as follows. A fundamental scheme for extracting the hierarchical structure for Web video retrieval is proposed in Chapter 3. To improve the scalability of the scheme in Chapter 3, Chapter 4 presents a scheme for accelerating the extraction of the hierarchical structure. Furthermore, Chapter 5 presents a scheme that enables improvement of the accuracy by the method presented in Chapter 4. Through these propositions, it becomes feasible to accurately retrieve desired Web videos from many Web videos even if users cannot input suitable keywords as a query.

2.5 Conclusions

In this chapter, related work of Web video retrieval is reviewed and limitations in conventional methods are described. Furthermore, to realize successful extraction of the hierarchical struc-

ture of Web video groups for performing advanced Web video retrieval, problems to be solved in this thesis are clarified.

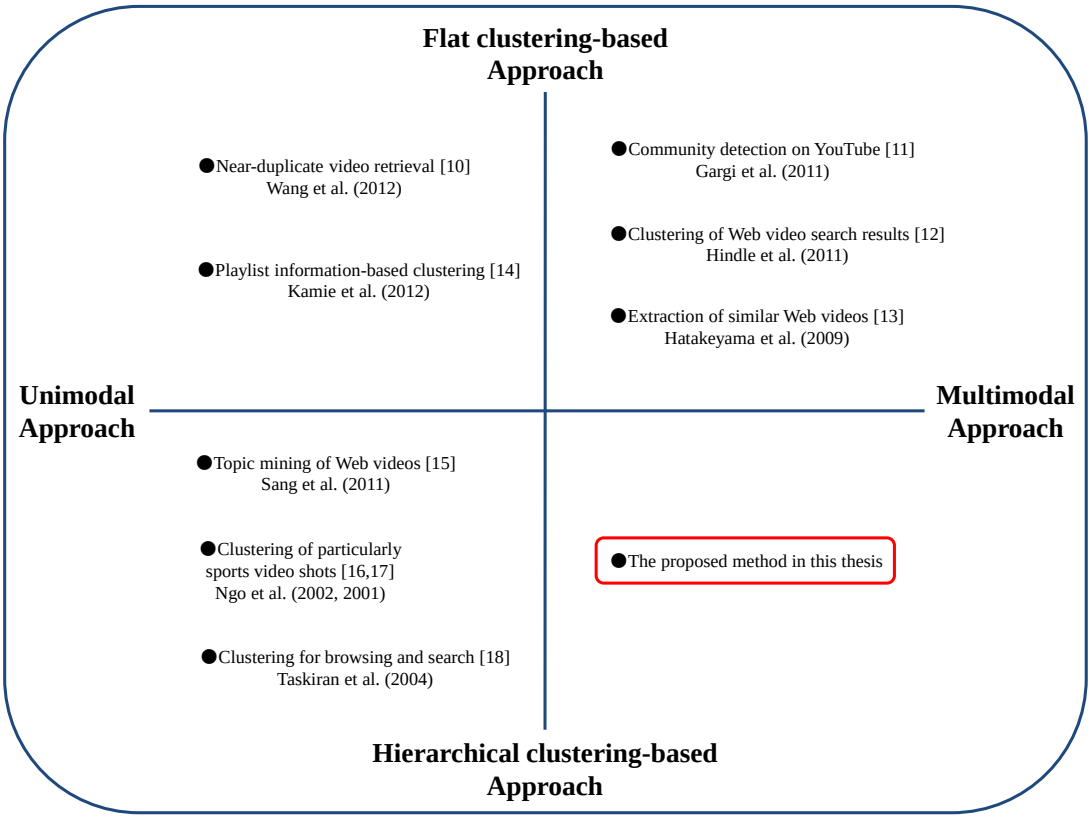


Figure 2.1: Overview of related work of this thesis and the proposed method in this thesis.

Chapter 3

Extraction of Hierarchical Structure of Web Video Groups for Web Video Retrieval

3.1 Introduction

To solve the problems explained in the previous chapter, this chapter proposes a method for extracting the hierarchical structure of Web video groups by utilizing relevance between heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos. The proposed method presents the hierarchical structure and users select Web video groups according to it. Thus, it becomes feasible to retrieve desired Web videos even if users cannot input suitable keywords as a query.

3.2 Outline of Extraction of Hierarchical Structure for Web Video Retrieval

In this section, an outline of the proposed method is presented. As shown in Fig. 3.1, the proposed method consists of three phases. An overview of them is shown below.

Phase I: Selection of Representative Web Videos Based on Web Video Features

This phase is useful for screening irrelevant Web videos from many Web videos. As a result, this phase enables both reduction of computational cost in Phase II and accurate retrieval of users' desired Web videos in Phase III.

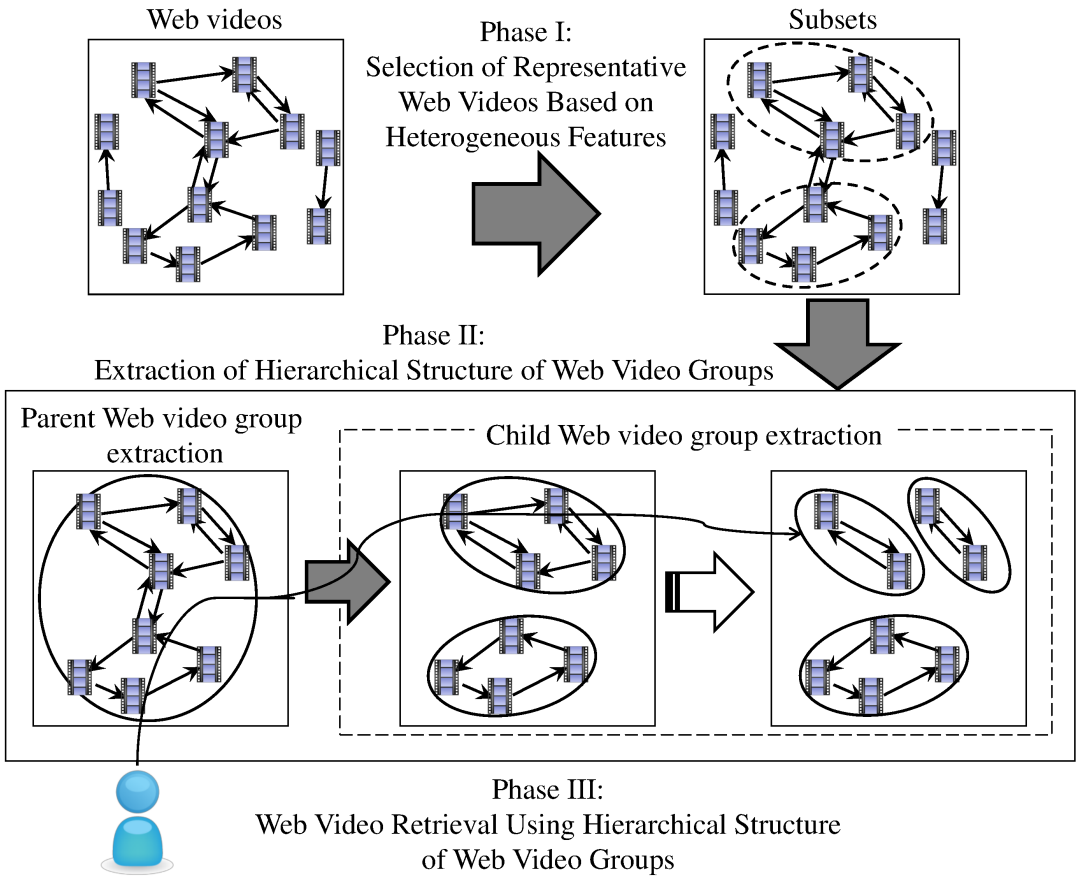


Figure 3.1: Outline of extraction of hierarchical structure of Web video groups for Web video retrieval.

Phase II: Extraction of Hierarchical Structure of Web Video Groups

This phase aims to provide the hierarchical structure of Web video groups related to a given query. By providing the hierarchical structure, it becomes feasible for users to easily find desired Web videos.

Phase III: Web Video Retrieval Using Hierarchical Structure of Web Video Groups

This phase enables to rank Web videos in each Web video group. Thus, users can retrieve Web videos in the descending order of the centrality in the selected Web video group.

The details of Phases I, II and III are presented in Sections 3.3, 3.4 and 3.5, respectively.

3.3 Selection of Representative Web Videos Based on Heterogeneous Features (Phase I)

In this section, a method for selecting representative Web videos based heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos, is explained.

3.3.1 Definition of Three Kinds of Features

This section describes the definition of three kinds of features, *i.e.*, visual, audio and textual features. Specifically, the three kinds of features are calculated from Web videos f_i ($i = 1, 2, \cdots, N$; N being the number of Web videos) according to a paper [13]. First, we apply the shot segmentation method [31] to each Web video f_i, and obtain several shots $s_i^{q_i}$ ($q_i = 1, 2, \cdots, M_i$; M_i being the number of shots within f_i). Then a visual feature vector $\boldsymbol{v}_i^{q_i}$, an audio feature vector $\boldsymbol{a}_i^{q_i}$ and a textual feature vector $\boldsymbol{t}_i^{q_i}$ are calculated from each shot $s_i^{q_i}$ as follows.

Visual Feature Vector (p dimensions)

The HSV color histogram with p bins is calculated every P_v frames of Web video f_i, and its vector is obtained. Then the vector median [32] is calculated for the frames in each shot $s_i^{q_i}$, and the calculated vector is defined as the visual feature vector $\boldsymbol{v}_i^{q_i}$ ($= [v_i^{q_i}(1), v_i^{q_i}(2), \cdots, v_i^{q_i}(p)]^{\mathrm{T}}$). Thus, we obtain M_i visual feature vectors $\boldsymbol{v}_i^{q_i}$ ($q_i = 1, 2, \cdots, M_i$) from each Web video f_i.

Audio Feature Vector (22 dimensions)

First, each shot $s_i^{q_i}$ in Web video f_i is divided into some clips.Then we classify each clip within the shot $s_i^{q_i}$ into four audio classes, *i.e.*, silence, speech, music and noise, based on a method [33]. Next, the audio class that is most included in the clips within each shot $s_i^{q_i}$ is selected. For all clips classified into the selected audio class in each shot $s_i^{q_i}$, we then calculate the averages and the standard deviations of the following 11 features:

- volume, zero-crossing rate, pitch, frequency centroid, frequency bandwidth, sub-

band energy ratio (0-630 Hz, 630-1720 Hz, 1720-4400 Hz and 4400-11025 Hz), non-silence ratio and zero-ratio.

Then, for each shot $s_i^{q_i}$, the audio feature vector $\boldsymbol{a}_i^{q_i}$ $(= [a_i^{q_i}(1), a_i^{q_i}(2), \cdots, a_i^{q_i}(22)]^{\mathrm{T}})$ is obtained by aligning the above obtained features. Thereby, M_i audio feature vectors $\boldsymbol{a}_i^{q_i}$ $(q_i = 1, 2, \cdots, M_i)$ are calculated from each Web video f_i.

Textual Feature Vector (r dimensions)

Let K be the total number of keywords that appear in text attached to Web videos f_i $(i = 1, 2, \cdots, N)$. We apply TF-IDF [34] to the K keywords, and obtain the weights of these K keywords. Then the vector $\boldsymbol{\eta}_i$ $(= [\eta_i(1), \eta_i(2), \cdots, \eta_i(K)]^{\mathrm{T}})$ is obtained by aligning the obtained weights. Furthermore, we apply principal component analysis (PCA) to $\boldsymbol{\eta}_i$ $(i = 1, 2, \cdots, N)$, and reduce their dimensions by the following procedures. First, from the obtained vectors $\boldsymbol{\eta}_i$ $(i = 1, 2, \cdots, N)$, we calculate a covariance matrix $\boldsymbol{D}$ as follows:

$$\boldsymbol{D} = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{\eta}_i - \overline{\boldsymbol{\eta}})(\boldsymbol{\eta}_i - \overline{\boldsymbol{\eta}})^{\mathrm{T}}, \tag{3.1}$$

where $\overline{\boldsymbol{\eta}}$ is the average vector of $\boldsymbol{\eta}_i$ $(i = 1, 2, \cdots, N)$. Moreover, by calculating the eigenvectors of the covariance matrix $\boldsymbol{D}$, we obtain the eigenvector matrix $\boldsymbol{U}$ $(\in \mathbb{R}^{K \times r})$ by the following equation:

$$\boldsymbol{D} \simeq \boldsymbol{U} \boldsymbol{\Lambda}_u^2 \boldsymbol{U}^{\mathrm{T}}, \tag{3.2}$$

where $\boldsymbol{\Lambda}_u^2$ is the eigenvalue matrix including r largest eigenvalues in the diagonal elements. Furthermore, a new r-dimensional vector $\boldsymbol{t}_i$ is calculated as follows:

$$\begin{aligned} \boldsymbol{t}_i &= \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{\eta}_i - \overline{\boldsymbol{\eta}}), \\ i &= 1, 2, \cdots, N. \end{aligned} \tag{3.3}$$

Thereby, for each shot $s_i^{q_i}$ in Web video f_i, we obtain M_i textual feature vectors $\boldsymbol{t}_i^{q_i}$ $(= \boldsymbol{t}_i)$ $(q_i = 1, 2, \cdots, M_i)$.

Thus, from each shot $s_i^{q_i}$ in the Web video f_i, the visual feature vector $\boldsymbol{v}_i^{q_i}$, the audio feature vector $\boldsymbol{a}_i^{q_i}$ and the textual feature vector $\boldsymbol{t}_i^{q_i}$ are calculated.

3.3.2 Selection of Representative Web Videos

In this section, a method for selecting representative Web videos by using the three kinds of features and features of link relationships between Web videos is explained. First, in order to obtain variates that can be compared between the different kinds of features, canonical correlation analysis (CCA) [19] is applied to the visual feature vector $\boldsymbol{v}_i^{q_i}$, the audio feature vector $\boldsymbol{a}_i^{q_i}$ and the textual feature vector $\boldsymbol{t}_i^{q_i}$. Then we obtain $\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}}, \boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}}$ and $\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}}$, which are the projection results of $\boldsymbol{v}_i^{q_i}, \boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$ into the latent space of l $(\in \{v, a, t\})$, where v, a and t correspond to visual, audio and textual features, respectively. Specifically,we first calculate three kinds of matrices $\boldsymbol{X}_v$, $\boldsymbol{X}_a$ and $\boldsymbol{X}_t$ by using $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$ as follows:

$$\boldsymbol{X}_v = \boldsymbol{H}\left[\boldsymbol{v}_1^1, \boldsymbol{v}_1^2, \cdots, \boldsymbol{v}_1^{M_1}, \cdots, \boldsymbol{v}_N^{M_N}\right]^{\mathrm{T}}, \tag{3.4}$$

$$\boldsymbol{X}_a = \boldsymbol{H}\left[\boldsymbol{a}_1^1, \boldsymbol{a}_1^2, \cdots, \boldsymbol{a}_1^{M_1}, \cdots, \boldsymbol{a}_N^{M_N}\right]^{\mathrm{T}}, \tag{3.5}$$

$$\boldsymbol{X}_t = \boldsymbol{H}\left[\boldsymbol{t}_1^1, \boldsymbol{t}_1^2, \cdots, \boldsymbol{t}_1^{M_1}, \cdots, \boldsymbol{t}_N^{M_N}\right]^{\mathrm{T}}. \tag{3.6}$$

In the above equations, $\boldsymbol{H}$ is the centering matrix defined by the following equations:

$$\boldsymbol{H} = \boldsymbol{I} - \frac{1}{D_H}\mathbf{1}\mathbf{1}^{\mathrm{T}}, \tag{3.7}$$

$$D_H = \sum_{i=1}^{N} M_i, \tag{3.8}$$

where $\boldsymbol{I}$ is the $D_H \times D_H$ identify matrix, and $\mathbf{1}$ is an D_H-dimensional vector in which each element equals one. Next, CCA calculates coefficient vectors $\boldsymbol{w}_k$ $(k \in \{v, a, t\})$, which maximize the correlation between the vectors $\boldsymbol{g}_k$ defined as follows:

$$\boldsymbol{g}_k = \boldsymbol{X}_k \boldsymbol{w}_k \quad (k \in \{v, a, t\}). \tag{3.9}$$

Specifically, by using a vector $\boldsymbol{y}$ whose elements are unknown, we estimate $\boldsymbol{w}_k$ which minimize

$$Q(\boldsymbol{y}, \boldsymbol{w}_k) = \sum_{k \in \{v,a,t\}} \|\boldsymbol{y} - \boldsymbol{X}_k \boldsymbol{w}_k\|^2 \quad (\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} = 1). \tag{3.10}$$

Here, the following inequality is obtained by the least squares method:

$$\begin{aligned} Q(\boldsymbol{y}, \boldsymbol{w}_k) &\geq \sum_{k \in \{v,a,t\}} \|\boldsymbol{y} - \boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{y}\|^2 & (3.11) \\ &= 3\|\boldsymbol{y}\|^2 - \boldsymbol{y}^{\mathrm{T}} \sum_{k \in \{v,a,t\}} (\boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}}) \boldsymbol{y}, & (3.12) \end{aligned}$$

with equality if and only if

$$\boldsymbol{w}_k = (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{y}. \tag{3.13}$$

Then we define

$$Q(\boldsymbol{y}) = 3\|\boldsymbol{y}\|^2 - \boldsymbol{y}^{\mathrm{T}} \sum_{k \in \{v,a,t\}} (\boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}}) \boldsymbol{y}. \tag{3.14}$$

Second, in order to minimize $Q(\boldsymbol{y})$, it is necessary to maximize the second term of Eq. (3.14). Thus, the vector $\boldsymbol{y}$ is calculated as the solution of the following eigenvalue problem:

$$\sum_{k\in\{v,a,t\}} (\boldsymbol{X}_k(\boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{X}_k)^{-1}\boldsymbol{X}_k^{\mathrm{T}})\boldsymbol{y} = \lambda\boldsymbol{y}. \tag{3.15}$$

From the above equation, according to N_e $(= \mathrm{rank}(\boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{X}_l),\ l \in \{v,a,t|l \neq k\})$ positive eigenvalues, N_e eigenvectors $\boldsymbol{y}_n$ $(n = 1, 2, \cdots, N_e)$ are obtained. Then, by using the obtained eigenvectors $\boldsymbol{y}_n$, the coefficient vectors $\boldsymbol{w}_k^n$ are calculated based on Eq. (3.13). Furthermore, the coefficient matrix $\boldsymbol{W}_k$ is defined as follows:

$$\boldsymbol{W}_k = [\boldsymbol{w}_k^1, \boldsymbol{w}_k^2, \cdots, \boldsymbol{w}_k^{N_e}]. \tag{3.16}$$

Then the following equation is obtained:

$$\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{X}_l\boldsymbol{W}_l = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{N_e} \end{bmatrix} \tag{3.17}$$

$$= \boldsymbol{\Lambda}_{k,l}, \tag{3.18}$$

where $\boldsymbol{\Lambda}_{k,l}$ is a correlation matrix whose diagonal elements are the canonical correlation coefficients λ_n $(n = 1, 2, \cdots, N_e)$ between $\boldsymbol{W}_k\boldsymbol{X}_k$ and $\boldsymbol{W}_l\boldsymbol{X}_l$. Next, we use $\boldsymbol{W}_k$, $\boldsymbol{\Lambda}_{k,l}$ and the three kinds of feature vectors $\boldsymbol{k}_i^{q_i}$ $(\in \{\boldsymbol{v}_i^{q_i} - \bar{\boldsymbol{v}}, \boldsymbol{a}_i^{q_i} - \bar{\boldsymbol{a}}, \boldsymbol{t}_i^{q_i} - \bar{\boldsymbol{t}}\})$, where $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{a}}$ and $\bar{\boldsymbol{t}}$ are respectively the average vectors of $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$. Then $\boldsymbol{\zeta}_{\boldsymbol{k}_i^{q_i}}^l$, which are the projection results of $\boldsymbol{k}_i^{q_i}$ into the latent space of $\boldsymbol{l}_i^{q_i}$ $(\in \{\boldsymbol{v}_i^{q_i} - \bar{\boldsymbol{v}}, \boldsymbol{a}_i^{q_i} - \bar{\boldsymbol{a}}, \boldsymbol{t}_i^{q_i} - \bar{\boldsymbol{t}}\})$, are calculated as follows:

$$\boldsymbol{\zeta}_{\boldsymbol{k}_i^{q_i}}^l = \begin{cases} \boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{k}_i^{q_i} & \text{if} \quad k = l \\ \boldsymbol{\Lambda}_{l,k}\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{k}_i^{q_i} & \text{otherwise.} \end{cases} \tag{3.19}$$

Hence, from the visual feature vector $\boldsymbol{v}_i^{q_i}$, the audio feature vector $\boldsymbol{a}_i^{q_i}$ and the textual feature vector $\boldsymbol{t}_i^{q_i}$, CCA calculates the vectors $\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}}$, $\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}}$ and $\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}}$ which can be compared between the different kinds of features.

