3-2006

Threading and autodocumenting news videos: A promising solution to rapidly browse news topics

Xiao WU

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Qing LI

Xiao Wu, Chong-Wah Ngo, and Qing Li

Threading and Autodocumenting News Videos

A promising solution to rapidly browse news topics

News videos constitute a huge volume of daily information. It has become necessary to provide viewers with a concise and chronological view of various news themes through story dependency threading and topical documentary. This article presents techniques in threading and autodocumenting news stories according to topic themes. Initially, we perform story clustering by exploiting the duality between stories and textual-visual concepts through a coclustering algorithm. The dependency among stories of a topic is tracked by exploring the textual-visual novelty and redundancy of stories. A novel topic structure that chains the dependencies of stories is then presented to facilitate the fast navigation of the news topic. By pruning the peripheral and redundant news stories in the topic structure, a main thread is extracted for autodocumentary.

INTRODUCTION

News videos form a major portion of information disseminated in the world everyday, which constitutes an important source for topic tracking and documentation. Most people are interested in keeping abreast of the main story or thread and new events. However, it is becoming very difficult to cope with the huge volume of information that arrives each day. The scenario is more complicated when considering news stories from multiple sources, such as CNN, BBC, ABC, and CCTV. Topic threading and autodocumentary appears as one prominent solution to this problem. For example, when searching the topic "London bombing," it is more interesting if the system can provide a concise overview or a fresh development of the topic, rather than just showing a list of items and leaving the viewers alone to find out the dependencies among them.

Clustering continuously reported or time-evolving stories in news videos is a critical step for topic tracking and summarization. Previous research has mostly focused on assembling news stories into a few coarse classes such as politics, sports, and health. For news video threading and autodocumentary, a fine granularity of classification that can put together evolving stories according to topic themes is more attractive. To this end, we present a two-way clustering algorithm (*coclustering*) to effectively fuse textual and visual concepts in news.

In news videos, a topic can comprise multiple events, and each event is usually under the umbrella of one theme [1]. News stories record the gradual changes of a topic over time and contain valuable information for topic threading and autodocumentary. Figure 1 gives an example of three evolving news stories in the "Arkansas school shooting" topic, in which each keyframe represents a shot. In a topic, new themes emerge over time. Some themes evolve slowly while others remain intact throughout the topic. These themes can be described by concepts that include key words and keyframes that may repeatedly appear, evolve, or change. In news videos, stories are often accompanied by a couple of shots and speech transcripts that tend to be used repeatedly during the course of a topic. Images that repeat in the topic, as shown in Figure 1, are called *near-duplicate keyframes* (NDK). Basically, they are similar to each other, but different in the capturing conditions, acquisition times, rendering conditions, and/or editing operations [2]. Recently, techniques for near-duplicate detection have been proposed in [2]–[4].

There is a great deal of redundancy in news stories of one topic, especially when they are broadcast from different channels. Generally speaking, the primary interest of viewers is to learn the evolution and highlights of a topic, rather than browsing through every part of the story. Thus, rating a story content by novelty/redundancy detection would prove to be useful. Identifying redundancy and overlap can help minimize the overhead associated with tracking and documentary. In addition, the degree of redundancy offers strong indication on how a topic has evolved.

When browsing through a very large-scale corpus, one important task is to construct the dependencies of news stories by threading them under the same topic. The process of identifying dependencies among news stories is called *topic threading*. Automatic tracking of the evolution of news stories can facilitate indexing, retrieval, and topic summarization. The outcome is the semantic organization of news stories that allows viewers to rapidly interpret and analyze a topic. Based on the constructed topic structure, we extract the main thread to represent the main development of a topic by removing the redundant and peripheral news stories from the topic structure.

