Kobe University at TRECVID 2009 Search Task

Kimiaki Shirahama
Graduate School of Economics, Kobe University
2-1, Rokkodai, Nada, Kobe, Japan
shirahama@econ.kobe-u.ac.jp

Chieri Sugihara, Yuta Matsuoka, Kana Matsumura, Kuniaki Uehara
Graduate School of Engineering, Kobe University
1-1, Rokkodai, Nada, Kobe, Japan
{chieri,matuoka,matsumura}@ai.cs.scitec.kobe-u.ac.jp, uehara@kobe-u.ac.jp

Abstract

In TRECVID 2009 search task, we have developed a method which defines any interesting topic from examples provided by a user, especially, positive and negative examples. Specifically, considering a large variation of features in a topic, we use "rough set theory" which defines the topic as a union of subsets. In each subset, some positive examples can be correctly distinguished from all negative examples. Based on such subsets, we can collectively retrieve shots which show the same topic but contain significantly different features.

For our method, it is crucial what kind of examples are used. To examine the influence of examples on the retrieval performance, we submitted the following three runs:
1. M_A_N_cs24_kobe1_1: In this run, a user manually selects positive and negative examples for each topic.
2. M_A_N_cs24_kobe2_2: It is difficult for the user to select effective negative examples for defining a topic, since a huge number of shots can be negative examples. So, in this run, we use "partially supervised learning" which defines the topic only from positive examples, by selecting negative examples from unlabeled examples (i.e. shots except for positive examples).
3. I_A_N_cs24_kobeS_3: In this supplemental run, from the result of M_A_N_cs24_kobe1_1, the user selects additional positive and negative examples. Note that due to the slow search speed, this run violates the maximum time limit.

From evaluation results, we find that our non-interactive methods M_A_N_cs24_kobe1_1 and M_A_N_cs24_kobe2_2 can achieve comparable performances to medians of interactive runs. Also, I_A_N_cs24_kobeS_3 indicates that the performance of our method can be significantly improved by using a large number of examples.

1. Introduction

This year we have participated in TRECVID 2009 search task, and submitted three runs which include two manually-assisted runs and one interactive run. In this paper, we present our topic retrieval method and examine its performance based on evaluation results of the above three runs.

Since users are interested in a great variety of topics, it is impossible to prepare models for retrieving all topics [13, 3]. Also, it is impossible to pre-define concepts which are need for representing all topics [11, 15]. In this paper, we introduce a method which defines a topic from examples provided by a user. Thus, our method can retrieve any topic as long as the user can provide examples. Particularly, in order to find differences between relevant shots and irrelevant shots to a topic, we use both "positive examples" where the topic is shown and "negative examples" where it is not shown.

In videos, the same topic can be taken by different camera techniques and in different situations. So, shots of the topic contain significantly different features. Regarding this, we use "rough set theory" which defines the topic as a union of subsets, where some positive examples can be correctly discriminated from all negative examples. Based on such subsets, we can collectively retrieve shots which show the same topic but contain significantly different features. Thus, our main objective in TRECVID 2009 is to examine the effectiveness of rough set theory for covering a large variation of features in the same topic.

The performance of our method depends on positive and negative examples. Note that a set of negative examples is just the complement of a set of positive examples. So, for a topic, a huge number of shots can be negative examples. But, from these shots, it is difficult for a user to select negative examples effective for defining the topic.

Hence, we use "partially supervised learning" which defines a topic only from positive examples, by selecting negative examples from unlabeled examples, that is, shots except for positive examples. So, our another important objective in TRECVID 2009 is to examine the effectiveness of negative examples selected by partially supervised learning.

2. Features

In our topic retrieval method, we extract the following five types of features from each shot:

Color distribution: This feature is used to characterize the color of an object in a certain region. For example, in a shot where the sky is shown, many blue-colored pixels are extracted from the upper part. Also, in a shot where a person talks to the camera, many skin-colored pixels are extracted from the central part. Based on the above observation, we partition the keyframe of each shot into 30 regions as shown in Fig. 1. And, from each region, we extract a 36-bins color histogram introduced by Zhang et al. [10].

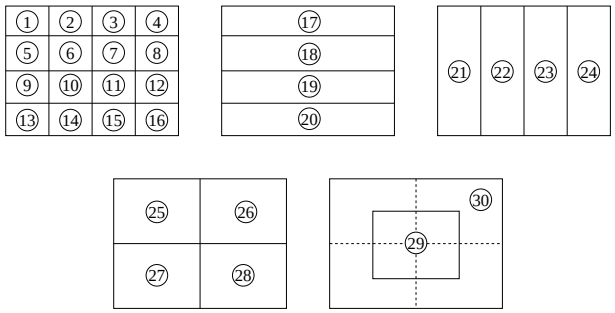

Figure 1. Partition of a keyframe into 30 regions.