Next, by using $\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}}$, $\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}}$ and $\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}}$, we calculate similarities $S(i,j)$ between Web videos f_i and f_j by the following equations:

$$S(i,j) = \max_{q_i,q_j} \left| \frac{(\boldsymbol{\xi}_i^{q_i})^{\mathrm{T}} \boldsymbol{\xi}_j^{q_j}}{\|\boldsymbol{\xi}_i^{q_i}\| \|\boldsymbol{\xi}_j^{q_j}\|} \right|, \tag{3.20}$$

$$\boldsymbol{\xi}_i^{q_i} = \left[(\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}})^{\mathrm{T}}, (\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}})^{\mathrm{T}}, (\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}})^{\mathrm{T}} \right]^{\mathrm{T}}. \tag{3.21}$$

In this way, we calculate the similarities between Web videos by integrating the three kinds of features on the basis of CCA. Furthermore, we weight the adjacency matrix that represents link relationships between Web videos and calculate the weighted adjacency matrix $\boldsymbol{L}_w$ whose (i,j)-th element $L_w(i,j)$ is as follows:

$$L_w(i,j) = \begin{cases} S(i,j) & \text{if } f_i \text{ links to } f_j \\ 0 & \text{otherwise} \end{cases}, \quad i = 1,2,\cdots,N, \quad j = 1,2,\cdots,N. \tag{3.22}$$

In the above equation, we build link relationships between Web videos based on metadata "related videos". In particular, we consider that a Web video f_i links to a Web video f_j if "related videos" of f_i include f_j. Since the metadata "related videos" are useful for obtaining Web videos related to each other [35], we introduce them into the proposed method. Moreover, we calculate N_g eigenvectors of $\boldsymbol{L}_w^{\mathrm{T}} \boldsymbol{L}_w$, where N_g is the number of the calculated eigenvectors. Each element of the eigenvectors of $\boldsymbol{L}_w^{\mathrm{T}} \boldsymbol{L}_w$ represents the attribution degree of each Web video to sets of Web videos with similar contents [36]. Next, we select N_m Web videos in descending order of the element values of each eigenvector, and represent these Web videos as C_q $(q = 1,2,\cdots,N_g)$. In this chapter, Web videos contained in C_q are called a "subset". By extracting the subset, we can obtain representative Web videos in the graph of link

relationships between Web videos since Web videos that densely link to each other can be extracted [36]. Thus, we can filter out irrelevant Web videos from many Web videos.

3.4 Extraction of Hierarchical Structure of Web Video Groups (Phase II)

In this section, a method for extracting the hierarchical structure of Web video groups is explained. In the proposed method, we summarize Web videos related to each other as Web video groups based on SCCs [20] and extract their hierarchical structure by using edge betweenness [22] and modularity [21]. Since SCCs are useful to find Web videos related to each other in the graph of link relationships between Web videos, we introduce this measure to obtain "parent Web video groups". Moreover, a study [22] reports that the combination use of edge betweenness and modularity enables extraction of the significant hierarchical structure. Therefore, these two measures are adopted to extract the hierarchical structure as in a work [22]. Specifically, we extract the hierarchical structure by repeatedly dividing parent Web video groups into their "child Web video groups"[1] on the basis of edge betweenness and modularity. In this thesis, we collectively call parent Web video groups and child Web video groups "Web video groups". Next, the details of this scheme are explained.

3.4.1 Construction of Parent Web Video Groups

First, from Web videos that belong to the subset C_q $(q = 1, 2, \cdots, N_g)$, a weighted and directed graph $G = (V, E)$, whose nodes and edges are respectively Web videos and links between these videos, is constructed. When a Web video $f_i \in V$ links to $f_j \in V$, the edge weight e_{ij} from f_i to f_j is defined as follows:

$$e_{ij} = L_w(i, j). \tag{3.23}$$

[1]When a child Web video group is divided into its child Web video groups, this child Web video group is considered to be the parent Web video group of its child Web video groups.

Thus, the edge weight represents the similarity between latent features of Web videos. When $f_i \in V$ does not link to $f_j \in V$, we do not define the edge e_{ij}. In the proposed method, parent Web video groups are obtained by extracting SCCs. SCCs are sub-graphs that have a directed path between any two nodes in a directed graph. A method for extracting SCCs from a graph is called SCC decomposition. In a study [20], related Web pages are grouped on the basis of SCC decomposition. In the proposed method, by applying SCC decomposition to G, all SCCs are extracted. Then these obtained components are defined as parent Web video groups. By extracting parent Web video groups, we can obtain Web video sets with similar topics.

3.4.2 Extraction of Child Web Video Groups

Next, we extract child Web video groups by repeatedly dividing the parent Web video groups on the basis of edge betweenness. Edge betweenness is used for detecting communities in a graph [22]. Edge betweenness is defined by the number of the shortest paths that go through a target edge, and it is useful for detecting influential edges in a graph. Let $c_B(e)$ be edge betweenness of an edge $e \in E$, and $c_B(e)$ is defined as follows:

$$\begin{aligned} c_B(e) &= \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}, \\ e &\in E, \end{aligned} \tag{3.24}$$

where $\sigma(s,t|e)$ is the number of shortest paths from a node s to a node t that pass an edge e, and $\sigma(s,t)$ is the total number of shortest paths from a node s to a node t. In the proposed method, by iteratively removing edges with the largest edge betweenness in G and re-applying SCC decomposition to G, we re-extract SCCs. In this chapter, we denote the number of these iterations by N_{cut}. Here, child Web video groups are extracted when a single SCC is divided into several SCCs. By repeatedly dividing the parent Web video groups into their child Web video groups, the hierarchical structure of the Web video groups is obtained. It should be noted that each child Web video group becomes equivalent to each Web video when all edges in G are removed. Since it is not effective for Web video retrieval to provide all hierarchies to

users, we have to decide when the proposed method stops dividing parent Web video groups into their child Web video groups.

3.4.3 Introduction of Modularity

To solve the above problem, a quality function called modularity, which evaluates the results of division of Web video groups in a graph, is introduced into the proposed method. Note that the node set V of $G = (V, E)$ does not contain all Web videos f_i $(i = 1, 2, \cdots, N)$ since G is constructed by Web videos that belong to the extracted subset C_q $(q = 1, 2, \cdots, N_g)$. Therefore, in order to calculate modularity of G, we define the matrix $\boldsymbol{M} = (m_{ij})$ as follows:

$$m_{ij} = \begin{cases} L_w(i,j) & \text{if } f_i \in V \text{ and } f_j \in V \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, 2, \cdots, N, \quad j = 1, 2, \cdots, N. \tag{3.25}$$

Let $Q_{N_{cut}}$ be modularity of G where the number of cut edges is N_{cut}, and $Q_{N_{cut}}$ is defined as follows [21]:

$$Q_{N_{cut}} = \frac{1}{2m} \sum_{i=1}^{N} \sum_{j=1}^{N} (m_{ij} - \frac{m_i^{out} m_j^{in}}{2m})\, \delta(i,j), \tag{3.26}$$

where

$$2m = \sum_{i=1}^{N} \sum_{j=1}^{N} m_{ij}, \tag{3.27}$$

$$m_i^{out} = \sum_{j=1}^{N} m_{ij}, \tag{3.28}$$

$$m_j^{in} = \sum_{i=1}^{N} m_{ij}, \tag{3.29}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ belong to the same Web video group} \\ 0 & \text{otherwise} \end{cases}. \tag{3.30}$$

Algorithm 1 : Extraction of Hierarchical structure of Web video groups.

Input: A weighted and directed graph $G = (V, E)$.
Output: Parent and child Web video groups.

$N_{cut} \leftarrow 0$
Apply SCC decomposition to G, and extract parent Web video groups.
Calculate modularity $Q_{N_{cut}}$.
while an edge of G remains **do**
  Calculate edge betweenness of remaining edges in G.
  if only an edge has the largest edge betweenness **then**
    Remove the edge with the largest edge betweenness.
  else
    Select one edge from the edges with the largest edge betweenness randomly, and remove the edge.
  end if
  $N_{cut} \leftarrow N_{cut} + 1$
  Apply SCC decomposition to G.
  if the number of SCCs is changed **then**
    Extract child Web video groups.
  end if
  Calculate modularity $Q_{N_{cut}}$.
end while
$N_{cut}^{opt} \leftarrow \arg\max_{N_{cut}} Q_{N_{cut}}$
Return parent and child Web video groups, where N_{cut} is from 0 to N_{cut}^{opt}.

When significant community structure is revealed, $Q_{N_{cut}}$ becomes close to 1, *i.e.*, the maximum value of $Q_{N_{cut}}$. On the other hand, a graph is divided into Web video groups randomly, and $Q_{N_{cut}}$ becomes close to 0. In this chapter, the N_{cut} value when $Q_{N_{cut}}$ is maximum is denoted by N_{cut}^{opt}. The proposed method uses the Web video groups, where N_{cut} is from 0 to N_{cut}^{opt} for Web video retrieval.

In this way, we hierarchically extract Web video groups based on SCCs, edge betweenness and modularity. Consequently, the entire contents of many Web videos can be easily grasped by obtaining the hierarchical structure of Web video groups. Algorithm 1 shows the specific procedures for extracting the hierarchical structure.

3.5 Web Video Retrieval Using Hierarchical Structure of Web Video Groups (Phase III)

In this section, a Web video retrieval method using the hierarchical structure of Web video groups is presented. We obtain Web video groups where N_{cut} is from 0 to N_{cut}^{opt}. In this chapter, these Web video groups are represented as $Com_{N_{cut}}^{q}$ ($N_{cut} = 0, 1, \cdots, N_{cut}^{opt},\ q = 1, 2, \cdots, T_{N_{cut}}; T_{N_{cut}}$ being the number of Web video groups in which the number of cut edges is N_{cut}). First, each Web video f_i ($i \in \{1, 2, \cdots, N\}$) that belongs to the Web video group $Com_{N_{cut}}^{q}$ is ranked in descending order of the following criterion $R_{N_{cut}}^{q}(i)$:

$$R_{N_{cut}}^{q}(i) = \sum_{j=1}^{N} m_{ji}\, \delta(i,j), \tag{3.31}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ belong to} \\ & \text{the same Web video group.} \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

Hence, Web videos belonging to each Web video group are ranked on the basis of weighted links from other Web videos in the same Web video group.

Next, we present the Web video groups $Com_{N_{cut}}^{q}$, where N_{cut} is from 0 to N_{cut}^{opt}. Here, the smaller N_{cut} is, the more various topics are contained in the Web video groups. On the other hand, the larger N_{cut} is, the more closely related Web videos are contained in the Web video groups. Then users select Web video groups associated with the desired Web videos according to the hierarchical structure. Furthermore, users retrieve the desired Web videos from the selected Web video group based on the criterion $R_{N_{cut}}^{q}(i)$. In this way, even if users cannot write suitable keywords as a query, the hierarchical structure can navigate users to the desired Web videos.

Table 3.1: Details of datasets: N_g and N_m are the parameters for constructing the subsets.

	Query keyword	Num. of Web videos in the dataset	N_g	N_m	Num. of Web videos in the subset
Dataset 1	sightseeing	3001	20	80	1038
Dataset 2	game	3003	20	80	1079

3.6 Experimental Results

In this section, experimental results for Web videos collected by using YouTube are shown to verify the effectiveness of the proposed method.

3.6.1 Datasets

In the proposed method, metadata "related videos" on each Web video plays an important role to extract the hierarchical structure for Web video retrieval. However, standard video retrieval datasets do not have the metadata. Therefore, we constructed datasets by our own for performing a fair comparison as follows. First, by using YouTube, a keyword was given as a query, and the top 50 Web videos were obtained. Then we repeatedly obtained 10 Web videos contained in links, *i.e.*, "related videos"[2] of the selected Web videos, and each dataset was constructed. Here, we collected only Web videos with lengths of less than 1800 seconds and ones to which Freebase topics [27] were attached. It should be noted that Freebase is a knowledge database maintained by a community supported by Google, and YouTube videos are associated with their relevant topics.Meanwhile, in order to compute visual feature vectors, p was set to 48 $(H = 12, S = 2, V = 2)$ and P_v was defined as the vectors were computed once a second. We used keywords that appeared in two and more Web videos for computing textual feature vectors. Table 3.1 shows the other detailed conditions of each dataset.

3.6.2 Evaluations

From each dataset, subsets were extracted according to Section 3.3 (Phase I). Next, from each subset, we extracted the parent Web video groups and the child Web video groups according to Section 3.4 (Phase II). Table 3.2 shows N_{cut}^{opt} and the maximum value of $Q_{N_{cut}}$ of each

[2]In the experiment, we used YouTube Data API (v3) to obtain the links, "related videos".

Table 3.2: N_{cut}^{opt} and maximum value of $Q_{N_{cut}}$ for each subset.

	N_{cut}^{opt}	Maximum value of $Q_{N_{cut}}$
Subset 1	88	0.857
Subset 2	148	0.812

subset are shown. In Figs. 3.2 and 3.3, each hierarchical structure for subsets 1 and 2, which were obtained by the proposed method, are shown. Note that we named each Web video group by checking contents of them manually since the proposition in this thesis is to extract the hierarchical structure and does not include the naming scheme. These figures show that the more closely related Web videos are obtained with an increase in the number of cut edges, N_{cut}. Thus, it can be seen that the hierarchical structure of Web video groups can be estimated by the proposed method.

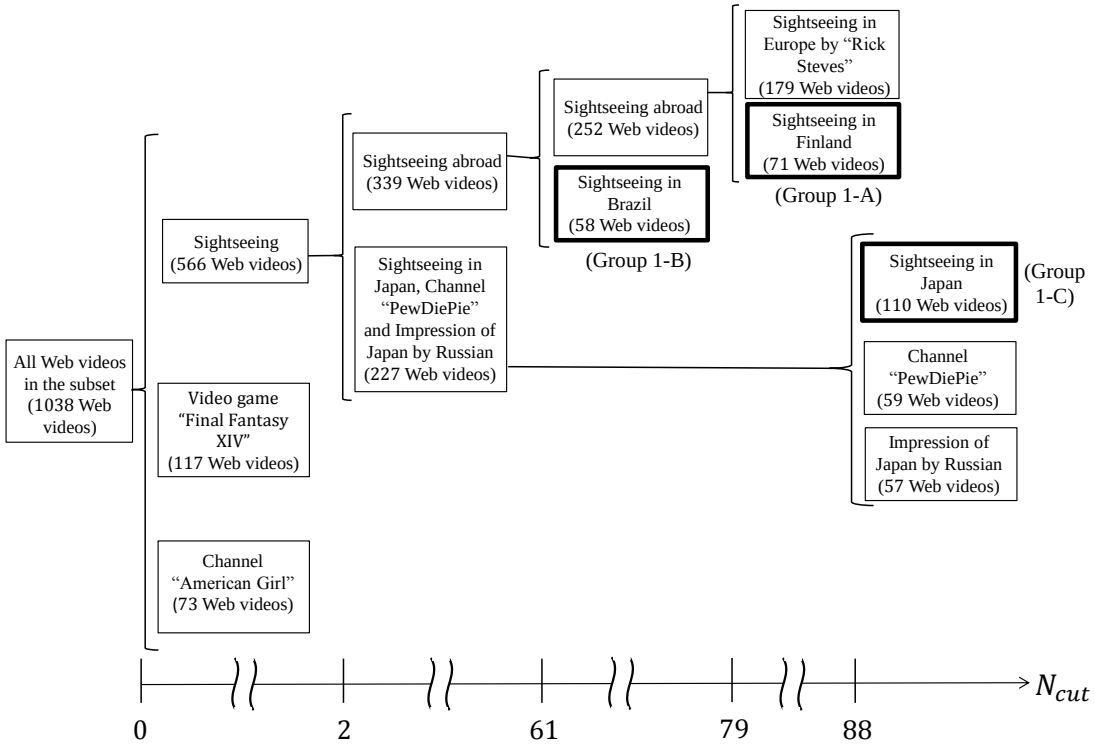
Next, we quantitatively verify the effectiveness of the proposed method shown in Section 3.5 (Phase III). We used recall, precision and average precision (AP@k) defined as follows:

$$\text{Recall} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of relevant Web videos}}, \tag{3.33}$$

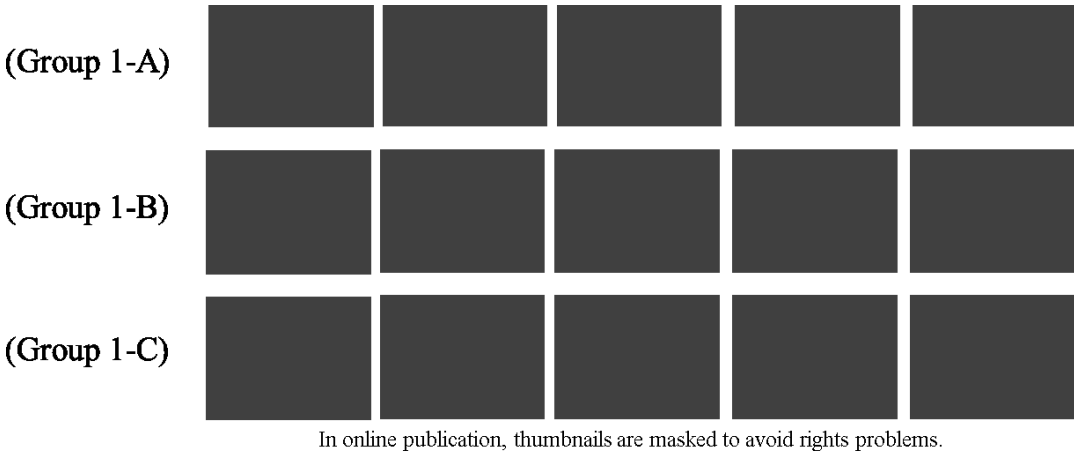
$$\text{Precision} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of retrieved Web videos}}, \tag{3.34}$$

$$\text{AP@}k = \frac{1}{R_k}\sum_{i=1}^{k} x_i \, prec_i, \tag{3.35}$$

where k is the number of Web videos provided as the retrieval results, R_k is the number of "relevant Web videos" within k Web videos of the retrieval results, x_i is 1 if the i-th retrieved Web videos are "relevant Web videos" and 0 otherwise, and $prec_i$ is the precision when i Web videos are retrieved.



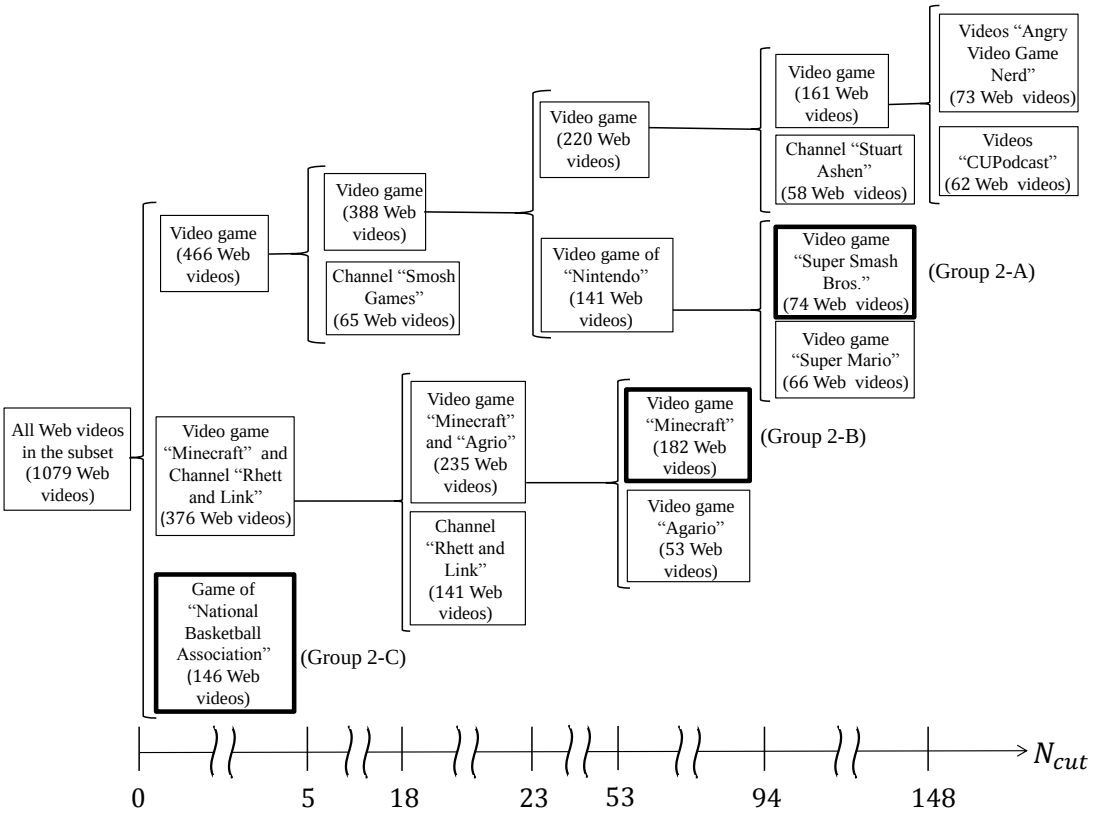
(a) Hierarchical structure of Web video groups.



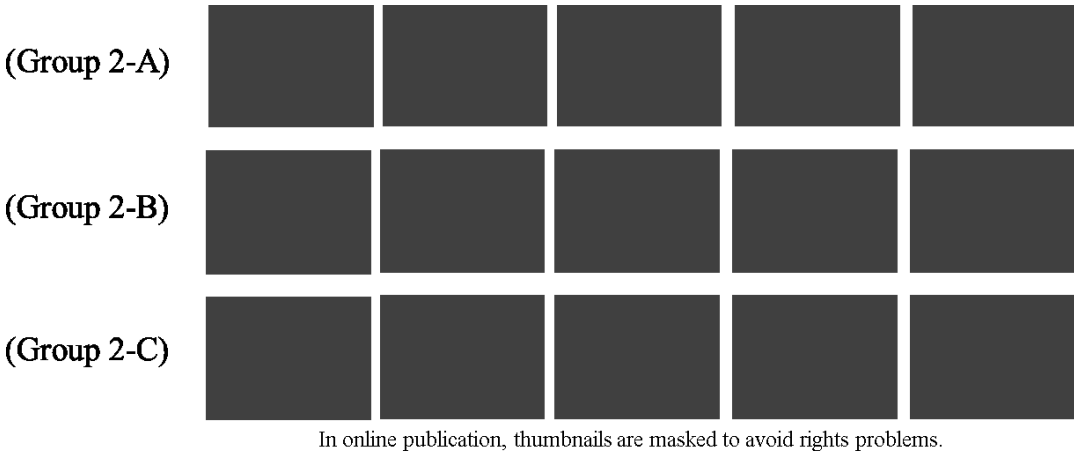
In online publication, thumbnails are masked to avoid rights problems.

(b) Thumbnails of the retrieved Web videos when (Group 1-A), (Group 1-B) and (Group 1-C) were respectively selected. The top five ranked Web videos were located in the order of left to right.

Figure 3.2: Hierarchical structure of Web video groups for subset 1, which were obtained by the proposed method.



(a) Hierarchical structure of Web video groups.



(b) Thumbnails of the retrieved Web videos when (Group 2-A), (Group 2-B) and (Group 2-C) were respectively selected. The top five ranked Web videos were located in the order of left to right.

Figure 3.3: Hierarchical structure of Web video groups for subset 2, which were obtained by the proposed method.

We compare the retrieval results via the proposed method with the following reference methods.

Reference method (i)

This is a method that extracts the hierarchical structure of Web video groups by using only link relationships between Web videos without the use of Web video features.

Reference method (ii)

This is a conventional method [13] that extracts Web video groups by using link relationships between Web videos and the Web video features. This method does not extract the hierarchical structure.

Reference method (iii)

This is a method based on [12] with a Web video group extraction scheme by affinity propagation [29]. This method uses only Web video similarities without the use of link relationships between Web videos, and does not extract the hierarchical structure. In the experiment, we did not use the original similarities presented by [12] but utilized our proposed similarities defined in Eq. (3.20). For the evaluation, we ranked Web videos within Web video groups obtained by this method in descending order of the sum of the above similarities.

Reference method (iv)

This is a method based on clustering via latent semantic indexing (LSI) that uses only textual features of Web videos [14], which does not present the hierarchical structure to users. Note that an original method in [14] utilizes information of YouTube Playlists. In this experiment, however, we used a video-keyword vectors whose elements are 1 if each keyword appears in title and description of the Web video and 0 otherwise. Moreover, the number of dimensions of the video-keyword vectors after LSI were set to 200 as in a paper [14]. Each Web video is ranked on the basis of the attribution degree to the belonging cluster.

For all methods, we selected a Web video group that included relevant Web videos the most,

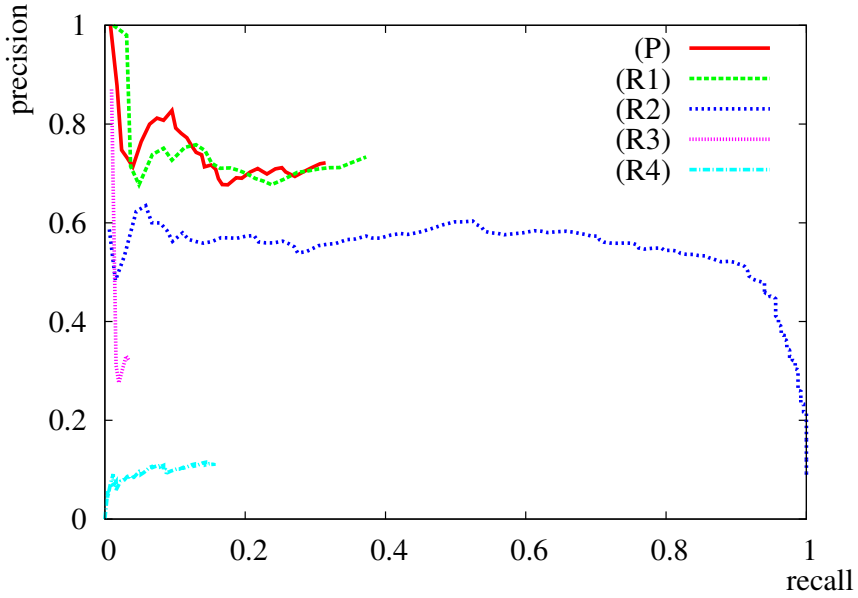
Table 3.3: Relevant Web videos used for drawing Fig. 3.4.

	Relevant Web videos
Dataset 1	Web videos with Freebase topics about "Sightseeing in Japan"
Dataset 2	Web videos with Freebase topics about "Game of National Basketball Association"

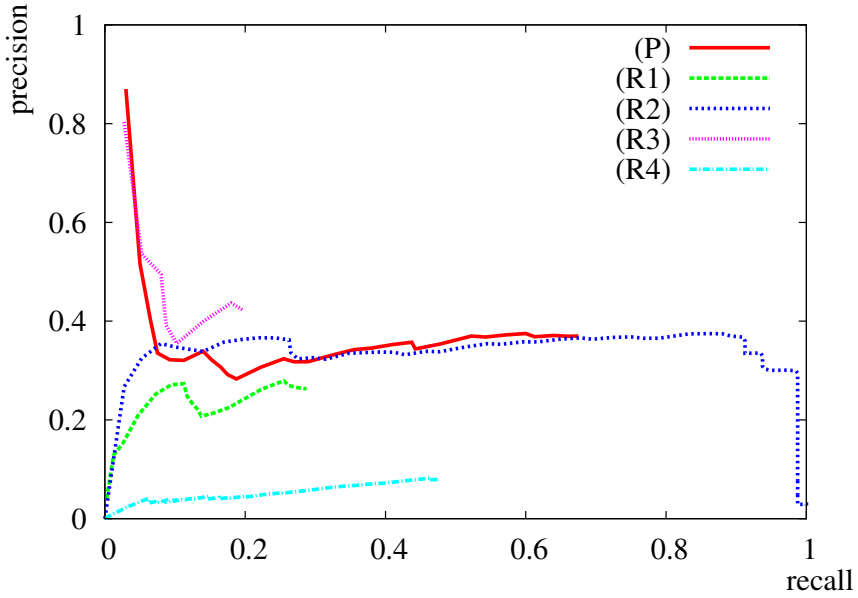
and Web videos were retrieved according to the rankings of Web videos. Then recall and precision were calculated and Fig. 3.4 shows precision-recall curves for the datasets. Table 3.3 shows relevant Web videos used for drawing this figure. Here, Freebase topics related to the Web video group were selected by human assessors, and then we set the ground truth for the evaluation. In Fig. 3.4 (a), it seems that precision of the proposed method was nearly same as reference method (i). Also, in Fig. 3.4 (b), it seems that precision of the proposed method was lower than reference method (iii). However, reference methods (i) and (iii) cannot show the high performance for all datasets although the proposed method can stably give the successful results for two datasets. Therefore, we can confirm the superiority of the proposed method.

For further evaluations, Table 3.4 shows the averages of AP@k. Here, we selected the most frequent Freebase topics within each Web video group as ground truths and calculated the averages, which were weighted by the numbers of Web videos in Web video groups. From these results, the effectiveness of using the heterogeneous features can be seen by comparing the proposed method with reference methods (i), (iii) and (iv). By comparing the proposed method with reference methods (iii) and (iv), it can be confirmed that link relationships between Web videos are useful for obtaining similar Web videos accurately whereas the only use of low-level video features is insufficient. When comparing the proposed method with reference methods (ii), (iii) and (iv), we can see that it becomes feasible to accurately retrieve the desired Web videos by using the hierarchical structure of Web video groups. In particular, accurate retrieval can be realized since Web video groups are refined into more similar topics according to the hierarchical structure.

As a consequence, this experiment has shown the effectiveness of the proposed method for Web video retrieval.



(a) Result for dataset 1.



(b) Result for dataset 2.

Figure 3.4: Precision-recall curves: (P) and (R1) respectively correspond to the proposed method and reference method (i), where N_{cut} were N_{cut}^{opt}. (R2), (R3) and (R4) correspond to reference methods (ii), (iii) and (iv), respectively. Note that reference method (ii) has higher recall than other methods since only this method is based on soft clustering and all Web videos can belong to every Web video group.

Table 3.4: AP@k weighted by the numbers of Web videos in each Web video group. For the proposed method and reference method (i), the results where N_{cut} were N_{cut}^{opt} are shown. For reference method (ii) based on soft clustering, ideal values are shown since we selected Freebase topics that provided the highest AP@k of all as ground truths. Also, k was defined as the number of Web videos in each Web video group.

	Proposed method	Reference method (i)	Reference method (ii)	Reference method (iii)	Reference method (iv)
Dataset 1	**0.765**	0.756	0.632	0.467	0.320
Dataset 2	**0.851**	0.834	0.563	0.648	0.373
Average	**0.808**	0.795	0.598	0.558	0.347

3.6.3 Conclusions

In this chapter, a method for extracting the hierarchical structure of Web video groups by utilizing relevance between heterogeneous features has been proposed. By utilizing heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos, limitation in conventional methods that used only a single modality can be overcome. Experimental results for actual Web videos have confirmed the effectiveness of the proposed method. In the next chapter, in order to apply the proposed method to many Web videos, a method for efficiently extracting the hierarchical structure is presented.

Chapter 4

Efficient Extraction of Hierarchical Structure of Web Video Groups for Web Video Retrieval

4.1 Introduction

This chapter proposes a method for efficiently extracting hierarchical structure of Web video groups. The method explained in Chapter 3 has the following drawbacks.

(i) Computational cost of similarity calculation between latent features of Web videos based on CCA is high.

(ii) Computational cost of graph analysis based on SCC, edge betweenness and modularity is high.

To solve the problem (i), efficient CCA, named sub-sampled CCA, is derived by introducing a clustering scheme to reduce computational cost of CCA. To solve the problem (ii), a new graph whose node contains multiple Web videos, named a group graph, is constructed to analyze many Web videos by a small number of nodes. Experimental results for actual Web videos show the effectiveness of the proposed method.

4.2 Latent Feature Extraction Based on Sub-sampled CCA

This section explains a method for calculating latent features of Web videos on the basis of sub-sampled CCA. In Section 4.2.1, we define visual, audio and textual features of Web videos

used in the proposed method. Section 4.2.2 shows a method for calculating latent features by applying sub-sampled CCA to the three kinds of features.

4.2.1 Definition of Three Kinds of Features of Web Videos

In this section, we show a method for extracting visual and audio features of Web videos based on a method [13] and computing textual features by using random projection [37]. Specifically, a shot segmentation method [31] is applied to each Web video f_i ($i = 1, 2, \cdots, N$; N being the number of Web videos), and we obtain several shots $s_i^{q_i}$ ($q_i = 1, 2, \cdots, M_i$; M_i being the number of shots within f_i) from each Web video. Next, we calculate a visual feature vector $\boldsymbol{v}_i^{q_i}$, an audio feature vector $\boldsymbol{a}_i^{q_i}$ and a textual feature vector $\boldsymbol{t}_i^{q_i}$ from each shot $s_i^{q_i}$ as follows.

Visual Feature Vector (p dimensions)

As in Section 3.3.1, for each shot $s_i^{q_i}$, visual feature vector $\boldsymbol{v}_i^{q_i} (= [v_i^{q_i}(1), v_i^{q_i}(2), \cdots, v_i^{q_i}(p)]^{\mathrm{T}})$ is calculated on the basis of HSV color histograms. Thus, from each Web video f_i, we calculate M_i visual feature vectors $\boldsymbol{v}_i^{q_i} (q_i = 1, 2, \cdots, M_i)$.

Audio Feature Vector (22 dimensions)

According to Section 3.3.1, for each shot $s_i^{q_i}$, audio feature vector $\boldsymbol{a}_i^{q_i}$ $(= [a_i^{q_i}(1), a_i^{q_i}(2), \cdots, a_i^{q_i}(22)]^{\mathrm{T}})$ is obtained based on the reference [33]. Thereby, M_i audio feature vectors $\boldsymbol{a}_i^{q_i}$ $(q_i = 1, 2, \cdots, M_i)$ are calculated from each Web video f_i.

Textual Feature Vector (U dimensions)

It should be noted that textual feature vectors, which can be efficiently calculated, are newly introduced although Section 3.3.1 adopts PCA-based feature vectors. First, the total number of keywords that appear in title and description of N Web videos f_i $(i = 1, 2, \cdots, N)$ is denoted by K. Then we obtain weights corresponding to these K keywords by applying TF-IDF [34] to the keywords. Next, we obtain the vector $\boldsymbol{\eta}_i$ $(= [\eta_i(1), \eta_i(2), \cdots, \eta_i(K)]^{\mathrm{T}})$ by aligning the obtained weights for each Web video. Furthermore, by applying random projection [37] to $\boldsymbol{\eta}_i$ $(i = 1, 2, \cdots, N)$, their dimensions are reduced to U $(< K)$ by the following procedures. First, a $K \times U$ matrix

$\boldsymbol{R} = (r_{ij})$ is calculated as follows:

$$r_{ij} = \begin{cases} \sqrt{3} & \text{with probability of } \frac{1}{6} \\ 0 & \text{with probability of } \frac{2}{3} \\ -\sqrt{3} & \text{with probability of } \frac{1}{6} \end{cases}. \tag{4.1}$$

Then a new U-dimensional vector $\boldsymbol{t}_i$ is calculated as follows:

$$\boldsymbol{t}_i = \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\eta}_i, \tag{4.2}$$
$$i = 1, 2, \cdots, N.$$

Thus, M_i textual feature vectors $\boldsymbol{t}_i^{q_i}$ $(= \boldsymbol{t}_i)$ $(q_i = 1, 2, \cdots, M_i)$ are obtained for each shot $s_i^{q_i}$ in Web video f_i.

In this way, the visual feature vector $\boldsymbol{v}_i^{q_i}$, audio feature vector $\boldsymbol{a}_i^{q_i}$ and textual feature vector $\boldsymbol{t}_i^{q_i}$ are calculated from each shot $s_i^{q_i}$ in the Web video f_i.

4.2.2 Extraction of Latent Features of Web Videos

This section presents a method for calculating latent features of Web videos by applying sub-sampled CCA to $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$. Specifically, we first perform k-means clustering [23] to $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$ $(i = 1, 2, \cdots, N,\ q_i = 1, 2, \cdots, M_i)$ for each modality. Then we obtain the cluster centers $\boldsymbol{v}_j^{cent}$, $\boldsymbol{a}_j^{cent}$ and $\boldsymbol{t}_j^{cent}$ $(j = 1, 2, \cdots, N_{clus}; N_{clus}$ being the number of cluster centers) corresponding to visual, audio and textual feature vectors, respectively. Furthermore, from $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$ $(i = 1, 2, \cdots, N,\ q_i = 1, 2, \cdots, M_i)$, we select vectors with the shortest Euclidean distance to the obtained cluster centers $\boldsymbol{v}_j^{cent}$, $\boldsymbol{a}_j^{cent}$ and $\boldsymbol{t}_j^{cent}$ $(j = 1, 2, \cdots, N_{clus})$ for each modality. In this chapter, the index of the visual feature vectors, which are selected from $\boldsymbol{v}_i^{q_i}$ $(i = 1, 2, \cdots, N, q_i = 1, 2, \cdots, M_i)$ that is most similar to the cluster centers $\boldsymbol{v}_j^{cent}$ $(j = 1, 2, \cdots, N_{clus})$, is denoted by I_v, and $|I_v|$ $(= N_{clus})$ denotes the number of elements. In the same manner, we respectively define I_a and I_t for the audio and textual feature vectors, where $|I_a| = |I_t| = N_{clus}$. Here, the selected vectors are denoted by $\boldsymbol{x}_{i'}^{v}$, $\boldsymbol{x}_{i'}^{a}$ and $\boldsymbol{x}_{i'}^{t}$

Algorithm 2 : Selection of vectors that become calculation targets of CCA.

Input: Visual, audio and textual feature vectors $\boldsymbol{v}_i^{q_i}, \boldsymbol{a}_i^{q_i}$, and $\boldsymbol{t}_i^{q_i}$ $(i = 1, 2, \cdots, N, q_i = 1, 2, \cdots, M_i)$, and the number of clusters N_{clus}.

Output: Vectors $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t (i' = 1, 2, \cdots, N_{sub})$ that become calculation targets of CCA.

Denote index sets of vectors selected from $\boldsymbol{v}_i^{q_i}, \boldsymbol{a}_i^{q_i}$, and $\boldsymbol{t}_i^{q_i}$ by I_v, I_a and I_t.
$I_v \leftarrow \phi$, $I_a \leftarrow \phi$, $I_t \leftarrow \phi$.
for k in $\{v, a, t\}$ **do**
    Apply k-means clustering to $\boldsymbol{k}_i^{q_i} (i = 1, 2, \cdots, N, q_i = 1, 2, \cdots, M_i)$, and obtain cluster centers $\boldsymbol{k}_j^{cent} (j = 1, 2, \cdots, N_{clus})$.
    for j in $\{1, 2, \cdots, N_{clus}\}$ **do**
        Select a vector with the shortest Euclidean distance to $\boldsymbol{k}_j^{cent}$ from $\boldsymbol{k}_i^{q_i} (i = 1, 2, \cdots, N, q_i = 1, 2, \cdots, M_i)$.
        Add the index of the selected vector to I_k.
    end for
end for
Denote the index set of vectors that become calculation targets for CCA by I.
$I \leftarrow I_v \cup I_a \cup I_t$ $(|I| = N_{sub}, |I_v| = |I_a| = |I_t| = N_{clus})$
Select vectors corresponding to I from $\boldsymbol{v}_i^{q_i}, \boldsymbol{a}_i^{q_i}$, and $\boldsymbol{t}_i^{q_i}$, and denote these vectors by $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t (i' = 1, 2, \cdots, N_{sub})$.
Return $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t (i' = 1, 2, \cdots, N_{sub})$.