Edge distribution: This feature is used to characterize the texture of an object in a certain region. For example, in a shot where a grass is shown, many edges with arbitrary directions are extracted from the bottom part. In addition, in a shot where a city street is shown, corresponding to buildings, many edges with horizontal and vertical directions are extracted from the right and left sides. So, from each region in Fig. 1, we extract a 5-bins histogram where one bin represents the frequency of a certain direction of edges, that is, vertical, horizontal, 45-degree diagonal, 135-degree diagonal or non-directional edges [12].

Visual word distribution: This feature is used to characterize patches (local shapes) of an object in a certain region. For example, in a shot where a car is mainly shown, visual words corresponding to the front window, headlight, number plate etc., are extracted from the central part. So, from each region in Fig. 1, we extract a visual word distribution as a 500-bins histogram where one bin represents the frequency of a visual word. In order to construct a set of visual words, we firstly extract SIFT descriptors which represent local gradient orientations around keypoints, obtained by Harris-Laplace keypoint detector [5]. Then, we group SIFT descriptors into 500 clusters by using k-means clustering. As a result, each cluster corresponds to a visual word.

Number of faces: This feature is used to characterize the number of faces with a certain size. For example, in a shot where a person talks to the camera, one face is shown in a large region. Also, in a shot where three persons talk to each other, three faces are shown in small regions. To extract such face features, we firstly detect faces in the keyframe by using Viola's face detection method [14]. Then, by using thresholds, we classify each detected face into large-size, middle-size or small-size. And, for each size, we count the number of faces.

Moving regions: This feature is used characterize object movements and camera works in a shot. For example, in a shot where a person walks to the left, visual words extracted from this person move to the left. Also, when the camera moves to the right, visual words extracted from the background moves to the reverse direction, that is, left. To extract such moving regions, we compute the movement of each visual word based on [6]. Specifically, for a visual word in the keyframe, we find the same visual word in the frame after five frames. Then, by using hierarchical clustering, we group visual words which move to similar directions at spatially close positions into one moving region. Here, we represent each moving region as a vector $(x\_position, y\_position, size, horizontal\_movement, vertical\_movement)$. After that, we represent a set of extracted moving regions as one feature.

Finally, by gathering all of the above features, we represent a shot by using the total 94 features, as shown in Fig. 2. The first row represents the index of each feature. So, features from 1st to 90th are color, edge and visual word distributions. Here, each of these features is denoted not only by its index, but also by the notation which consists of a capital letter representing the feature name and a hyphenated digit representing the region. For example, the visual word distribution in 1-st region is denoted as 61st feature or *V-1*. Also, features from 91th to 93th are numbers of faces. Each of them is represented not only by its index, but also by the notation which consists of a capital letter representing the feature name and a hyphenated capital letter representing the size. For example, the number of middle-size faces is denoted as 92nd feature or *F-M*. Finally, the 94th feature represents a set of moving regions, and is denoted by *MR*. Based on the above 94-dimensional shot representation, we present our topic retrieval method.

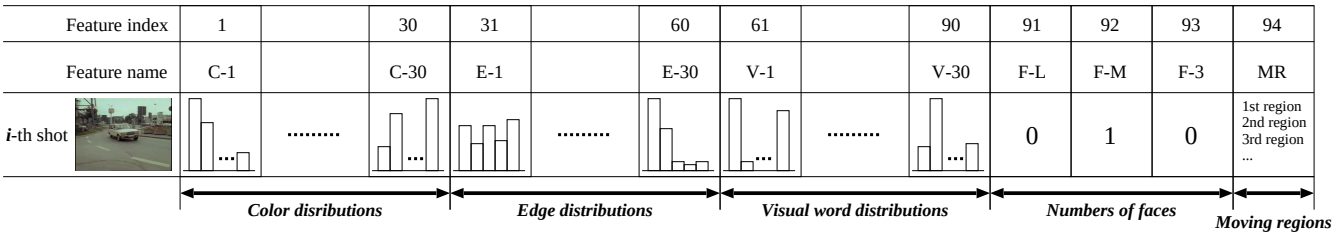

Figure 2. Our 94-dimensional shot representation.

3. Topic Retrieval based on Rough Set Theory

In this section, we firstly explain the motivation of using rough set theory to cover a large variation of features in the same topic. Then, we present our topic retrieval method based on rough set theory.

3.1. Motivation

Depending on various factors such as camera techniques, object movements, locations and so on, shots of the same topic contain significantly different features. Fig. 3 shows three shots of the topic "a car moves in the town". Here, from *shot 1* which takes a moving car in a tight shot, a large moving region is extracted. On the other hand, from *shot 2* and *shot 3* which take moving cars in long shots, middle-size moving regions are extracted. Also, since *shot 2* takes a car moving in a suburban area, few edges are extracted from the upper part where the sky is shown. On the other hand, since *shot 3* takes a car moving in an urban area, many vertical edges are extracted from the upper and middle parts where buildings are shown. Based on the above observation, we can assume that shots of the same topic are distributed in different subsets (subspaces) in a feature space.

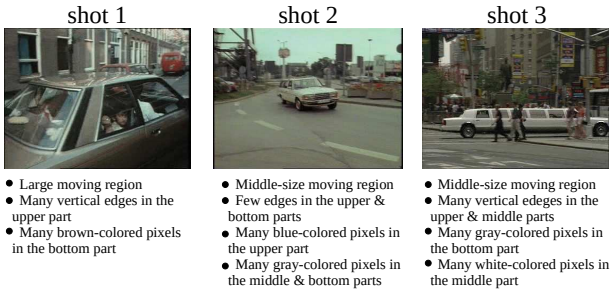

Figure 3. Example of shots which show the same topic but contain different features.

To find the above subsets, we use "rough set theory" which is a set-theoretic classification method based on indiscernibility relations among examples [9]. In our case, rough set theory examines whether positive examples can be discriminated from negative examples with respect to available features. Thereby, we can extract classification rules called "decision rules". Each of them characterizes a subset of positive examples which can be correctly discriminated from all negative examples. For example, in Fig. 3, a subset including shots taken in suburban areas like *shot 2* is characterized by the decision rule consisting of many blue-colored pixels in the upper part, many gray-colored pixels in the bottom part and a middle-size moving region. Therefore, by unifying such subsets, we can cover the whole set of positive examples for the topic.

However, a traditional rough set theory can deal only with categorical data [9]. On the other hand, as shown in Fig. 1, we extract different formats of features from each shot, such as histogram, integer number and a set of vectors. Note that crucial errors inevitably occur by discretizing a feature into a small number of categorical values. That is, the same categorical value is frequently assigned to semantically different shots. Thus, by using the idea of the recently proposed rough set theory for continuous data [16], we propose a rough set theory which can deal with various formats of features. Specifically, we define the indiscernibility relation between positive and negative examples based on their similarity for each feature.

3.2. Method

Given positive and negative examples, we aim to extract decision rules, each of which discriminates a subset of positive examples from all negative examples. Let p_i and n_j be i-th positive example $(1 \leq i \leq M)$ and j-th negative example $(1 \leq j \leq N)$, respectively. And, p_i^k and n_j^k represent p_i's and n_j's k-th feature $(1 \leq k \leq 94)$, respectively. By using the above notations, we explain rough set theory.

First, we represent positive and negative examples in the form of table called "decision table", as shown in Fig. 4 (a). In Fig. 4 (a), two positive examples p_1 and p_2 and two negative examples n_1 and n_2 are given for the topic "a car moves in the town". Each row represents an example. The

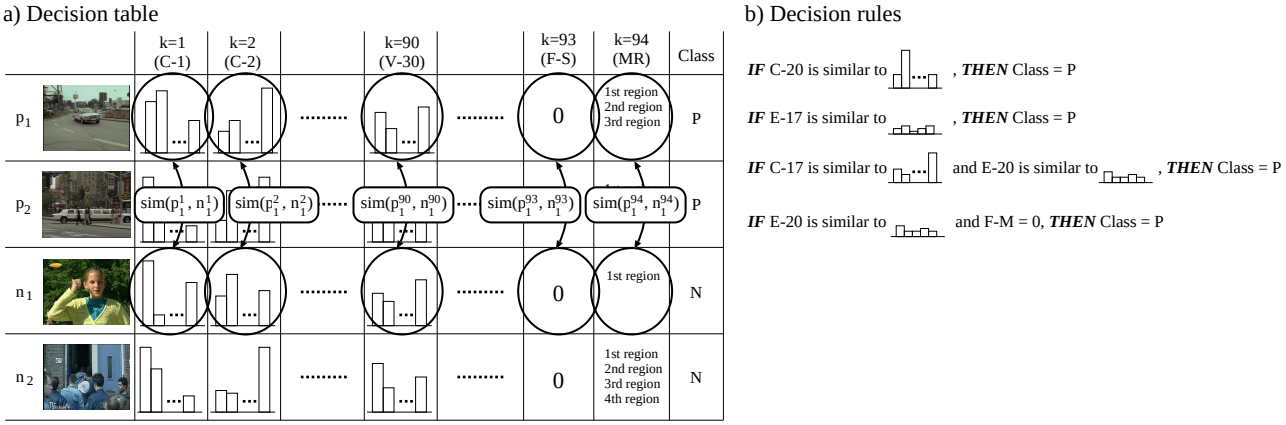

Figure 4. Example of a decision table and decision rules for the topic "a car moves in the town".

rightmost column indicates whether an example is positive ("P") or negative ("N"), while the other columns indicate features. Like this, the decision table provides available information for discriminating between positive and negative examples.

Then, for each pair of p_i and n_j, we collect "discriminative features" for which p_i and n_j are dissimilar to each other. In other words, p_i can be discriminated from n_j by using discriminative features. For example, by comparing p_1 to n_1 in Fig. 4 (a), we can find the color distribution in 17th region in Fig. 1 as a discriminative feature. It is because the sky is shown in 17th region in p_1, which is characterized by many blue-colored pixels. On the other hand, trees are shown in 17th region in n_1, which is characterized by many green-colored pixels. Thus, by using the color distribution in 17th region, we can discriminate between p_1 and n_1.