$(i' = 1, 2, \cdots, N_{sub}; N_{sub}$ being the number of selected vectors, where $N_{sub} = |I_v \cup I_a \cup I_t|)$. The detailed procedures for obtaining $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t$ are shown in Algorithm 2. Next, CCA is applied to these selected vectors, and latent features that can be compared between different kinds of features are defined as the following procedures.

First, three kinds of matrices $\boldsymbol{X}_v$, $\boldsymbol{X}_a$ and $\boldsymbol{X}_t$ are calculated by using $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t$ as follows:

$$\boldsymbol{X}_v = \left[\boldsymbol{x}_1^v - \bar{\boldsymbol{x}}_v, \boldsymbol{x}_2^v - \bar{\boldsymbol{x}}_v, \cdots, \boldsymbol{x}_{N_{sub}}^v - \bar{\boldsymbol{x}}_v\right]^{\mathrm{T}}, \tag{4.3}$$

$$\boldsymbol{X}_a = \left[\boldsymbol{x}_1^a - \bar{\boldsymbol{x}}_a, \boldsymbol{x}_2^a - \bar{\boldsymbol{x}}_a, \cdots, \boldsymbol{x}_{N_{sub}}^a - \bar{\boldsymbol{x}}_a\right]^{\mathrm{T}}, \tag{4.4}$$

$$\boldsymbol{X}_t = \left[\boldsymbol{x}_1^t - \bar{\boldsymbol{x}}_t, \boldsymbol{x}_2^t - \bar{\boldsymbol{x}}_t, \cdots, \boldsymbol{x}_{N_{sub}}^t - \bar{\boldsymbol{x}}_t\right]^{\mathrm{T}}, \tag{4.5}$$

where $\bar{\boldsymbol{x}}_v$, $\bar{\boldsymbol{x}}_a$ and $\bar{\boldsymbol{x}}_t$ are the average vectors of $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and $\boldsymbol{x}_{i'}^t$, respectively. Next, we calculate coefficient vectors $\boldsymbol{w}_k$ $(k \in \{v, a, t\})$, which maximize the correlation between the following

vectors $\boldsymbol{g}_k$:

$$\boldsymbol{g}_k = \boldsymbol{X}_k \boldsymbol{w}_k \quad (k \in \{v, a, t\}). \tag{4.6}$$

Specifically, by using a vector $\boldsymbol{y}$ in which each element is unknown, we estimate $\boldsymbol{w}_k$ that minimize

$$Q(\boldsymbol{y}, \boldsymbol{w}_k) = \sum_{k \in \{v,a,t\}} \|\boldsymbol{y} - \boldsymbol{X}_k \boldsymbol{w}_k\|^2 \ (\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} = 1). \tag{4.7}$$

Here, the following inequality is obtained by the least squares method:

$$Q(\boldsymbol{y}, \boldsymbol{w}_k) \geq \sum_{k \in \{v,a,t\}} \|\boldsymbol{y} - \boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{y}\|^2 \tag{4.8}$$

$$= 3\|\boldsymbol{y}\|^2 - \boldsymbol{y}^{\mathrm{T}} \sum_{k \in \{v,a,t\}} (\boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}}) \boldsymbol{y}, \tag{4.9}$$

with equality if and only if

$$\boldsymbol{w}_k \quad = \quad (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{y}. \tag{4.10}$$

Then we define

$$Q(\boldsymbol{y}) \quad = \quad 3\|\boldsymbol{y}\|^2 - \boldsymbol{y}^{\mathrm{T}} \sum_{k \in \{v,a,t\}} (\boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}}) \boldsymbol{y}. \tag{4.11}$$

Next, we need to maximize the second term of Eq. (4.11) to minimize $Q(\boldsymbol{y})$. Here, we can calculate the vector $\boldsymbol{y}$ as the solution of the following eigenvalue problem:

$$\sum_{k \in \{v,a,t\}} (\boldsymbol{X}_k (\boldsymbol{X}_k^{\mathrm{T}} \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^{\mathrm{T}}) \boldsymbol{y} = \lambda \boldsymbol{y}. \tag{4.12}$$

From this equation, N_e $(= \mathrm{rank}(\boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{X}_l),\ l \in \{v,a,t|l \neq k\})$ eigenvectors $\boldsymbol{y}_n$ $(n = 1,2,\cdots,N_e)$ are obtained according to N_e positive eigenvalues. Moreover, by using the eigenvectors $\boldsymbol{y}_n$, the coefficient vectors $\boldsymbol{w}_k^n$ are calculated on the basis of Eq. (4.10). Then the coefficient matrix $\boldsymbol{W}_k$ is defined as follows:

$$\boldsymbol{W}_k = [\boldsymbol{w}_k^1, \boldsymbol{w}_k^2, \cdots, \boldsymbol{w}_k^{N_e}]. \tag{4.13}$$

Then we obtain the following equation:

$$\begin{aligned} \boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{X}_l\boldsymbol{W}_l &= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{N_e} \end{bmatrix} && (4.14) \\ &= \boldsymbol{\Lambda}_{k,l}, && (4.15) \end{aligned}$$

where $\boldsymbol{\Lambda}_{k,l}$ is a correlation matrix in which each diagonal element is the canonical correlation coefficient λ_n $(n = 1,2,\cdots,N_e)$ between $\boldsymbol{W}_k\boldsymbol{X}_k$ and $\boldsymbol{W}_l\boldsymbol{X}_l$. Then the three kinds of variants are projected into the latent space of one of these variants. Specifically, by using $\boldsymbol{W}_k$, $\boldsymbol{\Lambda}_{k,l}$ and $\boldsymbol{k}_i^{q_i}$ $(\in \{\boldsymbol{v}_i^{q_i} - \bar{\boldsymbol{x}}_v, \boldsymbol{a}_i^{q_i} - \bar{\boldsymbol{x}}_a, \boldsymbol{t}_i^{q_i} - \bar{\boldsymbol{x}}_t\})$, $\boldsymbol{\zeta}^l_{\boldsymbol{k}_i^{q_i}}$, *i.e.*, the projection results of $\boldsymbol{k}_i^{q_i}$ into the latent space of $\boldsymbol{l}_i^{q_i}$ $(\in \{\boldsymbol{v}_i^{q_i} - \bar{\boldsymbol{x}}_v, \boldsymbol{a}_i^{q_i} - \bar{\boldsymbol{x}}_a, \boldsymbol{t}_i^{q_i} - \bar{\boldsymbol{x}}_t\})$, are calculated as follows:

$$\boldsymbol{\zeta}^l_{\boldsymbol{k}_i^{q_i}} = \begin{cases} \boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{k}_i^{q_i} & \text{if} \quad k = l \\ \boldsymbol{\Lambda}_{l,k}\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{k}_i^{q_i} & \text{otherwise.} \end{cases} \tag{4.16}$$

Hence, from the visual feature vector $\boldsymbol{v}_i^{q_i}$, the audio feature vector $\boldsymbol{a}_i^{q_i}$ and the textual feature vector $\boldsymbol{t}_i^{q_i}$, we obtain the vectors $\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}}$, $\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}}$ and $\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}}$ that can be compared between different kinds of features. In the experiments shown later, visual, audio and textual feature vectors are projected into the latent space of textual features by setting $l = t$ in Eq. (4.16). Note that the projection results of all feature vectors $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$ $(i = 1,2,\cdots,N,\ q_i = 1,2,\cdots,M_i)$ can be obtained by applying CCA to a small number of vectors $\boldsymbol{x}_{i'}^v$, $\boldsymbol{x}_{i'}^a$ and

$\boldsymbol{x}_{i'}^{t}$ $(i' = 1, 2, \cdots, N_{sub})$ selected on the basis of k-means clustering. In this way, we reduce the number of calculation targets for CCA, and latent features $\boldsymbol{\zeta}_{\boldsymbol{v}_i^{q_i}}^{l}$, $\boldsymbol{\zeta}_{\boldsymbol{a}_i^{q_i}}^{l}$ and $\boldsymbol{\zeta}_{\boldsymbol{t}_i^{q_i}}^{l}$ $(i = 1, 2, \cdots, N, q_i = 1, 2, \cdots, M_i)$ can be efficiently obtained.

4.3 Efficient Extraction of Hierarchical Structure of Web Video Groups

In this section, an efficient extraction method of hierarchical structure of Web video groups is presented. First, we construct "a group graph" of which each node consists of multiple Web videos, and each edge corresponds to links between Web videos by using the latent features $\boldsymbol{\zeta}_{\boldsymbol{v}_i^{q_i}}^{l}$, $\boldsymbol{\zeta}_{\boldsymbol{a}_i^{q_i}}^{l}$ and $\boldsymbol{\zeta}_{\boldsymbol{t}_i^{q_i}}^{l}$. Then hierarchical structure of Web video groups is estimated on the basis of SCCs [20], edge betweenness [22] and modularity [21] of the group graph. The above procedures are explained in detail below.

4.3.1 Construction of Group Graph

In this section, we explain a method for construction of a group graph. First, we build link relationships between Web videos based on metadata "related videos". In particular, we consider that a Web video f_i links to a Web video f_j if "related videos" of f_i include f_j. Then the link relationships are weighted by using the latent features $\boldsymbol{\zeta}_{\boldsymbol{v}_i^{q_i}}^{l}$, $\boldsymbol{\zeta}_{\boldsymbol{a}_i^{q_i}}^{l}$ and $\boldsymbol{\zeta}_{\boldsymbol{t}_i^{q_i}}^{l}$. Specifically, the weighted adjacency matrix $\boldsymbol{L} = (l_{ij})$ $(i = 1, 2, \cdots, N,\ j = 1, 2, \cdots, N)$ is calculated as follows:

$$l_{ij} = \begin{cases} sim_{ij} & \text{if } f_i \text{ links to } f_j \\ 0 & \text{otherwise} \end{cases}, \tag{4.17}$$

$$sim_{ij} = \max_{q_i, q_j} \left| \frac{(\boldsymbol{\xi}_i^{q_i})^{\mathrm{T}} \boldsymbol{\xi}_j^{q_j}}{\|\boldsymbol{\xi}_i^{q_i}\| \|\boldsymbol{\xi}_j^{q_j}\|} \right|, \tag{4.18}$$

$$\boldsymbol{\xi}_i^{q_i} = \left[(\boldsymbol{\zeta}_{\boldsymbol{v}_i^{q_i}}^{l})^{\mathrm{T}}, (\boldsymbol{\zeta}_{\boldsymbol{a}_i^{q_i}}^{l})^{\mathrm{T}}, (\boldsymbol{\zeta}_{\boldsymbol{t}_i^{q_i}}^{l})^{\mathrm{T}} \right]^{\mathrm{T}}, \tag{4.19}$$

$$i = 1, 2, \cdots, N, \quad j = 1, 2, \cdots, N.$$

Furthermore, between Web videos f_i and f_j, we calculate Tanimoto coefficients S_{ij} [38] that represent similarities of link structure between Web videos as follows:

$$
\begin{aligned}
S_{ij} &= \frac{\boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\alpha}_j}{\|\boldsymbol{\alpha}_i\|^2 + \|\boldsymbol{\alpha}_j\|^2 - \boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\alpha}_j}, && (4.20)\\
\boldsymbol{\alpha}_i &= \left[\tilde{A}_{i1}, \cdots, \tilde{A}_{iN}\right]^{\mathrm{T}}, && (4.21)\\
\tilde{A}_{ij} &= \frac{1}{|n(i)|} \sum_{k \in n(i)} l_{ik}\delta_{ij} + l_{ij}, && (4.22)\\
i &= 1, 2, \cdots, N,\ j = 1, 2, \cdots, N,
\end{aligned}
$$

where $n(i) = \{k \,|l_{ik} > 0\}$ and δ_{ij} is Kronecker delta. When the link structure between Web videos f_i and f_j is similar, S_{ij} becomes close to the maximum of S_{ij}, *i.e.*, one. Next, Web videos whose Tanimoto coefficients are high are grouped, and Web video sets are obtained. Specifically, for a Web video f_i, Web videos f_j that satisfy $S_{ij} > Th$ (Th being the manually set threshold) are grouped, and then Web video sets containing more than N'_m videos are selected. In this chapter, the selected Web video sets are denoted by C_m ($m = 1, 2, \cdots, N'_g$; N'_g being the number of obtained Web video sets). In this way, we obtain representative Web videos from many Web videos by extracting Web video sets.

Moreover, we construct a group graph, *i.e.*, a weighted and directed graph of which each node is a Web video set containing multiple Web videos and each edge corresponds to links between Web videos. In this chapter, the group graph is denoted by $G = (V, E)$ ($V \in \{C_1, C_2, \cdots, C_{N'_g}\}$), and the edge weight $e_{ij} \in E$ from a node C_i to a node C_j is defined as follows:

$$
e_{ij} = \frac{1}{|C_i||C_j|} \sum_{f_m \in C_i, f_n \in C_j} l_{mn}, \tag{4.23}
$$

where $|C_i|$ is the number of Web videos contained in C_i. Thereby, the edge weight is calculated on the basis of the similarities between Web videos. When Web videos that belong to $C_i \in V$ do not link to Web videos contained in $C_j \in V$, we do not define the edge from C_i to C_j.

Since each node of the group graph G consists of multiple Web videos, we can handle many Web videos by the group graph with a small number of nodes.

4.3.2 Efficient Extraction of Hierarchical Structure of Web Video Groups

In this section, an efficient extraction method of hierarchical structure of Web video groups using the group graph G is described. It should be noted that the proposed scheme is realized by replacing each Web video in the scheme presented in Section 3.4 by each node of the group graph G, which contains multiple Web videos. First, we extract "parent Web video groups", which are Web videos belonging to closely related Web video sets C_m $(m = 1, 2, \cdots, N'_g)$, by using SCCs [20]. SCCs are subgraphs with a directed path between any two nodes in a directed graph, and a scheme for extracting SCCs from a graph is called SCC decomposition. In the method [20], related Web pages are grouped on the basis of SCC decomposition. In the proposed method, all SCCs are extracted from the group graph G by applying SCC decomposition to G. Then Web videos contained in these obtained SCCs are defined as parent Web video groups. Thus, we can extract Web videos belonging to closely related Web video sets C_m.

Furthermore, we iteratively divide the parent Web video groups into "child Web video groups", *i.e.*, Web videos belonging to more closely related Web video sets C_m, based on edge betweenness [22]. Edge betweenness is the rate at which an edge exists on the shortest path between two nodes. Thereby, since edge betweenness is useful for detecting influential edges in a graph, the method [22] detects Web video groups by using edge betweenness. Suppose that $c_B(u)$ is edge betweenness of an edge $u \in E$, then $c_B(u)$ is defined as follows:

$$\begin{aligned} c_B(u) &= \sum_{s,t \in V} \frac{\sigma(s,t|u)}{\sigma(s,t)}, \\ u &\in E, \end{aligned} \tag{4.24}$$

where $\sigma(s,t|u)$ is the number of shortest paths from a node s to a node t that pass an edge u, and $\sigma(s,t)$ is the total number of shortest paths from a node s to a node t. The proposed method

iteratively removes edges with the largest edge betweenness in G, and reapplies SCC decomposition to G. When a single SCC is divided into several SCCs, child Web video groups are obtained from their parent Web video groups. In this chapter, the number of these cutting edge iterations is denoted by N_{cut}. In this way, the proposed method can estimate the hierarchical structure of Web video groups by repeatedly diving the parent Web video groups including various topics into child Web video groups containing more closely related topics. Although the calculation cost of edge betweenness is high, we can efficiently extract the hierarchical structure by using the group graph G with a small number of nodes.

It should be noted that when all edges in G are cut, each child Web video group becomes equivalent to each Web video set, *i.e.*, each node of the group graph G. Since it is not effective to provide all Web video groups until all edges in G have been cut when Web video retrieval is performed, we have to decide how many hierarchies of the Web video groups to provide. In order to meet this requirement, our method uses modularity, *i.e.*, a quality function that evaluates the division results of communities in a graph. Suppose that $Q_{N_{cut}}$ is modularity of G, where the number of cut edges is N_{cut}, and $Q_{N_{cut}}$ is defined as follows [21]:

$$Q_{N_{cut}} = \frac{1}{2e} \sum_{i=1}^{N'_g} \sum_{j=1}^{N'_g} (e_{ij} - \frac{e_i^{out} e_j^{in}}{2e})\ \delta(i,j), \tag{4.25}$$

where

$$2e = \sum_{i=1}^{N'_g} \sum_{j=1}^{N'_g} e_{ij}, \tag{4.26}$$

$$e_i^{out} = \sum_{j=1}^{N'_g} e_{ij}, \tag{4.27}$$

$$e_j^{in} = \sum_{i=1}^{N'_g} e_{ij}, \tag{4.28}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } C_i \text{ and } C_j \text{ belong to} \\ & \text{the same Web video group.} \\ 0 & \text{otherwise} \end{cases} \tag{4.29}$$

Algorithm 3 : Efficient Extraction of Hierarchical Structure Web Video Groups

Input: A group graph $G = (V, E)$.
Output: Web video groups.

1: $N_{cut} \leftarrow 0$.
2: Apply SCC decomposition to G, and extract parent Web video groups.
3: Calculate modularity $Q_{N_{cut}}$.
4: **while** an edge of G remains **do**
5: Calculate edge betweenness of remaining edges in G.
6: **if** only an edge has the largest edge betweenness **then**
7: Remove the edge with the largest edge betweenness.
8: **else**
9: Select one edge from the edges with the largest edge betweenness randomly, and remove the edge.
10: **end if**
11: $N_{cut} \leftarrow N_{cut} + 1$.
12: Apply SCC decomposition to G.
13: **if** the number of SCCs is changed **then**
14: Extract child Web video groups.
15: **end if**
16: Calculate modularity $Q_{N_{cut}}$.
17: **end while**
18: $N_{cut}^{opt} \leftarrow \arg\max_{N_{cut}} Q_{N_{cut}}$.
19: Return Web video groups where N_{cut} is from 0 to N_{cut}^{opt}.

When community structure is significant, $Q_{N_{cut}}$ becomes close to 1, *i.e.*, the maximum value of $Q_{N_{cut}}$. On the other hand, $Q_{N_{cut}}$ becomes close to 0 when a graph is divided into Web video groups randomly. In this chapter, the N_{cut} value when $Q_{N_{cut}}$ is maximum is represented as N_{cut}^{opt}. Then we use the Web video groups where N_{cut} is from 0 to N_{cut}^{opt} for Web video retrieval. Thus, the number of hierarchies of Web video groups to present to users can be decided by using modularity.

In this way, the proposed method extracts the hierarchical structure of Web video groups based on SCCs, edge betweenness and modularity of the group graph G. Since each node of G consists of multiple Web videos and many Web videos can be handled by a small number of nodes, we can efficiently obtain the hierarchical structure of Web video groups containing many Web videos. The detailed procedures for efficiently extracting the hierarchical structure of Web video groups are presented in Algorithm 3.

4.4 Web Video Retrieval Using Hierarchical Structure of Web Video Groups

In this section, a Web video retrieval scheme using the hierarchical structure of Web video groups is described. First, Web video groups, where N_{cut} is from 0 to N_{cut}^{opt}, are obtained. In this chapter, we denote these Web video groups by $Com_{N_{cut}}^{n}$ ($N_{cut} = 0, 1, \cdots, N_{cut}^{opt},\ n = 1, 2, \cdots, T_{N_{cut}}; T_{N_{cut}}$ being the number of Web video groups in which the number of cut edges is N_{cut}). Then each Web video f_i ($i \in \{1, 2, \cdots, N\}$) contained in Web video sets belonging to the Web video group $Com_{N_{cut}}^{n}$ is ranked in descending order of the following criterion $R_{N_{cut}}^{n}(i)$:

$$R_{N_{cut}}^{n}(i) = \sum_{j=1}^{N} l_{ji}\,\delta(i,j), \tag{4.30}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ belong to} \\ & \text{the same Web video group.} \\ 0 & \text{otherwise} \end{cases} \tag{4.31}$$

Thus, Web videos contained in the Web video groups are ranked based on links from other Web videos in the same Web video group.

Next, the proposed method provides users Web video groups $Com_{N_{cut}}^{n}$, where N_{cut} is from 0 to N_{cut}^{opt}. Note that the smaller N_{cut} is, the more various topics are included in the Web video groups, and the larger N_{cut} is, the more closely related topics are contained in the Web video groups. Then users select Web video groups including their desired Web videos according to the hierarchical structure. Moreover, users retrieve their desired Web videos from the selected Web video group on the basis of the criterion $R_{N_{cut}}^{n}(i)$. In the proposed method, users can easily select Web video groups including the desired Web videos by using the hierarchical structure. Consequently, it becomes feasible for users to retrieve desired Web videos by using the hierarchical structure of Web video groups even if users cannot write suitable keywords as a query.

Table 4.1: Detailed conditions of each dataset: K is Num. of keywords that appear in text of Web videos, U is Num. of dimensions after random projection, N_{clus} is Num. of cluster centers in k-means clustering.

	Query keyword	Num. of Web videos in datasets	Num. of shots in datasets	K	U	N_{clus}
Dataset 1	sightseeing	3001	10249	10832	1000	1000
Dataset 2	game	3003	9644	10255	1000	1000

4.5 Experimental Results

In this section, experimental results for Web videos collected by using YouTube are shown to verify the effectiveness of the proposed method.

4.5.1 Settings

First, we prepared datasets, which were same as those used in the experiments shown in Chapter 3. Table 4.1 shows the details of the datasets. In this experiment, the following methods were compared with each other for verifying the effectiveness of the proposed method.

(P-1)

This is the proposed method that constructs the group graph which contains about 1000 Web videos.

(P-2)

This is the proposed method that constructs the group graph which contains about 1500 Web videos.

(R1-1)

This is a method based on extraction of hierarchical structure of Web video groups using a group graph. However, this method uses only link relationships between Web videos and does not use Web video features. For a fair comparison, parameters Th and N'_m were defined as in (P-1).

(R1-2)

This is the same method as (R1-1). However, parameters Th and N'_m were defined as in (P-2).

(R2-1)

This is the proposed method in Chapter 3 that performs extraction of hierarchical structure of Web video groups. Although this method calculates latent features of Web videos, k-means clustering before CCA is not performed and this method does not use a group graph. This method performs screening of Web videos to select about 1000 Web videos.

(R2-2)

This is the same method as (R2-1). However, this method performs screening of Web videos to select about 1500 Web videos.

(R3)

This is a conventional method [13] that extracts Web video groups but does not extract their hierarchical structure.

(R4)

This is a method based on [12] with a Web video group extraction scheme by affinity propagation [29]. This method uses only Web video similarities without the use of link relationships between Web videos, and does not extract the hierarchical structure. In the experiment, we did not use the original similarities presented by [12] but utilized our proposed similarities defined in Eq. (4.18). For the evaluation, we ranked Web videos within Web video groups obtained by this method in descending order of the sum of the above similarities.

(R5)

This is a method based on clustering via latent semantic indexing (LSI) that uses only textual features of Web videos [14], which does not present the hierarchical structure to users. Note that an original method in [14] utilizes information of YouTube Playlists. In

Table 4.2: Parameter settings: Th and N'_m are parameters to perform screening of Web videos for (P-1) and (P-2). N'_g is Num. of the extracted Web video sets, *i.e.*, Num. of nodes of a group graph. Also, N_m and N_g are parameters to perform the screenings for (R2-1) and (R2-2) (see Chapter 3). Note that Web videos after performing the screening are called "subsets".

(a) Dataset 1.

	Th	N'_m	N'_g	N_m	N_g	Num. of Web videos in subsets
(P-1)	0.2	9	77	—	—	1021
(P-2)	0.2	7	129	—	—	1410
(R2-1)	—	—	—	80	20	1038
(R2-2)	—	—	—	100	40	1584

(b) Dataset 2.

	Th	N'_m	N'_g	N_m	N_g	Num. of Web videos in subsets
(P-1)	0.2	10	67	—	—	1033
(P-2)	0.2	7	134	—	—	1561
(R2-1)	—	—	—	80	20	1079
(R2-2)	—	—	—	100	40	1509

this experiment, however, we used a video-keyword vectors whose elements are 1 if each keyword appears in title and description of the Web video and 0 otherwise. Moreover, the number of dimensions of the video-keyword vectors after LSI were set to 200 as in a paper [14]. Each Web video is ranked on the basis of the attribution degree to the belonging cluster.

The details of parameter settings for (P-1), (P-2), (R2-1) and (R2-2) are shown in Table 4.2.

4.5.2 Evaluations

Under the above settings, we verify the effectiveness of the proposed method. First, from each dataset, we constructed the group graph by using latent features of Web videos. Then, by using the group graph, we extracted hierarchical structure of Web video groups from each subset. Table 4.3 shows N_{cut}^{opt} and the maximum value of $Q_{N_{cut}}$ of each subset. Furthermore, each obtained hierarchical structure of Web video groups is shown in Figs. 4.1 and 4.2. Note that we named each Web video group by checking contents of them manually since the proposition in this thesis is to extract the hierarchical structure and does not include the naming scheme. Since Web video groups with various topics are divided into ones with similar topics, we can see that the proposed method enables extraction of the hierarchical structure.

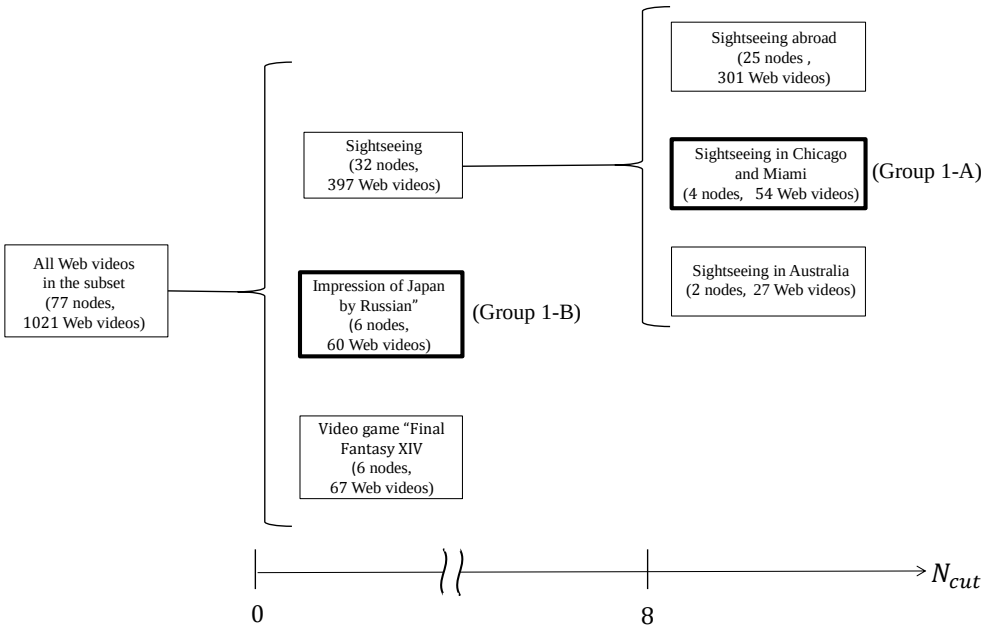
Table 4.3: N_{cut}^{opt} and maximum value of $Q_{N_{cut}}$ for each subset.

(a) Results for (P-1).

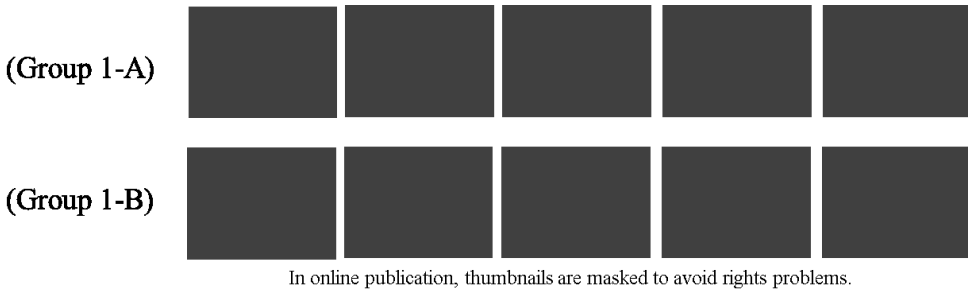
	N_{cut}^{opt}	Maximum value of $Q_{N_{cut}}$
Subset 1	9	0.794
Subset 2	3	0.815

(b) Results for (P-2).

	N_{cut}^{opt}	Maximum value of $Q_{N_{cut}}$
Subset 1	42	0.791
Subset 2	14	0.829



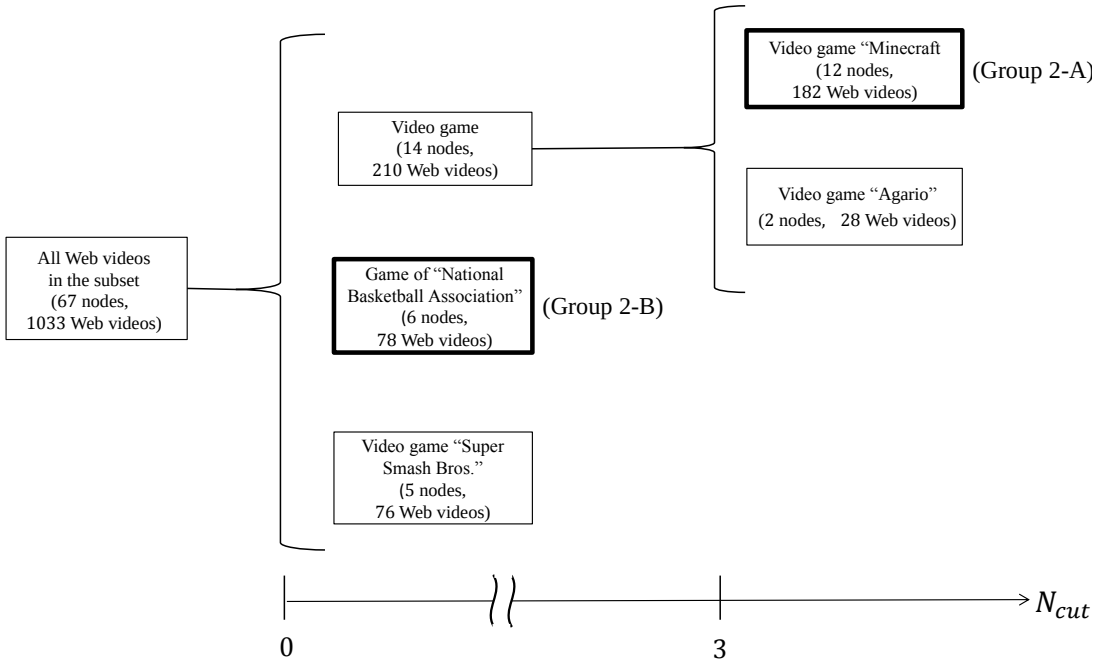
(a) Hierarchical structure of Web video groups.



In online publication, thumbnails are masked to avoid rights problems.

(b) Thumbnails of the retrieved Web videos when (Group 1-A) and (Group 1-B) were respectively selected. The top five ranked Web videos were located in the order of left to right

Figure 4.1: Hierarchical structure of Web video groups for subset 1, which were obtained by (P-1).



(a) Hierarchical structure of Web video groups.

In online publication, thumbnails are masked to avoid rights problems.

(b) Thumbnails of the retrieved Web videos when (Group 2-A) and (Group 2-B) were respectively selected. The top five ranked Web videos were located in the order of left to right

Figure 4.2: Hierarchical structure of Web video groups for subset 2, which were obtained by (P-1).

Table 4.4: Relevant Web videos used for drawing Figs 4.3 and 4.4.

	Relevant Web videos
Dataset 1	Web videos with Freebase topics about "Sightseeing in Japan"
Dataset 2	Web videos with Freebase topics about "Game of National Basketball Association"

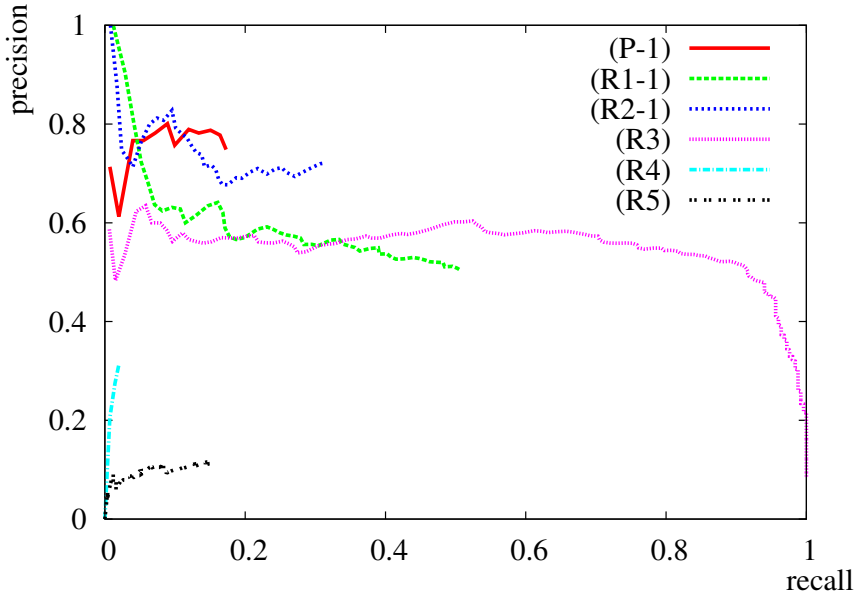
Next, the effectiveness of the proposed method is quantitatively evaluated. We used recall, precision and average precision (AP@k) defined as follows:

$$\text{Recall} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of relevant Web videos}}, \tag{4.32}$$

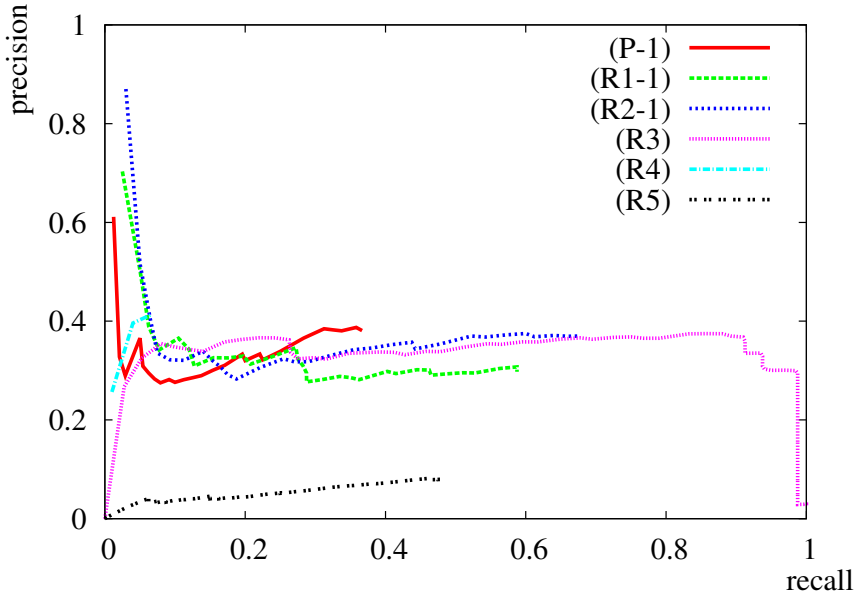
$$\text{Precision} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of retrieved Web videos}}, \tag{4.33}$$

$$\text{AP@}k = \frac{1}{R_k}\sum_{i=1}^{k} x_i \, prec_i, \tag{4.34}$$

where k is the number of Web videos provided as the retrieval results, R_k is the number of "relevant Web videos" within k Web videos of the retrieval results, x_i is 1 if the i-th retrieved Web videos are "relevant Web videos" and 0 otherwise, and $prec_i$ is the precision when i Web videos are retrieved. For all methods shown in the previous section, we selected a Web video group that included relevant Web videos the most, and Web videos were retrieved according to the rankings of Web videos. Then recall and precision were calculated and Figs. 4.3 and 4.4 show precision-recall curves for the datasets. For drawing these figures, relevant Web videos were defined as shown in Table 4.4. Here, we defined ground truths as in the experiments shown in Chapter 3. Also, Table 4.5 shows the averages of AP@k. Here, we selected the most frequent Freebase topics within each Web video group as ground truths and calculated the averages, which were weighted by the numbers of Web videos in Web video groups. By comparing (P-1) and (P-2) with (R3), (R4) and (R5), which does not extract the hierarchical structure of Web video groups, we can see that accurate retrieval becomes feasible by using the hierarchical structure. Furthermore, the effectiveness of using latent features of Web videos can be seen when comparing (P-1) and (P-2) with (R1-1), (R1-2) and (R5). It is remarkable

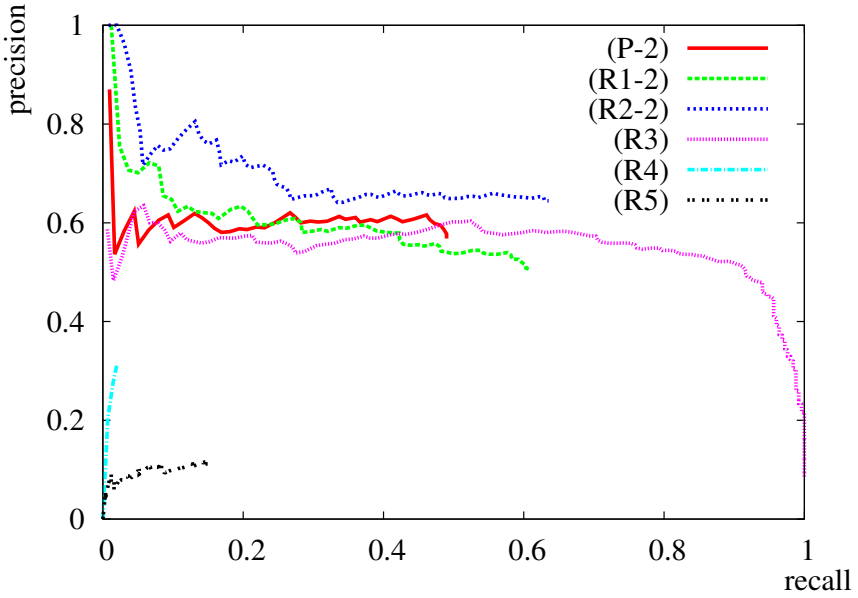


(a) Results for dataset 1.

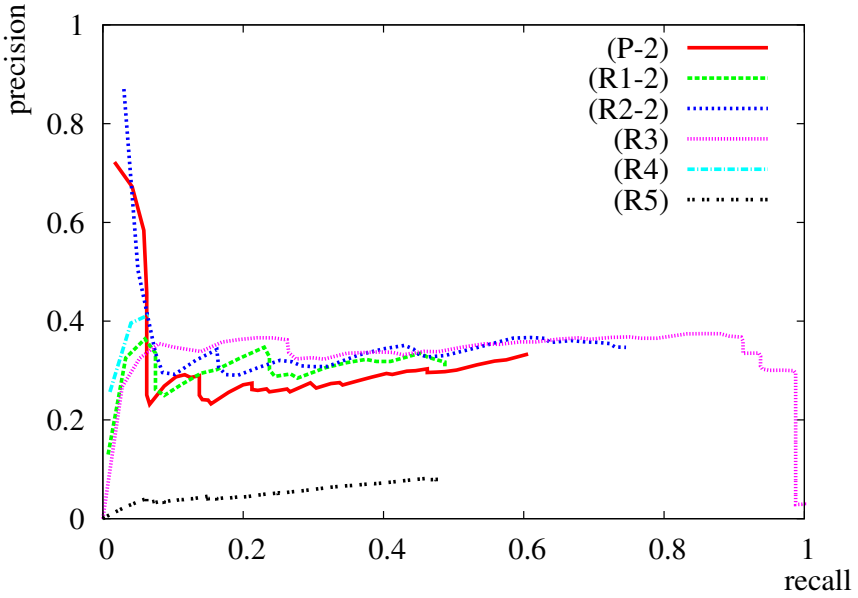


(b) Results for dataset 2.

Figure 4.3: Precision-recall curves for (P-1), (R1-1), (R2-1), (R3), (R4) and (R5): (P-1), (R1-1) and (R2-1) respectively correspond to the results where N_{cut} were N_{cut}^{opt}. Note that (R3) has higher recall than other methods since only this method is based on soft clustering and all Web videos can belong to every Web video group.



(a) Results for dataset 1.



(b) Results for dataset 2.