To extract such discriminative features, we calculate the similarity $sim(p_i^k, n_j^k)$ between p_i and n_j for k-th feature by using the following similarity measures; for color, edge and visual word distributions, we use histogram intersection. For numbers of faces, if p_i and n_j contain the same number of faces, we regard their similarity as 1, otherwise 0. For moving regions, we define the similarity between p_i and n_j as the similarity between the pair of the most similar moving regions. Here, we compare moving regions in p_i and n_j by using euclid distance.

Fig. 4 (a) illustrates the process of extracting discriminative features between p_1 and n_1. Thereby, we can collect the following set of discriminative features $f_{i,j}$ between p_i and n_j:

$$f_{i,j} = \{k \mid sim(p_i^k, n_j^k) < \beta^k\}, \tag{1}$$

where β^k is a pre-defined threshold for k-th feature. $f_{i,j}$ means that when at least one feature in $f_{i,j}$ is used, p_i can be discriminated from n_j.

Next, we extract sets of features which are needed to discriminate p_i from all negative examples. To do this, we need to simultaneously use at least one feature in $f_{i,j}$ for all negative examples. That is, we compute the following conjunction of $\vee f_{i,j}$:

$$df_i = \wedge\{\vee f_{i,j} |\ 1 \leq j \leq N\} \tag{2}$$

Suppose that the set of discriminative features between p_1 and n_1 is $f_{1,1} = \{$*C-17*, *C-20*, *E-17*, *F-M*$\}$ and the one between p_1 and n_2 is $f_{1,2} = \{$*C-20*, *E-17*, *E-20*$\}$. Then, we compute $df_1 = ($*C-17* $\vee$ *C-20* $\vee$ *E-17* $\vee$ *F-M*$) \wedge ($*C-20* $\vee$ *E-17* $\vee$ *E-20*$)$. df_1 can be simplified into $df_1^* = ($*C-20*$) \vee ($*E-17*$) \vee ($*C-17* $\wedge$ *E-20*$) \vee ($*E-20* $\wedge$ *F-M*$)$. This simplification is achieved by using the distributive law $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ and the absorption law $A \vee (A \wedge B) = A$. As a result, we can know that p_1 can be discriminated from all negative examples n_1 and n_2, by using the singleton of *C-20*, the singleton of *E-17*, the set of *C-17* and *E-20* or the set of *E-20* and *F-M*. Each of these represents a "reduct" which is a minimal set of features needed to discriminate p_1 from all negative examples

From each reduct, we construct a decision rule in the form of *IF-THEN* rule. For example, from the above four reducts, we can construct decision rules shown in Fig. 4 (b). Here, the conditional part of each decision rule is obtained by describing a reduct with p_1's features and similarities. Such a decision rule indicates a subset where p_1 can be correctly identified. Then, we gather decision rules extracted for all positive examples, and merge similar decision rules into one rule. As a result, we can extract decision rules which characterize subsets where multiple positive examples can be correctly identified. Finally, we retrieve shots which match with a larger number of decision rules than a pre-defined threshold.

4. Partially Supervised Learning for Negative Example Selection

In this section, we firstly explain the motivation of using partially supervised learning to select negative examples. Also, we describe the necessity of distinguishing relevant and irrelevant dimensions to appropriately calculate similarities among examples. Then, we present our partially supervised learning method.

4.1. Motivation

For the topic "a car moves in the town", Fig. 5 shows one positive example p_1 and two negative examples n_1 and n_2. In general, classifiers including decision rules in rough set theory are constructed, by comparing positive examples to negative examples and finding features which are contained only in positive examples. So, if p_1 is compared to n_1, a resulting classifier tends to retrieve shots which contain many red-colored pixels. But, this classifier is clearly meaningless because red-colored pixels are unimportant for the topic. On the other hand, if p_1 is compared to n_2, red-colored pixels are regarded as unimportant and visual words corresponding to the front widow and headlight are regarded as important. Thus, we can say that compared to n_1, n_2 is more effective for defining the topic.

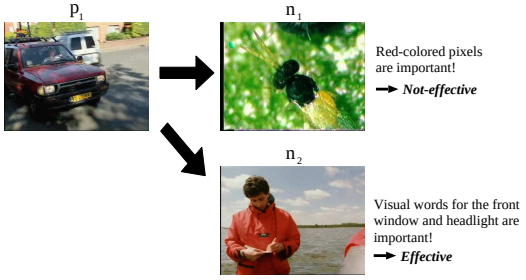

Figure 5. Example of effective and not-effective negative examples.

However, since a huge number of shots can be negative examples, it is difficult for a user to select the above kind of effective negative examples. Thus, we use "partially supervised learning" which builds a classifier only from positive examples, by selecting negative examples from unlabeled examples [7, 8, 1]. This technique is frequently used in Web document classification. Specifically, when constructing a classifier for identifying documents of a certain topic, we can easily select several positive examples, but cannot select effective negative examples from a huge number of documents on the Web. So, partially supervised learning is used to select effective negative examples by regarding documents on the Web as unlabeled examples. Inspired by this, we incorporate partially supervised learning into topic retrieval by regarding shots except for positive examples as unlabeled examples.

In our topic retrieval, we assume that for a topic, a user can provide only a small number of positive examples (at most 20). Considering this, most of existing partially supervised learning are not suitable for our topic retrieval, because they are based on the statistical distribution of positive examples. For example, methods in [8] and [1] use SVM and Naive Bayse to estimate the distribution of positive examples, respectively. But, such methods work well only when a sufficient number of positive examples are available for estimating the true distribution. On the other hand, the method in [7] selects negative examples based on similarities between positive and unlabeled examples. And, it is validated as effective when only a small number of positive examples are available. Thus, we use the method in [7] in our topic retrieval.

Note that since we represent examples by using various features, due to many irrelevant features, we cannot appropriately calculate similarities between positive and unlabeled examples (i.e. "curse of dimensionality") [2]. Hence, we have to distinguish relevant and irrelevant features. With respect to this, unlabeled examples show various topics and are characterized by different features. Thus, we detect features specific to each unlabeled example. To this end, we use "subspace clustering" which finds clusters of unlabeled examples in different subspaces of the high-dimensional feature space [2]. That is, each cluster is associated with a different set of features. For example, a cluster of unlabeled examples where the sky is shown is characterized by color and edge features in the upper part. Also, a cluster of unlabeled examples where an object moves on the road is characterized by the middle-size moving region and the color in the bottom part. Like this, for each unlabeled example, we detect specific features by finding the cluster including it. And, only by using these features, we calculate similarities of the unlabeled example to positive examples.

4.2. Method

Given positive examples for a topic, we collect negative examples based on two steps shown in Table 1. In the first step, we select "reliable negative examples" as unlabeled examples which are unlikely to be positive. That is, reliable negative examples are completely dissimilar to positive examples. For example, for the topic "a car moves in the town", reliable negative examples should include shots where the mountain is shown, shots where the beach is shown, and so on. But, as can be seen from n_1 in Fig. 5, reliable negative examples are not effective for defining the

Table 1. Overview of our partially supervised learning method.

Input: P (set of p-examples), U (set of u-examples),
Output: N (set of n-examples)
/* Reliable negative example selection */
1. Detect a set of positive features PF
2. Extract a set of rn-examples RN based on PF
/* Additional negative example selection */
3. $N = RN$
4. **while** true **do**
5. Cluster N into k clusters using PROCLUS
6. Extract a set of an-examples AN based on P and k clusters of N
7. **If** $|AN| == 0$, **then** break
8. $N = N \cup AN$
9. **end while**
10. return N

topic. So, in the second step, starting with reliable negative examples, we iteratively select "additional negative examples" as unlabeled examples which are more similar to positive examples than previously selected negative examples. For the above topic, additional negative examples should include shots where a person walks in the mountain, shots where the town is taken from the air, and so on. In this way, we aim to select effective negative examples as unlabeled examples which are as similar to positive examples as possible.

The two-step framework in Table 1 is based on the method proposed in [7]. But, we extend it for the following two points. First, although the method in [7] targets text data where each feature is a word frequency, we extend it to deal with our shot representation where features are represented in various formats, such as histogram, integer number and a set of vectors. Second, we use subspace clustering to appropriately calculate similarities between positive and unlabeled examples. Below, we mainly explain points extended from [7]. Note that, for the simplicity, we denote positive, negative, reliable negative, additional negative and unlabeled examples as "p-examples", "n-examples", "rn-examples", "an-examples" and "u-examples", respectively.

In the 1st line in Table 1, in order to accurately select rn-examples, we detect a set of "positive features" PF which are strongly associated with p-examples. For example, for the topic "a car moves in the town", the color feature in the 20th region in Fig. 1 may be selected as a positive feature, because it characterizes the gray-colored road shown in the bottom part. And, if a u-example matches with few positive features, it should be regarded as an rn-example. To detect positive features, we measure the association of each feature with p-examples based on similarities among p-examples. Specifically, for a feature f, we group p-examples into clusters with similar feature values (i.e. histogram, integer number or set of vectors). Here, depending on f, we use the similarity measure described in section 3.2. After that, we count the number of p-examples $n_P(f)$ in the largest cluster $C(f)$. Also, by applying $C(f)$ to u-examples, we count the number of u-examples $n_U(f)$ included in $C(f)$. Then, we evaluate how much f is associated with p-examples as follows:

$$H(f) = \frac{n_P(f)}{max_P} - \frac{n_U(f)}{max_U}, \tag{3}$$

where max_P and max_U are the largest value of $n_P(f)$ and the one of $n_U(f)$ among all features, respectively. They are used to normalize $n_P(f)$ and $n_U(f)$. Thus, $H(f)$ becomes larger if $C(f)$ includes a larger number of p-examples and a smaller number of u-examples. We regard f as a positive feature, if $H(f)$ is larger than the average of $H(j)$ for all features. Afterward, in the 2nd line in Table 1, we use the same ranking-based approach as [7], in order to calculate the similarity between a u-example and the set of p-examples in terms of PF. And, we regard the u-example as an rn-example, if the similarity is smaller than the average similarity for all u-examples.