Figure 4.4: Precision-recall curves for (P-2), (R1-2), (R2-2), (R3), (R4) and (R5): (P-2), (R1-2) and (R2-2) respectively correspond to the results where N_{cut} were N_{cut}^{opt}. Note that (R3) has higher recall than other methods since only this method is based on soft clustering and all Web videos can belong to every Web video group.

Table 4.5: AP@k weighted by the numbers of Web videos in each Web video group. For (P-1), (P-2), (R1-1), (R1-2), (R2-1) and (R2-2), the results where N_{cut} were N_{cut}^{opt} are shown. For (R3) based on soft clustering, ideal values are shown since we selected Freebase topics that provided the highest AP@k of all as ground truths. Also, k was defined as the number of Web videos in each Web video group.

	Proposed method		Method without the use of Web video features		Method proposed in Chapter 3		Reference method [13]	Reference method [12]	Reference method [14]
	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)	(R3)	(R4)	(R5)
Dataset 1	0.530	0.585	0.532	0.452	**0.765**	0.636	0.632	0.496	0.320
Dataset 2	0.822	0.726	0.612	0.654	**0.851**	0.841	0.563	0.640	0.373
Average	0.666		0.563		**0.774**		0.598	0.568	0.347

that the proposed method can reduce the computational cost of extraction of the hierarchical structure and perform more accurate retrieval than methods without the use of Web video features or the hierarchical structure. On the other hand, the retrieval accuracy of (P-1) and (P-2) is lower than that of (R2-1) and (R2-2), which are the methods presented in Chapter 3. However, the effectiveness of the proposed method can be seen in the the computational efficiency shown below.

Next, the efficiency on extracting the hierarchical structure via the proposed method is verified. Table 4.6 shows computational time for obtaining the hierarchical structure by (P-1), (P-2), (R1-1), (R1-2), (R2-1) and (R2-2), and the explanation of this table are described below.

(A) Since this preprocessing is common to (P-1), (P-2), (R2-1) and (R2-2), this computational time is same.

(B) Although (P-1) and (P-2) need more computational time than (R2-1) and (R2-2) since our sub-sampled CCA includes k-means clustering, (P-1) and (P-2) can reduce the number of the target vectors for CCA with high computational cost. Therefore, (P-1) and (P-2) have higher scalability than (R2-1) and (R2-2).

(C) Although (P-1),(P-2), (R1-1) and (R1-2) need to construct a group graph unlike (R2-1) and (R2-2), this can be computed faster by multi-thread parallel computation of Eq. (4.20).

(D) (P-1), (P-2), (R1-1) and (R1-2) can efficiently extract the hierarchical structure by using

Table 4.6: Comparison of computational time between (P-1), (P-2), (R1-1), (R1-2), (R2-1) and (R2-2): we used a computer with Intel® CPU 2.50GHz and 32GB RAM for (A). For (B), (C), (D) and (E), Intel® CPU 3.50GHz and 16GB RAM is used. Units of each element in this table are the second.

(A) Web video feature extraction. (The image size was 320 × 180 pixels and the frame rate was 25 fps. The audio signal was sampled at 22.05 kHz.)

	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)
Visual feature	54.570	54.570	—	—	54.570	54.570
Audio feature	16.940	16.940	—	—	16.940	16.940
Textual feature[1]	0.020000	0.020000	—	—	0.020000	0.020000

(B) Derivation of CCA-based link relationships between Web videos (see Eq. (4.17)). Note that (P-1) and (P-2) include k-means clustering but (R2-1) and (R2-2) do not.

	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)
Dataset 1	490.73	490.73	—	—	404.18	404.18
Dataset 2	432.79	432.79	—	—	418.83	418.83

(C) Screening of Web videos for extracting hierarchical structure of Web video groups. For (P-1), (P-2), (R1-1) and (R1-2), computational time for constructing the group graph is shown and the upper values in this table show the results without parallel computation and [ ] denote ones with eight threads parallel computation[2] of Eq. (4.20). For (R2-1) and (R2-2), we describe computational time for selecting representative Web videos[3].

	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)
Dataset 1	1581.1 [351.60]	1645.8 [335.71]	1625.7 [340.35]	1611.9 [338.51]	0.94974	1.1769
Dataset 2	1492.1 [348.03]	1495.8 [338.92]	1482.5 [341.72]	1469.4 [335.10]	0.94616	1.7133

(D) Repeated division based on SCCs, edge betweenness and modularity for extracting the hierarchical structure of Web video groups.

	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)
Dataset 1	0.12250	1.4461	0.99123	4.2181	325.31	2935.4
Dataset 2	0.024504	0.54902	0.72954	5.6310	802.07	2821.5

(E) Total time for obtaining the hierarchical structure except for Web video feature extraction, sum of (B), (C) and (D): the upper values show the results without parallel computation and [ ] denote ones with eight threads parallel computation[2] in (C).

	(P-1)	(P-2)	(R1-1)	(R1-2)	(R2-1)	(R2-2)
Dataset 1	2071.9 [842.46]	2138.0 [827.88]	1626.6 [341.34]	1616.1 [342.73]	730.44	3340.8
Dataset 2	1924.9 [780.84]	1929.2 [772.25]	1483.2 [342.45]	1475.1 [340.73]	1221.9	3242.0

the group graph.

(E) When comparing (P-1) with (R2-1), there is a case when (P-1) needs more computational

[1]Time to obtain $\boldsymbol{\eta}_i$ for one Web video in Section 4.2.1 is shown.

[2]The implementation was performed by using multiprocessing interface of Python.

[3]Calculation time of N_g eigenvectors of $\boldsymbol{L}_w^{\mathrm{T}}\boldsymbol{L}_w$ is described (see Chapter 3).

time than (R2-1) (see dataset 1). However, computational time for (P-1) will be reduced if more threads are used for parallel computation in **(C)**. Moreover, for reference methods, computational time increases too much if the number of target Web videos increases (see (R2-2)). On the other hand, the proposed method realizes fast computation even in such a case (see (P-2)). Thus, it can be seen that the proposed method enables more efficient extraction of the hierarchical structure than reference methods.

Here, we describe more discussion of computational efficiency. If Web videos are added and removed, it is necessary to recalculate CCA in **(B)**, **(C)** and **(D)** whereas **(A)** and k-means clustering in **(B)** can be incrementally computed. Moreover, although the computational cost of CCA in **(B)** and **(D)** is polynomial time [22], these cannot be parallelized easily. Therefore, it becomes difficult to reduce the computational cost of (R2-1) and (R2-2) if the number of Web videos becomes larger. Meanwhile, the proposed method can handle such many Web videos by performing k-means clustering before CCA and constructing the group graph before **(D)**. Moreover, since the computational cost of k-means clustering and the construction of the group graph can be reduced, the total computational cost of the proposed method can be also reduced. As a result, it can be seen that the proposed method can extract the hierarchical structure more efficiently than (R2-1) and (R2-2). When comparing (P-1) and (P-2) with (R1-1) and (R2-2), the computation time is close. However, the retrieval accuracy of (P-1) and (P-2) is superior to that of (R1-1) and (R1-2) as shown in Figs. 4.3 and 4.4 and Table 4.5.

Consequently, it can be seen that the proposed method enables efficient extraction of the hierarchical structure of Web video groups by using a group graph. Also, the experimental results have verified that the proposed method can reduce the computational cost of extraction of the hierarchical structure and perform more accurate retrieval than methods without the use of Web video features or the hierarchical structure. In the next chapter, a method that enables improvement of the retrieval accuracy as well as the computational efficiency is proposed.

4.5.3 Conclusions

This chapter has proposed a method for efficiently extracting hierarchical structure of Web video groups. The proposed method introduced sub-sampled CCA to enable reduction of computational cost of similarity calculation between latent features of Web videos. Furthermore, graph analysis based on SCC, edge betweenness and modularity can be efficiently realized by introducing a group graph that can analyze many Web videos by a small number of nodes. In the next chapter, a method for realizing further improvement concerning accuracy and efficiency is presented.

Chapter 5

Accurate and Efficient Extraction of Hierarchical Structure of Web Video Groups for Web Video Retrieval

5.1 Introduction

This chapter proposes a method for accurately and efficiently extracting the hierarchical structure of Web video groups by improving the method presented in the previous chapter. The proposed method adopts sub-sampled CCA as well as the method in the previous chapter to efficiently obtain latent features of Web videos. Furthermore, the proposed method constructs a graph whose edges represent similarities between latent features of Web videos, and then performs an algorithm based only on local structure of the graph to extract the hierarchical structure. Different from the method in the previous chapter, the proposed method can extract the hierarchical structure for the whole target dataset since the algorithm enables recursive reduction of its processing targets. This means it becomes unnecessary to perform screening of Web videos, and we can avoid performance degradation caused by discarding relevant Web videos in the screening, which occurred in the previously reported methods. Consequently, it becomes feasible to extract the hierarchical structure with high accuracy as well as low computational cost.

5.2 Outline of the Proposed Method

The proposed method consists of the following two phases.

Phase I: Derivation of CCA-based Link Relationships between Web Videos

We calculate CCA [19]-based link relationships via visual, audio and textual features, which represent similarities between latent features of Web videos. Here, efficient CCA, named sub-sampled CCA, is derived to obtain the canonical correlation for large training pairs with low computational cost according to the proposed method in Section 4.2.

Phase II: Accurate and Efficient Extraction of Hierarchical Structure of Web Video Groups

By constructing a graph whose nodes are Web videos based on the obtained link relationships, we enable application of the algorithm based on recursive modularity optimization [24] to extract the hierarchical structure of Web video groups. Here, screening of Web videos for reducing computational cost becomes unnecessary since the graph analysis algorithm enables recursive reduction of the target nodes. Thus, we can avoid the performance degradation caused by discarding relevant Web videos in the screening, which occurred in the methods presented in Chapters 3 and 4. Finally, users can retrieve the desired Web videos by selecting Web video groups associated with the desired Web videos according to the hierarchical structure.

The following sections show these details.

5.3 Phase I: Derivation of CCA-based Link Relationships between Web Videos

According to the proposed scheme in Section 4.2.1, for each shot $s_i^{q_i}$ ($q_i = 1, 2, \cdots, M_i; M_i$ being the number of shots within f_i), we obtain visual, audio and textual feature vectors $\boldsymbol{v}_i^{q_i}$, $\boldsymbol{a}_i^{q_i}$ and $\boldsymbol{t}_i^{q_i}$, respectively. Furthermore, by using the three kinds of features, we obtain the latent feature vectors $\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}}$, $\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}}$ and $\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}}$ that can be compared between different kinds of features via efficient CCA [19], named sub-sampled CCA, according to the proposed scheme in Section 4.2.2. Next, we compute similarities s_{ij} between Web videos f_i and f_j via the

obtained latent features as follows:

$$s_{ij} = \max_{q_i, q_j} \left| \frac{(\boldsymbol{\xi}_i^{q_i})^{\mathrm{T}} \boldsymbol{\xi}_j^{q_j}}{\|\boldsymbol{\xi}_i^{q_i}\| \|\boldsymbol{\xi}_j^{q_j}\|} \right|, \tag{5.1}$$

$$\boldsymbol{\xi}_i^{q_i} = \left[(\boldsymbol{\zeta}^l_{\boldsymbol{v}_i^{q_i}})^{\mathrm{T}}, (\boldsymbol{\zeta}^l_{\boldsymbol{a}_i^{q_i}})^{\mathrm{T}}, (\boldsymbol{\zeta}^l_{\boldsymbol{t}_i^{q_i}})^{\mathrm{T}} \right]^{\mathrm{T}}. \tag{5.2}$$

Moreover, we build link relationships between Web videos based on the obtained similarities and the metadata of Web videos, namely "related videos". In particular, we consider that a Web video f_i links to a Web video f_j if "related videos" of f_i include f_j. Although the only use of the low-level features extracted from Web videos may cause the semantic gap [25], *i.e.*, the difference between the low-level features and high-level interpretation of humans, the metadata "related videos" is useful for obtaining Web videos that are similar to each other [35]. Therefore, we introduce the metadata into the proposed method. Then we construct a graph $G = (V, E)$ whose nodes and edges are respectively Web videos f_i $(i = 1, 2, \cdots, N)$ and weighted links. The edge weight e_{ij} between Web videos f_i and f_j is defined as follows:

$$e_{ij} = \begin{cases} 2s_{ij} \text{ if } f_i \text{ links to } f_j \text{ and } f_j \text{ links to } f_i \\ s_{ij} \quad \text{if } f_i \text{ links to } f_j \text{ exclusive or } f_j \text{ links to } f_i \end{cases} \tag{5.3}$$
$$i = 1, 2, \cdots, N, \quad j = 1, 2, \cdots, N.$$

If f_i and f_j do not link to each other, we do not build the edge between f_i and f_j. Note that we need to define an undirected graph to adopt the method [24] in the next section meanwhile the proposed methods in Chapters 3 and 4 define directed graphs. Thus, by Eq. (5.3), we define an undirected graph that can preserve information of edge directions. By constructing a graph G, we enable application of a method [24] to extract the hierarchical structure for Web video retrieval as shown later.

5.4 Phase II: Accurate and Efficient Extraction of Hierarchical Structure of Web Video Groups

In Section 5.4.1, we present a method for accurately and efficiently extracting the hierarchical structure. A Web video retrieval method that uses the hierarchical structure is presented in Section 5.4.2.

5.4.1 Extraction of Hierarchical Structure of Web Video Groups

After deriving the link relationships between Web videos, *i.e.*, the graph $G = (V, E)$, we extract the hierarchical structure of Web video groups, *i.e.*, Web video sets with similar topics. Specifically, by using G, the hierarchical structure of Web video groups is extracted via an algorithm based on recursive modularity optimization [24]. The algorithm [24] is a very fast method for graph analysis by which 118 million nodes can be processed in 152 minutes. In addition, the proposed similarities can be easily introduced into this method. Therefore, since this method is optimal for extracting the hierarchical structure accurately and efficiently, we adopt this method. Extraction of the hierarchical structure consists of the following two phases.

In the first phase, each node f_i $(i = 1, 2, \cdots, N)$ is assigned to each different Web video group. For each node f_i, the gain of modularity Q is evaluated when a node f_i is set to a Web video group containing a neighbourhood node f_j. Then f_i is re-assigned to a Web video group for which the positive gain is maximum. Note that modularity Q is an evaluation measure for the division results of Web video groups in the graph, which are defined by the following equation [24]:

$$Q = \frac{1}{2m} \sum_{i=1}^{N} \sum_{j=1}^{N} (e_{ij} - \frac{k_i k_j}{2m}) \delta(i, j), \tag{5.4}$$

where

$$2m = \sum_{i=1}^{N}\sum_{j=1}^{N} e_{ij}, \tag{5.5}$$

$$\tag{5.6}$$

$$k_i = \sum_{j=1}^{N} e_{ij}, \tag{5.7}$$

$$\tag{5.8}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ belong to the same Web video group} \\ 0 & \text{otherwise} \end{cases}. \tag{5.9}$$

Thus, the higher the modularity is, the better division results of Web video groups are. This process is applied to all nodes iteratively and sequentially until no more improvement of the modularity can be obtained.

In the second phase, we construct a new graph whose nodes are the Web video groups obtained in the first phase. Here, the edge weight between the two new nodes is the sum of the edge weight of the original graph found during the first phase. Also, each new node has a self-loop that is derived from the weighted edges of the corresponding original nodes obtained in the first phase. In this chapter, we call a pair of the first and second phases "a pass" and this iteration number is denoted by q $(= 1, 2, \cdots, Q_h; Q_h$ being the number of all passes). Moreover, the passes, *i.e.*, the first phase to find the Web video groups from the new graph and the second phase to construct the newer graph, are iterated until no more improvement of the positive gain of modularity can be obtained. Then we denote the obtained Web video groups by $Com_{n_q}^{q}$ $(n_q = 1, 2, \cdots, T_q; T_q$ being the number of Web video groups where pass is $q)$. Consequently, since we can attain Web video groups with different levels of resolution according to the number of passes, it becomes feasible to extract the hierarchical structure of Web video groups.

Finally, we note the following for the above recursive processes. According to the increase of q, the number of Web videos within the new nodes becomes larger and the number of new nodes decreases in the second phase. Thereby, unlike the previously reported methods that

Algorithm 4 : Accurate and Efficient Extraction of Hierarchical Structure of Web Video Groups

Input: A graph $G = (V, E)$ whose nodes and edges are Web videos f_i $(i = 1, 2, \cdots, N)$ and weighted links, respectively

Output: Web video groups with the hierarchy $Com_{n_q}^q$ $(q = 1, 2, \cdots, Q_h,\ n_q = 1, 2, \cdots, T_q)$

1: Assign each node f_i $(i = 1, 2, \cdots, N)$ to each different Web video group.
2: Set the index of pass as $q \leftarrow 1$.
3: **while** True **do**
4: **while** Improvement of modularity Q of a graph G is obtained **do**
5: **for** each node **do**
6: Evaluate the gain of Q when a node is set to each Web video group containing neighbourhood nodes.
7: Re-assign a node to the Web video group for which the positive gain of Q is maximum.
8: **end for**
9: Calculate Q of G.
10: **end while**
11: Denote the obtained Web video groups by $Com_{n_q}^q$ $(n_q = 1, 2, \cdots, T_q)$.
12: **if** Improvement of Q of G is not obtained **then**
13: Break the while loop.
14: **end if**
15: Build the new graph G with a self-loops whose nodes are the obtained Web video groups, where each edge weight is defined by the sum of the edge weights of the original graph.
16: $q \leftarrow q + 1$
17: **end while**
18: Return Web video groups with the hierarchy $Com_{n_q}^q$ $(q = 1, 2, \cdots, Q_h,\ n_q = 1, 2, \cdots, T_q)$.

need to perform screening of Web videos for handling many Web videos, the proposed method can extract the hierarchical structure for the whole target Web videos by the above efficient recursive phases. Therefore, the proposed method can avoid the performance degradation caused by discarding relevant Web videos in the screening unlike the methods presented in Chapters 3 and 4. For detailed procedures of extraction of the hierarchical structure, refer to Algorithm 4.

5.4.2 Web Video Retrieval Using Hierarchical Structure of Web Video Groups

In the proposed method, it becomes feasible to perform Web video retrieval by using the obtained hierarchical structure of Web video groups. First, we obtain the hierarchically extracted Web video groups $Com_{n_q}^q$ $(n_q = 1, 2, \cdots, T_q,\ q = 1, 2, \cdots, Q_h)$. Then we rank each Web video f_i $(i \in \{1, 2, \cdots, N\})$ that belongs to the Web video group $Com_{n_q}^q$ in the descending order of the following measure $R_{n_q}^q(i)$:

$$R_{n_q}^q(i) = \sum_{j=1}^{N} e_{ji}\delta(i,j), \tag{5.10}$$

$$\delta(i,j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ belong to} \\ & \text{the same Web video group.} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

Thus, Web videos in each Web video group are ranked on the basis of the number of weighted links in the graph of each Web video group, *i.e.,* the degree centrality of the graph. Moreover, we present the Web video groups in the order of $Com_{n_{Q_h}}^{Q_h}, Com_{n_{Q_h-1}}^{Q_h-1}, \cdots, Com_{n_1}^1$, that is, from larger Web video groups to smaller Web video groups. We notice that the larger Web video groups include Web videos with various topics and the smaller Web video groups contain Web videos with similar topics. Then users select the Web video groups associated with the desired Web videos according to the presented hierarchical structure and retrieve the desired Web videos based on the rankings $R_{n_q}^q(i)$. Hence, the proposed method enables users to easily grasp an overview of many Web videos via the hierarchical structure of Web video groups. As a consequence, users can retrieve the desired Web videos even if varied topics are contained in the retrieval results since they cannot input suitable keywords as a query.

Table 5.1: Detailed conditions of each dataset: K, U and N_{clus} are the numbers of dimensions of keywords used for computing textual feature vectors, dimensions after random projection, and cluster centers in k-means clustering, respectively.

	Query keyword	Num. of Web videos in datasets	Num. of shots in datasets	K	U	N_{clus}
Dataset 1	sightseeing	3001	10249	10832	1000	1000
Dataset 2	game	3003	9644	10255	1000	1000

5.5 Experimental Results

In this section, we present experimental results for actual Web videos to verify the accuracy and computational cost of the proposed method.

5.5.1 Datasets

Datasets, which were same as ones used in the experiments shown in Chapters 3 and 4, were prepared in this section. Table 5.1 shows the details of the datasets.