In lines from 3th to 9th in Table 1, we regard a set of rn-examples as the initial set of n-examples. Then, we iteratively select an-examples as u-examples which are similar to already selected n-examples. Since n-examples show a variety of topics and contain different features, an an-example is not similar to all n-examples. Let us recall the above example. Here, an-examples where a person walks in the mountain are similar only to rn-examples where the mountain is shown. Considering such a variety of n-examples, we firstly group n-examples into clusters and calculate the similarity between a u-example and each cluster. Particularly, because of the high-dimensionality of our shot representation, we use subspace clustering "PROCLUS" proposed in [2]. PROCLUS iteratively improves k clusters where bad clusters such as the ones with few n-examples are substituted with new clusters by randomly selecting cluster centers. In each cluster, if the average similarity among n-examples for one feature is larger than the statistical expectation, this feature is associated with the cluster. As a result, the cluster represents a subspace consisting of its associated features.

Then, for i-th cluster of n-examples, we compute the centroid C_i in the subspace consisting of associated features F_i. Also, we compute the centroid of p-examples C_P in the subspace consisting of positive features PF. Then, we examine whether a u-example u can be regarded as an an-example in the following way:

$$Sim_{F_i}(u, C_i) > \mu_i, \tag{4}$$

$$Sim_{F_i}(u, C_i) - Sim_{PF}(u, C_P) > \gamma_i, \tag{5}$$

where $Sim_{F_i}(u, C_i)$ is the similarity between u and C_i in

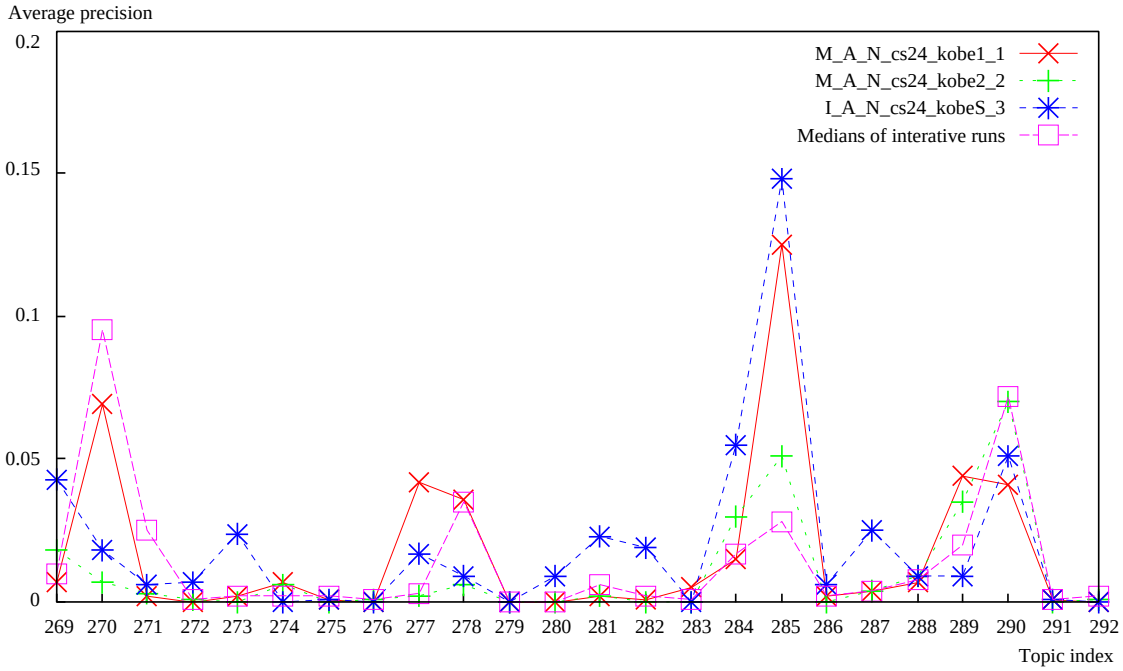

Figure 6. Overview of evaluation results for our submitted runs.

terms of F_i. Specifically, we calculate $Sim_{F_i}(u, C_i)$ as the average of similarities for all features in F_i. Similarly, $Sim_{PF}(u, C_P)$ is calculated as the similarity between u and C_P in terms of PF. Also, μ_i and γ_i are respectively average values of equations (4) and (5) for n-examples in i-th cluster. Thus, u is selected as an an-example if it is not only sufficiently similar to i-th cluster, but also much more similar to i-th cluster than to the set of p-examples. Finally, as shown in the 7th and 8th lines, we iterate the above an-example selection until no an-example is selected.

5. Experimental Results

In TRECVID 2009 search task, we submitted the following three runs. In $M\_A\_N\_cs24\_kobe1\_1$, we apply rough set theory to manually selected positive and negative examples. In $M\_A\_N\_cs24\_kobe2\_2$, we apply rough set theory to manually selected positive examples and negative examples selected by partially supervised learning. Here, we regard shots in the development videos as unlabeled examples. In $I\_A\_N\_cs24\_kobeS\_3$, from results of $M\_A\_N\_cs24\_kobe1\_1$, we select additional positive and negative examples, and apply rough set theory to all of positive and negative examples. Note that $I\_A\_N\_cs24\_kobeS\_3$ is a supplemental run which violates the maximum time limit due to the slow search speed.

Fig. 6 shows the overview of the above three submitted runs. Here, for comparison, we show medians of all interactive runs (since the number of manually-assisted runs is only three, we do not use their medians). As can be seen from Fig. 6, $M\_A\_N\_cs24\_kobe1\_1$ and $M\_A\_N\_cs24\_kobe2\_2$ which need no user interaction, achieve comparable performances to medians of interactive runs. This validates the effectiveness of our method.

Now, by using results of $M\_A\_N\_cs24\_kobe1\_1$, we examine whether rough set theory can cover a large variation of features in the same topic. Fig. 7 shows retrieved shots for 277-th, 285-th and 289th topics. As can be seen from Fig. 7, for the same topic, we can retrieve shots which are characterized by different backgrounds, different numbers of persons and different shot sizes. The reason for this is that rough set theory extracts decision rules which characterize essential semantic contents in a topic. For example, decision rules for the 289th topic characterize a person, table and indoor situation. Thus, by combining such decision rules, we can retrieve various shots of the same topic.

However, the performance of our current method is far from the satisfactory. Regarding this, one of main reasons is that numbers of positive and negative examples are too small to define a topic. Specifically, on average, we only use 15.7 positive and 24.2 negative examples in $M\_A\_N\_cs24\_kobe1\_1$. Compared to $M\_A\_N\_cs24\_kobe1\_1$, $I\_A\_N\_cs24\_kobeS\_3$ uses more positive and negative examples and achieves better perfor-

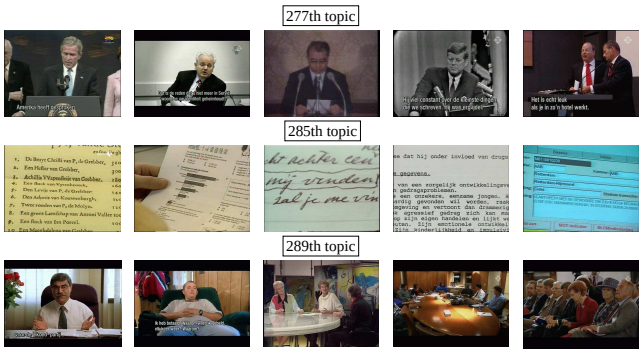

Figure 7. Shots retrieved by rough set theory for 277th, 285th and 289th topics.

mances for most topics. So, we conduct an additional experiment to examine the change of performances depending on numbers of positive and negative examples. Here, due to the difficulty of selecting many positive examples, we select positive examples from TRECVID 2008 test videos, where shots relevant to each topic are marked by NIST. Also, we select negative examples from TRECVID 2008 test videos. Then, based on the above positive and negative examples, we search TRECVID 2009 test videos.

Table 2 shows the result of the above additional experiment. Here, we retrieve 285th and 286th topics by using different numbers of positive and negative examples. And, for each retrieval result, we compute the precision in 1000 retrieved shots. As shown in Table 2, although the precision of 100 positive and 100 negative examples for 286th topic is slightly smaller than the one of 50 positive and 50 negative examples, using a large number of examples generally leads to a significant improvement of the retrieval performance. Thus, it is one of our important future works how to efficiently collect a sufficient number of positive and negative examples.

Table 2. Performances using different numbers of positive and negative examples.

# of positives	10	20	50	100
# of negatives	50	50	50	100
285th topic	0.027	0.219	0.317	0.400
286th topic	0	0.001	0.027	0.025

In order to examine the effectiveness of partially supervised learning, we conduct a preliminary experiment. Here, we use TRECVID 2008 development and test videos. And, we retrieve the following three topics; *Topic 1:* a person opens a door, *Topic 2:* a person talks on the street and *Topic 3:* a car moves in the town. Table 3 summarizes the result for the preliminary experiment. As shown in the second column, partially supervised learning PSL is compared to two different negative example selection methods, $Manual$ and $Random$. In $Manual$, negative examples are manually selected while they are randomly selected in $Random$. For all of $Manual$, $Random$ and PSL, as in the third column, we use the same positive examples for each topic. In addition, the fourth column shows that in both of $Random$ and PSL, we select the same number of negative examples (i.e. 50). Finally, the fifth column presents precisions in 300 retrieved shots by using different sets of negative examples.

Table 3. Comparison of partially supervised learning to the other negative example selection methods.