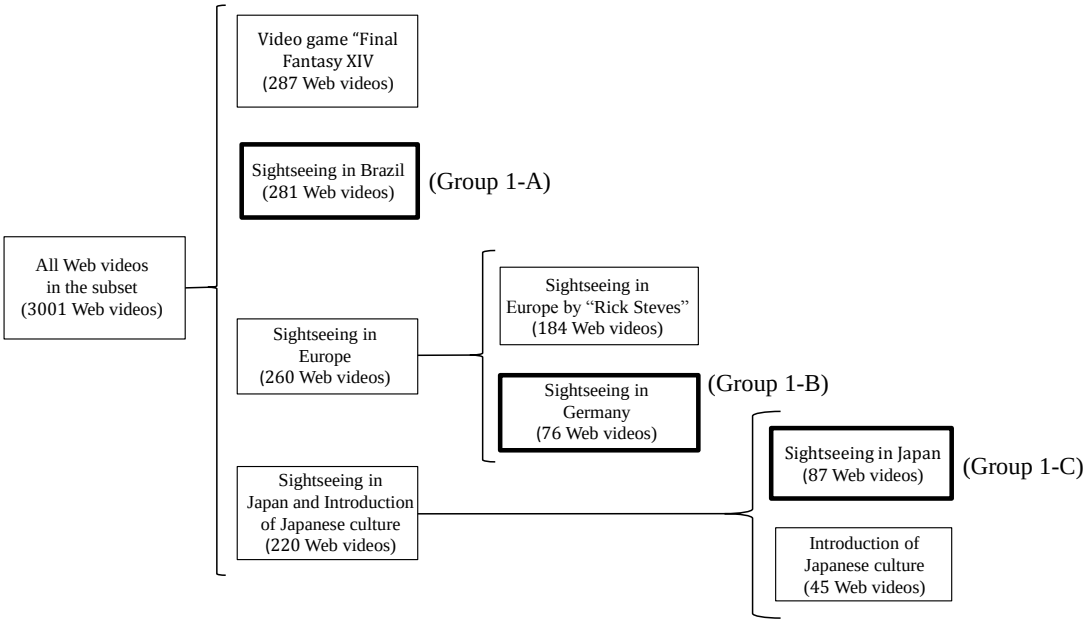
5.5.2 Evaluations

In Figs. 5.1 and 5.2, we show the hierarchical structure of Web video groups obtained by the proposed method. Note that we named each Web video group by checking contents of them manually since the proposition in this thesis is to extract the hierarchical structure and does not include the naming scheme. From this figure, we can see that the hierarchical structure enables users to easily grasp the overview of many Web videos since Web videos containing varied topics are clustered into ones with similar topics.

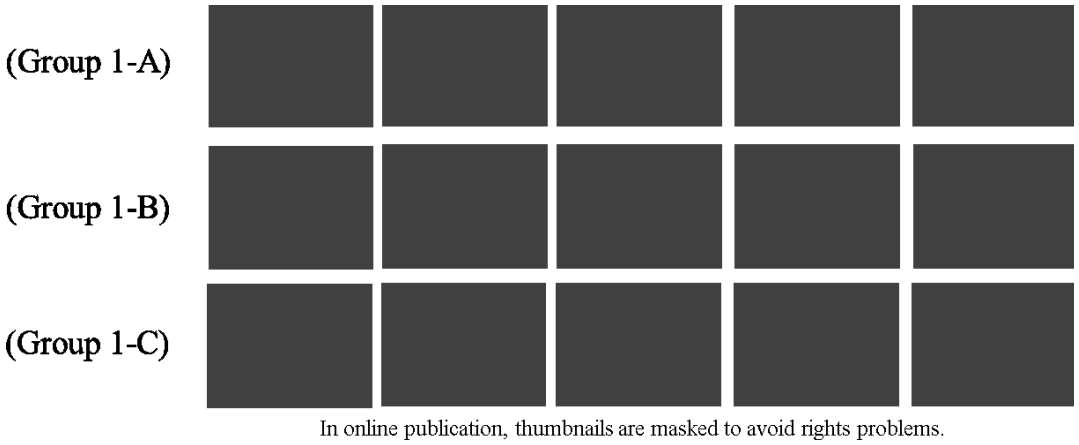
Next, we quantitatively evaluate the accuracy of the proposed method. In this experiment, we denote the proposed method by **(P)**. We compared (P) with the following reference methods.

(R1)

This is a method to extract the hierarchical structure of Web video groups in the same manner as the proposed method, but this method does not use similarities between Web videos. Specifically, we defined edge weights e_{ij} between Web videos f_i and f_j as



(a) Hierarchical structure of Web video groups. Web video groups shown in the second, third and fourth columns correspond to ones where $q = 3$, $q = 2$ and $q = 1$, respectively.

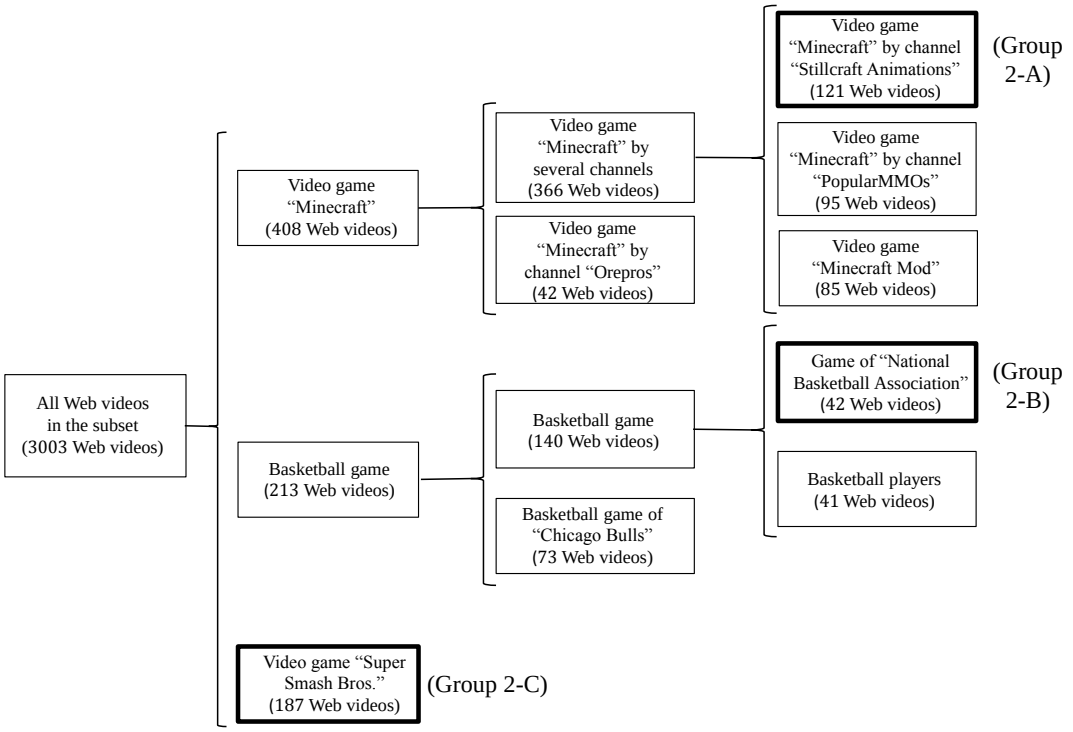


(b) Thumbnails of the retrieved Web videos when (Group 1-A), (Group 1-B) and (Group 1-C) were respectively selected. The top five ranked Web videos were located in the order of left to right

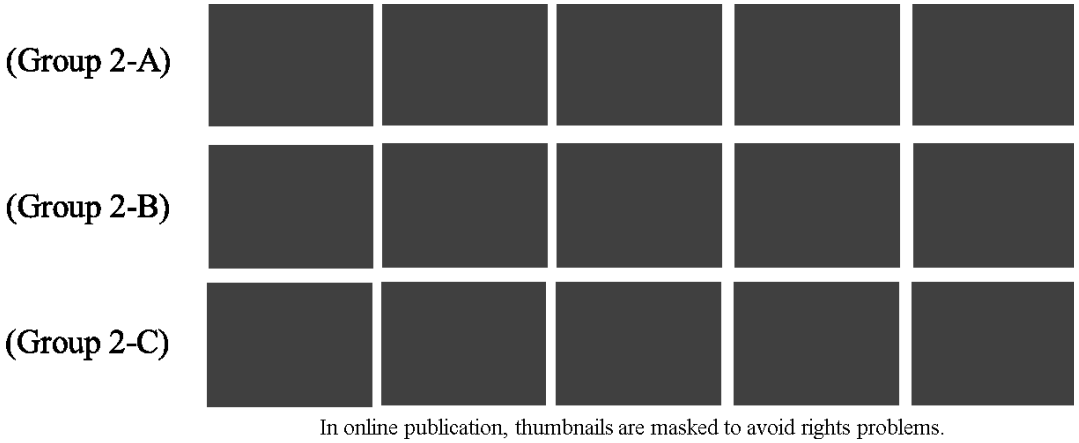
Figure 5.1: Hierarchical structure of Web video groups for subset 1, which were obtained by the proposed method.

follows:

$$e_{ij} = \begin{cases} 1 & \text{if } f_i \text{ links to } f_j \text{ or } f_j \text{ links to } f_i \\ 0 & \text{otherwise} \end{cases}$$
$$i = 1, 2, \cdots, N, \quad j = 1, 2, \cdots, N.$$



(a) Hierarchical structure of Web video groups. Web video groups shown in the second, third and fourth columns correspond to ones where $q = 3$, $q = 2$ and $q = 1$, respectively.



(b) Thumbnails of the retrieved Web videos when (Group 2-A), (Group 2-B) and (Group 2-C) were respectively selected. The top five ranked Web videos were located in the order of left to right

Figure 5.2: Hierarchical structure of Web video groups for subset 2, which were obtained by the proposed method.

(R2-1)

This is the proposed method in Chapter 3 to extract the hierarchical structure of Web

video groups. This method performs screening of Web videos to select about 1000 Web videos in a dataset.

(R2-2)

This is the same method as (R2-1). However, this method performs screening of Web videos to select about 1500 Web videos.

(R3-1)

This is the proposed method in Chapter 4 to efficiently extract the hierarchical structure of Web video groups. This method performs calculation of sub-sampled CCA-based link relationships between Web videos as in the proposed method in this chapter, but screening of Web videos is performed to select about 1000 Web videos in a dataset.

(R3-2)

This is the same method as (R3-1). However, this method performs screening of Web videos to select about 1500 Web videos.

(R4)

This is a conventional method [13] that extracts Web video groups via Web video features and metadata "related videos"; however, this method does not extract the hierarchical structure.

(R5)

This is a method based on a work [12] with a Web video group extraction scheme by affinity propagation [29]. This method uses only Web video features without the use of metadata "related videos", and the hierarchical structure is not explored. Note that we do not use the original similarities in a paper [12] but use the proposed similarities defined in Eq. (5.1) in this experiment.

(R6)

This is a method based on clustering via latent semantic indexing (LSI) that uses only textual features of Web videos [14], which does not present the hierarchical structure to

Table 5.2: Number of Web videos after screening of Web videos by (R2-1), (R2-2), (R3-1) and (R3-2). Table 4.2 in Chapter 4 shows the details of the parameter settings.

	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Dataset 1	1038	1584	1021	1410
Dataset 2	1079	1509	1033	1561

users. Note that an original method in [14] utilizes information of YouTube Playlists. In this experiment, however, we used a video-keyword vectors whose elements are 1 if each keyword appears in title and description of the Web video and 0 otherwise. Moreover, the number of dimensions of the video-keyword vectors after LSI were set to 200 as in a paper [14]. Each Web video is ranked on the basis of the attribution degree to the belonging cluster.

Table 5.2 shows the number of Web videos after screening of Web videos by (R2-1), (R2-2), (R3-1) and (R3-2).

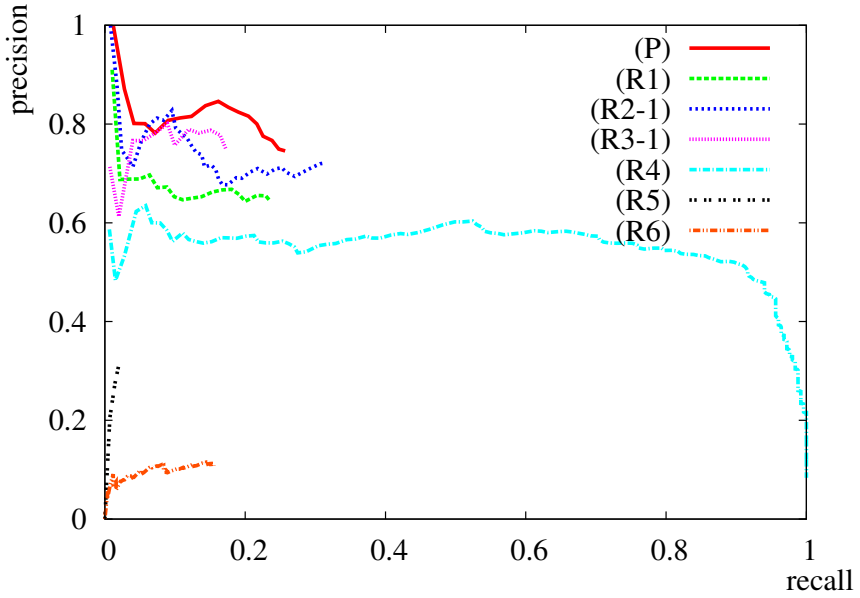
For the evaluation, we used recall, precision and average precision (AP@k) defined as follows:

$$\text{Recall} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of relevant Web videos}}, \tag{5.12}$$

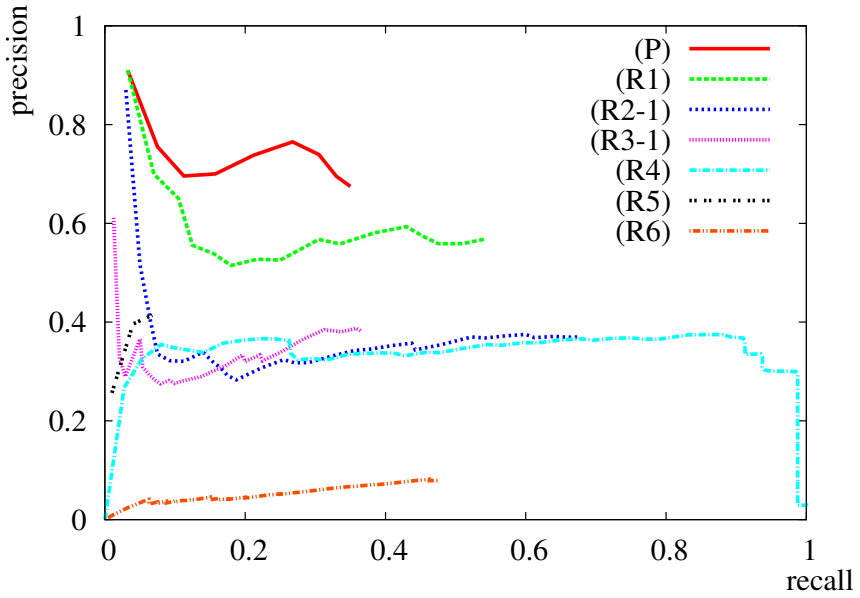
$$\text{Precision} = \frac{\text{Num. of correctly retrieved Web videos}}{\text{Num. of retrieved Web videos}}, \tag{5.13}$$

$$\text{AP@}k = \frac{1}{R_k}\sum_{i=1}^{k} x_i \, prec_i, \tag{5.14}$$

where k is the number of Web videos provided as the retrieval results, R_k is the number of "relevant Web videos" within k Web videos of the retrieval results, x_i is 1 if the i-th retrieved Web videos are "relevant Web videos" and 0 otherwise, and $prec_i$ is the precision when i Web videos are retrieved. Below, we verify the accuracy of extracting the hierarchical structure for Web video retrieval. First, for all methods, we selected a Web video group that included relevant Web videos the most, and Web videos were retrieved according to the rankings of Web videos. Then recall and precision were calculated and Figs. 5.3 and 5.4 show precision-recall

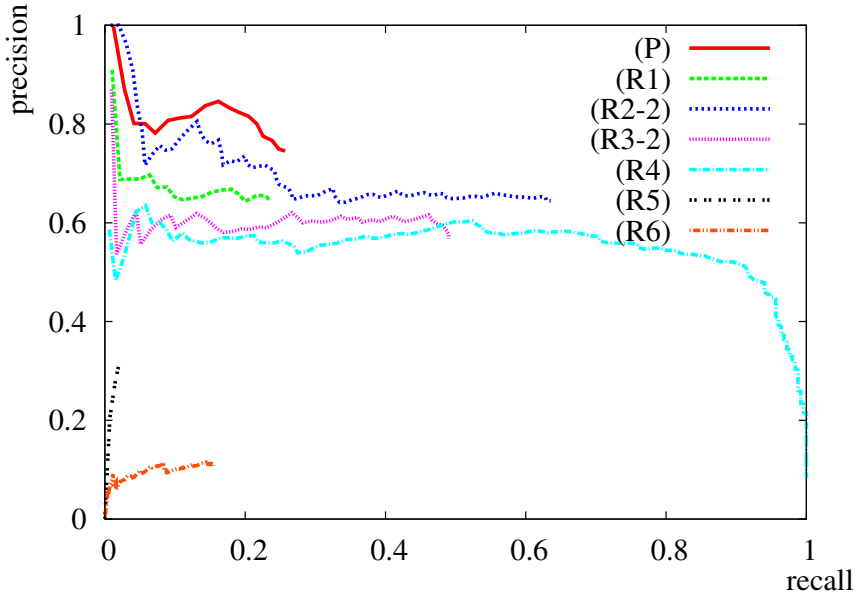


(a) Result for dataset 1. Retrieval results when (Group 1-C) was selected were evaluated.

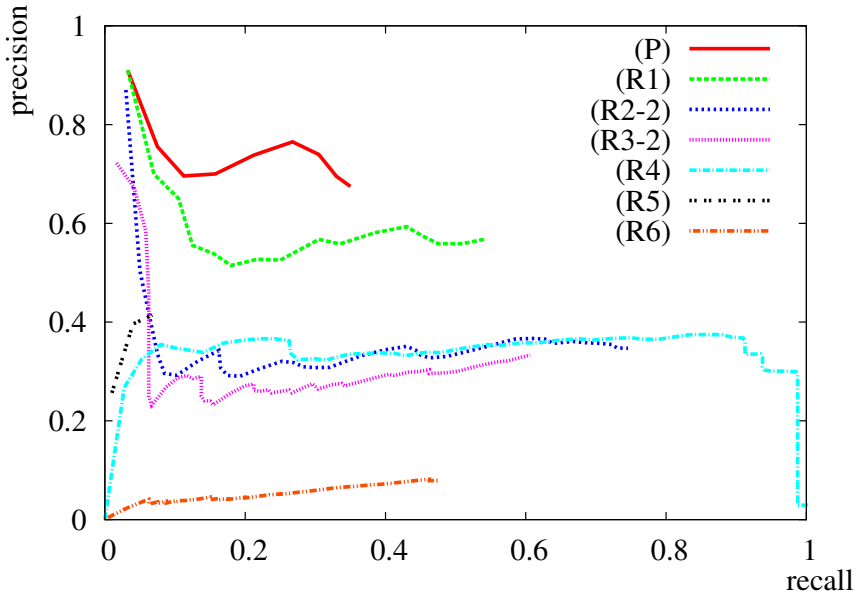


(b) Result for dataset 2. Retrieval results when (Group 2-B) was selected were evaluated.

Figure 5.3: Precision-recall curves for (P), (R1), (R2-1), (R3-1), (R4), (R5) and (R6): (P) and (R1) respectively correspond to the retrieval results where $q = 1$. (R2-1) and (R3-1) respectively correspond to the retrieval results where N_{cut} were N_{cut}^{opt}. Note that (R4) has higher recall than other methods since only this method is based on soft clustering and all Web videos can belong to every Web video group.



(a) Result for dataset 1. Retrieval results when (Group 1-C) was selected were evaluated.



(b) Result for dataset 2. Retrieval results when (Group 2-B) was selected were evaluated.

Figure 5.4: Precision-recall curves for (P), (R1), (R2-2), (R3-2), (R4), (R5) and (R6): (P) and (R1) respectively correspond to the retrieval results where $q = 1$. (R2-2) and (R3-2) respectively correspond to the retrieval results where N_{cut} were N_{cut}^{opt}. Note that (R4) has higher recall than other methods since only this method is based on soft clustering and all Web videos can belong to every Web video group.