	Method	# of pos.	# of neg.	P@300
Topic 1	$Manual$	9	16	0.070
	$Random$	9	50	0.087
	PSL	9	50	0.070
Topic 2	$Manual$	11	16	0.087
	$Random$	11	50	0.050
	PSL	11	50	0.050
Topic 3	$Manual$	9	14	0.217
	$Random$	9	50	0.127
	PSL	9	50	0.170

As can be seen from Table 3, except for *Topic 1*, $Manual$ usually leads to the best performances. Also, we find that the effectiveness of PSL depends on the number of shots relevant to a topic. Specifically, the number of shots relevant to *Topic 3* is relatively large while the one to *Topic 1* is very small. For *Topic 3*, PSL can accurately select negative examples by analyzing features in shots. On the other hand, $Random$ wrongly selects some relevant shots as negative, since it does not analyze any features. But, for *Topic 1*, due to the rarity of relevant shots, PSL cannot appropriately select effective negative examples for defining *Topic 1*. Compared to this, due to the rarity of relevant shots and the randomness, $Random$ can select effective negative examples without selecting relevant shots as negative. Considering the above result, it seems to be useful to change negative example selection methods depending on topics. But, before this conclusion, we need to further explore the characteristic of partially supervised learning by testing it on various topics, such as topics in TRECVID 2009.

6. Conclusion and Future Works

In this paper, we introduced a method which defines any interesting topic from positive and negative examples. Particularly, to cover a large variation of features in a topic, we use rough set theory which defines the topic as a union of subsets. In each subset, some positive examples are correctly discriminated from all negative examples. Also, considering the difficulty of selecting effective negative examples for defining a topic, we use partially supervised learning to select negative examples from unlabeled examples. Here, to appropriately calculate similarities among examples in a high-dimensional feature space, we use subspace clustering which finds subspaces characterized by different sets of features. Evaluation results on TRECVID 2009 video data validate the effectiveness of rough set theory. For partially supervised learning including subspace clustering, we need further experiments.

In order to improve the performance of our current method, we will explore the following three issues. First, for our method, it is crucial what kind of similarity measure is used. In order to obtain a similarity measure which is closely related to human perception, we plan to learn the similarity measure from pairs of training images (or regions) which are labeled as "similar" or "dissimilar" [4]. Second, all of decision rules extracted by rough set theory are not useful. For example, for the topic "one or more dogs walk, run or jump", many decision rules characterize unimportant semantic contents, such as grass in the background. Thus, for a topic, we aim to examine the usefulness of each decision rule based on external resources. Especially, we use images and videos on the Web, which are searched by words related to the topic. Finally, although topics are independently retrieved in our current method, we plan to retrieve a topic by considering its relation to previously retrieved topics. Specifically, for the topic "a car moves in the town", we should not retrieve shots which match with decision rules for unrelated topics, such as "a ship in the water", "a person talks indoor" and so on. In order to define such relations among topics, we aim to organize previously retrieved topics into a "topic ontology".

Acknowledgments: This research is supported in part by Strategic Information and Communications R&D Promotion Programme (SCOPE) by the Ministry of Internal Affairs and Communications, Japan.

References

[1] B. Liu, Y. Dai, X. Li, W. Lee and P. Yu. Text classifiers using positive and unlabeled examples. In *Proc. of ICDM 2003*, pages 179–188, 2003.

[2] C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu and J. Park. Fast algorithms for projected clustering. In *Proc. of SIGMOD 1999*, pages 61–72, 1999.

[3] C. Snoek and M. Worring. Multimedia event-based video indexing using time intervals. *IEEE Transactions on Multimedia*, 7(4):638–647, 2005.

[4] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. of CVPR 2007*, pages 1–8, 2007.

[5] E. Sande, T. Gevers and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proc. of CVPR 2008*, pages 1–8, 2008.

[6] F. Wang, Y. Jiang and C. Ngo. Video event detection using motion relatively and visual relatedness. In *Proc. of ACM MM 2008*, pages 239–248, 2008.

[7] G. Fung, J. Yu, H. Ku and P. Yu. Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):6–20, 2006.

[8] H. Yu, J. Han and K. Chang. PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

[9] J. Komorowski, Z. Pawlak, L. Polkowski and A. Skowron. Rough sets: A tutorial. In S. Pal and A. Skowron, editor, *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pages 3 – 98. Springer, 1999.

[10] L. Zhang, F. Lin and B. Zhang. A CBIR method based on color-spatial feature. In *Proc. of TEHNCON 1999*, pages 166–169, 1999.

[11] M. Naphade, J. Smith, J. Tešić, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[12] Moving Picture Coding Group (MPEG). *MPEG-7 Overview (version 10)*. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, 2004.

[13] N. Haering, R. Qian and M. Sezan. A semantic event-detection approach and its application to detecting hunts in wildlife video. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):857–868, 2000.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR 2001*, pages 511–518, 2001.

[15] S. Ebadollahi, L. Xie, S. Chang and J. Smith. Visual event detection using multi-dimensional concept dynamics. In *Proc. of ICME 2006*, pages 881–884, 2006.

[16] Y. Leung, M. Fischer, W. Wu and J. Mi. A rough set approach for the discovery of classification rules in interval-valued information systems. *International Journal of Approximate Reasoning*, 47(2):233–246, 2008.