Table 5.3: Ground truths used for drawing Figs. 5.3 and 5.4.

	Relevant Web videos
Dataset 1	Web videos with Freebase topics about "Sightseeing in Japan"
Dataset 2	Web videos with Freebase topics about "Game of National Basketball Association"

Table 5.4: AP@k weighted by the numbers of Web videos in each Web video group. We show the results where $q = 1$ for (P) and (R1). For (R2-1), (R2-2), (R3-1) and (R3-2), the results where N_{cut} were N_{cut}^{opt} are shown. For (R4) based on soft clustering, ideal values are shown since we selected Freebase topics that provided the highest AP@k of all as ground truths. Also, k was defined as the number of Web videos in each Web video group.

	Proposed method	Method without the use of Web video features	Method proposed in Chapter 3		Method proposed in Chapter 4		Reference method [13]	Reference method [12]	Reference method [14]
	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)	(R4)	(R5)	(R6)
Data-set 1	**0.805**	0.758	0.765	0.636	0.530	0.585	0.632	0.496	0.320
Data-set 2	**0.876**	0.833	0.851	0.841	0.822	0.726	0.563	0.640	0.373
Aver-age	**0.840**	0.796	0.774		0.666		0.598	0.568	0.347

curves for the datasets. Table 5.3 shows ground truths used for drawing these figures. Here, we defined ground truths as in the experiments shown in Chapters 3 and 4. Also, Table 5.4 shows the averages of AP@k. Here, we selected the most frequent Freebase topics within each Web video group as ground truths and calculated the averages, which were weighted by the numbers of Web videos in Web video groups. When comparing (P) with (R1) and (R6), we can see the effectiveness of the proposed CCA-based link relationships via visual, audio and textual features. Also, since (P) does not need to perform screening of Web videos by our efficient recursive processing unlike (R2-1), (R2-2), (R3-1) and (R3-2), evaluation results obtained by (P) can outperform those obtained by these reference methods. When comparing (P) with (R4), (R5) and (R6), we can confirm that the use of the hierarchical structure enables accurate retrieval.

Next, we show computational times for obtaining the hierarchical structure of Web video groups in Table 5.5. From this table, the computational efficiency of (P) can be confirmed.

In particular, it is significant to realize the results although (P) does not perform screening of Web videos unlike (R2-1), (R2-2), (R3-1) and (R3-2). Thus, we can show the effectiveness of introducing the method [24] into Web video retrieval. Specifically, although the method [24] has not been proposed for Web video retrieval originally, we can confirm that introduction of the method [24] successfully enables accurate and efficient extraction of the hierarchical structure for Web video retrieval. Meanwhile, it should be noted that computational time by (R1) is shorter than that by (P). However, the retrieval accuracy of (P) is higher than that of (R1) as shown in Figs. 5.3 and 5.4 and Table 5.4. Thus, the superiority of (P) can be confirmed.

5.6 Conclusions

In this chapter, a method for accurately and efficiently extracting the hierarchical structure of Web video groups has been proposed. The proposed method enabled efficient calculation of latent features of Web videos via sub-sampled CCA as in the method in Chapter 4. Furthermore, the proposed method introduced the graph analysis algorithm that can accurately and efficiently extract the hierarchical structure by focusing only on local structure of the graph. Thus, screening of Web videos, which is necessary for the methods in Chapters 3 and 4, becomes unnecessary. As a result, we can avoid performance degradation caused by discarding relevant Web videos in the screening. Experimental results for actual Web videos have confirmed that the proposed method enables extraction of the hierarchical structure with high accuracy as well as low computational cost to retrieve desired Web videos even if users cannot input suitable keywords as a query.

Table 5.5: Computational time for obtaining the hierarchical structure of Web video groups. We used a computer with Intel® CPU 2.50GHz and 32GB RAM for (A). For (B), (C), (D) and (E), Intel® CPU 3.50GHz and 16GB RAM is used. Units of each element in this table are the second.

(A) Web video feature extraction. (The image size was 320 × 180 pixels and the frame rate was 25 fps. The audio signal was sampled at 22.05 kHz.)

	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Visual feature	54.570	—	54.570	54.570	54.570	54.570
Audio feature	16.940	—	16.940	16.940	16.940	16.940
Textual feature[1]	0.020000	—	0.020000	0.020000	0.020000	0.020000

(B) Derivation of CCA-based link relationships between Web videos (see Eq. (5.3)). Note that (P) and (R3-1), (R3-2) include k-means clustering but (R2-1) and (R2-2) do not.

	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Dataset 1	490.76	—	404.18	404.18	490.73	490.73
Dataset 2	432.82	—	418.83	418.83	432.79	432.79

(C) Screening of Web videos for extracting the hierarchical structure of Web video groups. Note that (R2-1), (R2-2), (R3-1) and (R3-2) need to perform this processing. The upper values in this table show the results without parallel computation and [ ] denote ones with eight threads parallel computation (see Table 4.6 for the details).

	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Dataset 1	—	—	0.94974	1.1769	1581.1 [351.60]	1645.8 [335.71]
Dataset 2	—	—	0.94616	1.7133	1492.1 [348.03]	1495.8 [338.92]

(D) Extraction of the hierarchical structure of Web video groups.

	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Dataset 1	0.66878	0.54224	325.31	2935.4	0.12250	1.4461
Dataset 2	0.69547	0.46041	802.07	2821.5	0.024504	0.549015

(E) Total time for obtaining the hierarchical structure except for Web video feature extraction, sum of (B), (C) and (D): the upper values show the results without parallel computation and [ ] denote ones with eight threads parallel computation.

	(P)	(R1)	(R2-1)	(R2-2)	(R3-1)	(R3-2)
Dataset 1	491.43	0.54224	730.44	3340.8	2071.9 [842.46]	2138.0 [827.88]
Dataset 2	433.52	0.46041	1221.9	3242.0	1924.9 [780.84]	1929.2 [772.25]

Chapter 6

Conclusions

As conclusions of this thesis, this chapter reviews the overview of the proposition and shows future directions.

6.1 Overview of the Proposition in this Thesis

In this section, the background of this study and overview of the proposition in this thesis are reviewed.

With the widespread use of video hosting services, more and more users are retrieving Web videos to access desired contents. Therefore, it is necessary to develop a scheme for effectively retrieving desired Web videos. However, there is a limitation that existing search engines require users to input suitable keywords as a query to accurately retrieve the desired Web videos. To overcome this limitation, this thesis has proposed a method for extracting the hierarchical structure of Web video groups on the basis of the relevance between heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos. By utilizing the relevance between heterogeneous features, advanced Web video retrieval becomes feasible. Below, the overview of each chapter of this thesis is reviewed.

In Chapter 2, related work of Web video retrieval was presented and problems to be solved in this thesis were clarified. Chapter 3 presented a method to extract the hierarchical structure of Web video groups by utilizing the relevance between heterogeneous features. In particular, the proposed method constructed a graph, whose edges represent similarities between latent features of Web videos, on the basis of visual, audio and textual features and features

of link relationships between Web videos. By analyzing the graph on the basis of SCCs, edge betweenness and modularity, sub-graphs and their inclusion relationships became clear; therefore, the hierarchical structure of Web video groups can be extracted. In Chapter 4, a method that enabled acceleration of the method presented in Chapter 3 was proposed. Specifically, the proposed method derived efficient CCA, named sub-sampled CCA, to calculate latent features of Web videos efficiently. Furthermore, efficient extraction of the hierarchical structure was realized by constructing a group graph that can handle many Web videos by a small number of nodes. Chapter 5 presented a method that enabled improvement of the accuracy and efficiency by the method presented in Chapter 4. In particular, introduction of a graph analysis algorithm, which focuses only on local structure of a graph that represents similarities between latent features of Web videos, enabled significant acceleration of extraction of the hierarchical structure. The methods presented in Chapters 3 and 4 needed to perform screening of Web videos when they targeted many Web videos. On the other hand, since the proposed method made the screening unnecessary, performance degradation by discarding relevant Web videos in the screening can be avoided.

Consequently, for realizing advanced Web video retrieval, this thesis has proposed a method for extracting the hierarchical structure of Web video groups by utilizing relevance between heterogeneous features. The contributions of this thesis are as follows.

(Contribution-i)

The hierarchical structure of Web video groups is successfully extracted by utilizing relevance between heterogeneous features, *i.e.*, visual, audio and textual features and features of link relationships between Web videos.

(Contribution-ii)

Even if users cannot input suitable keywords as a query, accurate retrieval of the desired Web videos is realized by presenting the obtained hierarchical structure to users.

Experimental results for actual Web videos, which were presented in Chapters 3, 4 and 5, have verified the effectiveness of the proposed method.

6.2 Future Directions

This section describes future directions of this study. In this thesis, we named each Web video group in the hierarchies shown in the experiments by checking contents in Web video groups manually. In the future, a scheme for automatically naming the hierarchies should be developed. By revealing salient keywords in each Web video group and estimating hierarchical relationships between the keywords [30, 39], automatic naming of the hierarchies will be realized.

Moreover, for a real-world deployment, a user interface for Web video retrieval should be developed. I consider that the following approach will be useful. First, for a given query, the interface presents the hierarchical structure of Web video groups to a user. When a user selects a Web video group, the interface presents the details of the selected Web videos group to a user. To realize this, a graph drawing algorithm [40] will be useful for grasping the details of the selected Web video groups at a glance. Then, until desired Web videos are retrieved, a user repeatedly selects Web video groups in several hierarchies by browsing names of each Web video group. Future work includes the development of such a user interface for a real-world deployment. Also, usability of the interface should be evaluated by subjective experiments in the future.

Bibliography

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," in *IDC iView*, 2012.

[2] V. Turner and J. Gantz, "The digital universe of opportunities: Rich data and the increasing value of the internet of things," in *IDC iView*, 2014.

[3] M. Haseyama, T. Ogawa, and N. Yagi, "A review of video retrieval based on image and video semantic understanding," *ITE Trans. Media Technology and Applications*, vol. 1, no. 1, pp. 2–9, 2013.

[4] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 6, pp. 797–819, 2011.

[5] P. Greetha and V. Narayanan, "A survey of content-based video retrieval," *Journal of Computer Science*, vol. 4, no. 6, pp. 474–486, 2008.

[6] I-H. Jhuo and D. T. Lee, "Video event detection via multi-modality deep learning," in *Proc. IEEE Int. Conf. Pattern Recognition*, 2014, pp. 666–671.

[7] P. Natarajan, S. Wu, S. Vitaladevuni., X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan, "Multimodal feature fusion for robust event detection in web videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 1298–1305.

[8] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann, "Complex event detection via multi-source video attributes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 2627–2633.

[9] X-J. Chen, Y-Z. Zhan, J. Ke, and X-B. Chen, "Complex video event detection via pairwise fusion of trajectory and multi-label hypergraphs," *Multimedia Tools and Applications*, pp. 1–22, 2015.

[10] Y. Wang, M. Belkhatir, and B. Tahayna, "Near-duplicate video retrieval based on clustering by multiple sequence alignment," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 941–944.

[11] U. Gargi, L. Wenjun, M. Vahab, and Y. Sangho, "Large-scale community detection on youtube for topic discovery and exploration," in *Proc. Int. AAAI Conf. Weblogs and Social Media*, 2011, pp. 486–489.

[12] A. Hindle, J. Shao, D. Lin, J. Lu, and R. Zhang, "Clustering web video search results based on integration of multiple features," *World Wide Web*, vol. 14, no. 1, pp. 53–73, 2011.

[13] Y. Hatakeyama, T. Ogawa, S. Asamizu, and M. Haseyama, "A novel video retrieval method based on web community extraction using features of video materials," *IEICE Trans. Fundamentals*, vol. E92-A, no. 8, pp. 1961–1969, 2009.

[14] M. Kamie, T. Hashimoto, and H. Kitagawa, "Effective web video clustering using playlist information," in *Proc. ACM Symp. Applied Computing*, 2012, pp. 949–956.

[15] J. Sang and C. Xu, "Browse by chunks: Topic mining and organizing on web-scale social media," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7S, no. 1, pp. 30:1–30:18, 2011.

[16] C-W Ngo, T-C Pong, and H-J Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 446–458, 2002.

[17] C-W Ngo, T-C Pong, and H-J Zhang, "On clustering and retrieval of video shots," in *Proc. ACM Int. Conf. Multimedia*, 2001, pp. 51–60.

[18] C. Taskiran, J.-Y. Chen, A. Albiol, L. Torres, C. A. Bouman, and E. J. Delp, "Vibe: A compressed video database structured for active browsing and search," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 103–118, 2004.

[19] A. Nielsen, "Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data," *IEEE Trans. Image Processing*, vol. 11, no. 3, pp. 293–305, 2002.

[20] R. A. Butafogo and B. Schneiderman, "Identifying aggregates in hypertext structures," in *Proc. 3rd ACM Conf. Hypertext*, 1991, pp. 63–74.

[21] A. Arenas, J. Duch, A. Fernandez, and S. Gomez, "Size reduction of complex networks preserving modularity," *New J. Phys.*, vol. 9 (176), pp. 604–632, 2007.

[22] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69 (026113), 2004.

[23] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967, pp. 281–297.

[24] V. D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, p. P10008, 2008.

[25] A. W. M. Smeulders, M. Worring, S. Santini., A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[26] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[27] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD*, 2008, pp. 1247–1250.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[29] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[30] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[31] A. Nagasaka and Y. Tanaka, "Automatic video indexing and fullvideo search for object appearances," in *Proc. IFIP 2nd Working Conf. Visual Database Systems*, 1991, pp. 113–127.

[32] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," in *Proc. IEEE*, 1990, pp. 678–689.

[33] N. Nitanda and M. Haseyama, "Audio-based shot classification for audiovisual indexing using pca, mgd and fuzzy algorithm," *IEICE Trans. Fundamentals*, vol. E90-A, no. 8, pp. 1542–1548, 2007.

[34] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[35] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. IEEE Int. Workshop on Quality of Service*, 2008, pp. 229–238.

[36] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Jounal of ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[37] D. Achlioptas, "Database-friendly random projections," in *Proc. ACM Symp. Principles of Database Systems*, 2001, pp. 274–281.

[38] T. T. Tanimoto, "An elementary mathematical theory of classification and prediction," *IBM Internal Report*, 1958.

[39] R.J. Brachman, "What is-a is and isn't: An analysis of taxonomic links in semantic networks," *IEEE Computer*, vol. 16, no. 10, pp. 30–36, 1983.

[40] Y. Hatakeyama and M. Haseyama, "An effective visualization method based on web community extraction using hyperlink between web videos and its application to retrieval," in *Proc. Int. Tech. Conf. Circuits/Systems, Computers and Communications*, 2010, pp. 371–374.

Achievements of the Author

Papers

(A) Journal Papers

[**A-1**] R. Harakawa, T. Ogawa, and M. Haseyama, "An efficient extraction method of hierarchical structure of Web communities for Web video retrieval," *ITE Trans. Media Technology and Applications*, vol. 2, no. 3, pp. 287–297, 2014.

[**A-2**] R. Harakawa, T. Ogawa, and M. Haseyama, "A Web video retrieval method using hierarchical structure of Web video groups," *Multimedia Tools and Applications*, DOI:10.1007/s11042-015-2976-8, 2015.

[**A-3**] R. Harakawa, T. Ogawa, and M. Haseyama, "Accurate and efficient extraction of hierarchical structure of Web communities for Web video retrieval," *ITE Trans. Media Technology and Applications*, vol. 4, no. 1, pp. 49–59, 2016.

(B) International Conferences

[**B-1**] R. Harakawa, Y. Hatakeyama, T. Ogawa, and M. Haseyama, "Web community extraction with weighting three kinds of features for Web video retrieval," in *Proc. Int. Tech. Conf. Circuits/Systems, Computers and Communications*, 2012, P-M2-20.

[**B-2**] R. Harakawa, Y. Hatakeyama, T. Ogawa, and M. Haseyama, "An extraction method of hierarchical Web communities for Web video retrieval," in *IEEE Int. Conf. Image Processing*, 2013, pp. 4397–4401.

[**B-3**] R. Harakawa, T. Ogawa, and M. Haseyama, "Exhibition method of hierarchical structure of Web communities using community density for Web video retrieval," in *IEEE Int. Symp. Consumer Electronics*, 2014, pp. 194–195.

[**B-4**] R. Harakawa, T. Ogawa, and M. Haseyama, "Hierarchical structure extraction of Web communities based on new similarity estimation from heterogeneous video features," in *Proc. Int. Workshop on Advanced Image Technology*, 2015, 314.

[**B-5**] R. Harakawa, T. Ogawa, and M. Haseyama, "Extraction of hierarchical structure of Web communities including salient keyword estimation for Web video retrieval," in *Proc. IEEE Int. Conf. Image Processing*, 2015, ARS-P18.2.

[**B-6**] R. Harakawa, T. Ogawa, and M. Haseyama, "Extraction of hierarchical structure of Web video groups via adaptive feature selection," in *Proc. Int. Workshop on Advanced Image Technology*, 2016, P.1C-7.

[**B-7**] D. Takehara, R. Harakawa, T. Ogawa, and M. Haseyama, "Organizing Web video collections into optimal category hierarchies for Web video retrieval," in *Proc. Int. Workshop on Advanced Image Technology*, 2016, P.1C-6.

(C) **Domestic Conferences**

[**C-1**] 原川 良介, 畠山 泰貴, 小川 貴弘, 長谷山 美紀, "映像の特徴を用いた Web コミュニティ抽出の高精度化に関する検討 ―最短リンク経路長の導入による試み―", 電気・情報関係学会北海道支部連合大会 講演論文集, p. 150, 2012.

[**C-2**] 原川 良介, 小川 貴弘, 長谷山 美紀, "Web 映像検索のための Web コミュニティの階層構造抽出の高精度化に関する検討", 第 28 回信号処理シンポジウム, vol. P1-17, pp. 125–130, 2013.

[**C-3**] 原川 良介, 小川 貴弘, 長谷山 美紀, "Web 映像コミュニティの階層構造抽出の大規模データ適用に関する一検討", 映像情報メディア学会技術報告, vol. 38, no. 7, pp. 275–280, 2014.

[**C-4**] 竹原 大智, 原川 良介, 小川 貴弘, 長谷山 美紀, “Web 映像検索のための意味内容を考慮した Web コミュニティの階層構造抽出に関するー検討”, 電気・情報関係学会北海道支部連合大会 講演論文集, p. 164, 2014.

[**C-5**] 原川 良介, 小川 貴弘, 長谷山 美紀, “Web コミュニティの階層構造抽出に関する検討 ー映像特徴ベクトル間の距離に注目した類似度の導入による高精度化ー”, 第 29 回信号処理シンポジウム, vol. P6-2, pp. 515–520, 2014.

[**C-6**] 原川 良介, 小川 貴弘, 長谷山 美紀, “Web 映像検索のための Web コミュニティの階層構造提示法に関する一検討 ー Web コミュニティを代表するキーワード抽出の試みー”, 映像情報メディア学会技術報告 vol. 39, no. 7, pp. 89–94, 2015.

[**C-7**] R. Harakawa, T. Ogawa, and M. Haseyama, “User interest estimation using heterogeneous features and its application to YouTube user recommendation,” 第 18 回 画像の認識・理解シンポジウム, SS4-34, 2015.

[**C-8**] D. Takehara, R. Harakawa, T. Ogawa, and M. Haseyama, “Organizing Web video collections into optimal category hierarchies via multimodal features for Web video retrieval,” 第 18 回 画像の認識・理解シンポジウム, pp. SS1-26, 2015.

[**C-9**] 竹原 大智, 原川 良介, 小川 貴弘, 長谷山 美紀, “Web 映像検索のための Web コミュニティの提示法に関する検討 ー Web コミュニティに含まれるトピックを考慮した代表キーフレーズ抽出の試みー,” 電気・情報関係学会北海道支部連合大会講演論文集, p. 122, 2015.

(D) **Awards**

[**D-1**] 北海道大学新渡戸賞 (2010 年 6 月)

[**D-2**] 北海道大学工学部 William Wheeler Prize (2013 年 3 月)

[**D-3**] The 2014 IEEE Sapporo Section Encouragement Award (2015 年 2 月)

[**D-4**] 平成 26 年度電子情報通信学会北海道支部学生員奨励賞 (2015 年 3 月)