For story clustering, novelty/redundancy detection, and topic threading, most previous works (such as [1] and [5]–[9]) consider these problems in the text domain. However, the employment of either textual or visual concepts may not be sufficient since either concept can emerge differently over time. The pure textual method may overlook the interactions between visual and textual information, i.e., the visual contents determine the set of shots on which text summarization will be considered, but the textual information does not have a say about how the set of shots is selected. One robust method is to take into account both textual and visual concepts while exploiting the significance of both.

We present issues in story coclustering and dependency threading, equipped with the fusion of textual-visual concepts. These issues ultimately lead to the construction of the topic structure and framework (shown in Figure 2), which facilitates the automatic generation of a documentary.

[FIG1] An example of time-evolving news stories and near-duplicate keyframes.

COCLUSTERING OF NEWS STORIES

Two-way clustering, namely coclustering, was originally proposed in [6] to exploit the duality of documents and words. Recently, this technique is employed in [3] and [8] for shot-based topic labeling [3] and cross-source story clustering [8]. While both [3] and [8] consider solely textual information, we exploit both textual and visual concepts for more effective story-level clustering.

BIPARTITE GRAPH MODEL

We model the relationship of stories and textual-visual concepts with a bipartite graph model (Figure 3). We denote S, T, and V as a set of stories and textual and visual concepts, respectively. There is no intersection among these sets. A bipartite graph is described as $G = (P, S, E)$, where $P = T \cup V$, $T \cap V = \phi$ and $P \cap S = \phi$, $V = \{v_1, \ldots, v_g\}$, $T = \{t_1, \ldots, t_m\}$, $S = \{s_1, \ldots, s_n\}$ and E is a set of edges $\{\{s_i, p_j\} : s_i \in S, p_j \in V \text{ or } p_j \in T\}$. In this case, P represents the concepts that include words and keyframe clusters. An edge $\{s_i, t_j\}$ exists if news story s_i contains word t_j, while an edge $\{s_i, v_j\}$ exists if news story s_i contains a keyframe in cluster v_j (note that $T \cap V = \phi$). In this model, there is no edge between stories, keyframe clusters, and words.

COCLUSTERING ALGORITHM

The problem of coclustering is to partition the bipartite graph into subgraphs by considering the cooccurrence between stories and textual-visual concepts based on spectral graph partitioning. More details can be found in our recent work [10].

We treat words and keyframe clusters that are included in the news stories as concepts. The whole news story collection is represented by a concept-by-story matrix A, whose rows correspond to concepts and columns to news stories. The matrix A is represented as

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where A_1 is a word-by-story matrix whose rows correspond to words and columns to news stories. The association between a news story j and a word i is calculated by

$$A_{ij} = W_{ij} = tf_{ij} \times idf_i,$$

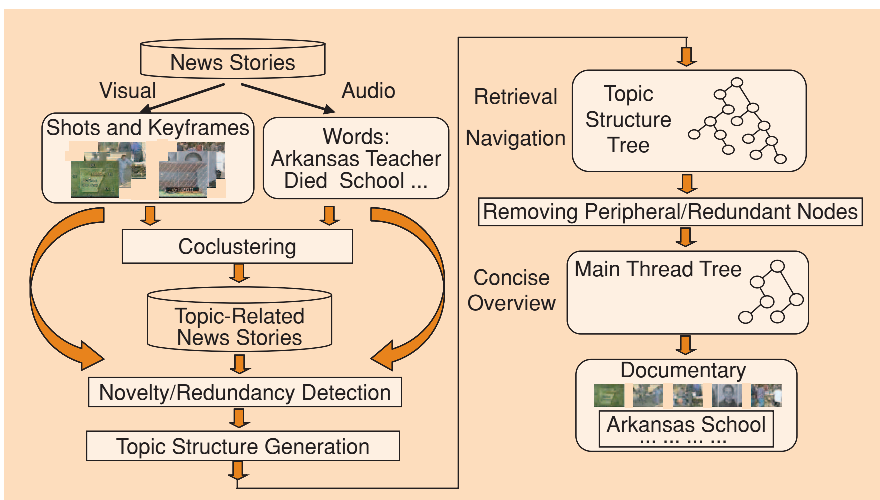
where tf_{ij} is the term frequency of word i in story j, and idf_i is the inverse document frequency of word i. These story-word associations form the matrix A_1. A_2 is a keyframe-by-story matrix whose rows correspond to keyframe clusters and columns to news stories. For keyframes, color histograms are extracted, and the k-means algorithm is applied to cluster the keyframes into g groups. We regard the keyframes in one group as one visual concept since they are mostly similar or nearly duplicate keyframes. The matrix A_2 is constructed by computing story-keyframe association:

$$A_{ij} = W_{ij} = kf_{ij} \times \log_2(N/sf_i),$$

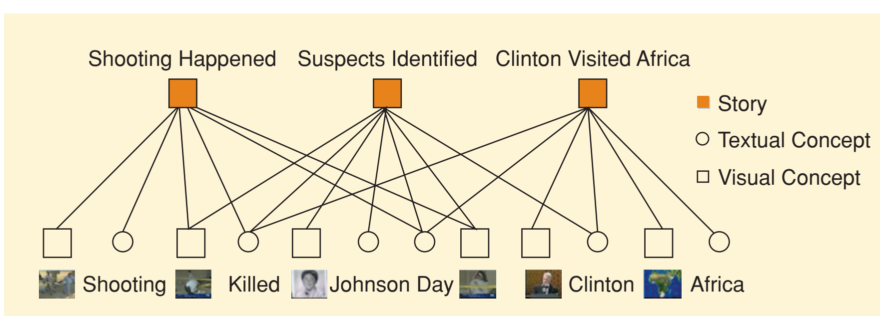
where kf_{ij} is the frequency of keyframe group i in story j, N is the number of news stories, and sf_i is the story frequency of keyframe group i. In this case, A_1 and A_2 are $m \times n$ and $g \times n$ matrices, respectively, where m is the number of words, g is the number of keyframe clusters, and n is the number of news stories.

Our coclustering algorithm, which considers textual and visual concepts, consists of the following steps:

1) Set up word-by-story matrix A_1 and keyframe-by-story matrix A_2.
2) Construct the affinity matrix A.
3) Calculate the normalized matrix $A_n = D_1^{-1/2} A D_2^{-1/2}$, where D_1 and D_2 are diagonal matrices.
4) Compute $l = \lceil \log_2 k \rceil$ singular vectors of A_n, $u_2, u_3, \ldots, u_{l+1}$, and $v_2, v_3, \ldots, v_{l+1}$, and form the matrix



[FIG2] Proposed framework.



[FIG3] A bipartite graph.

[TABLE1] THE PERFORMANCE COMPARISON OF THE STORY COCLUSTERING.

			WORD + KF		WORD		KF	
ID	TOPIC THEME	#	PREC	RECALL	PREC	RECALL	PREC	RECALL
1	YELTSIN FIRED THE WHOLE CABINET	6	6/8	6/6	5/15	5/6	1/1	1/6
2	CLINTON VISITED AFRICA	19	19/26	19/19	6/15	6/19	9/81	9/19
3	CLINTON SEXUAL SCANDAL	7	4/11	4/7	3/7	3/7	2/25	2/7
4	ARKANSAS SCHOOL SHOOTING	33	32/43	32/33	15/21	15/33	19/81	19/33
5	IRAQ WEAPON PROBLEM	5	5/11	5/5	2/7	2/5	5/81	5/5
6	HOSPITAL EMPLOYEE KILLED PATIENTS	5	5/6	5/5	4/21	4/5	2/34	2/5
	AVERAGE F-MEASURE OF 22 TOPICS		**0.643**		**0.346**		**0.270**	

$$Z = \begin{bmatrix} D_1^{-1/2} & U \\ D_2^{-1/2} & V \end{bmatrix},$$

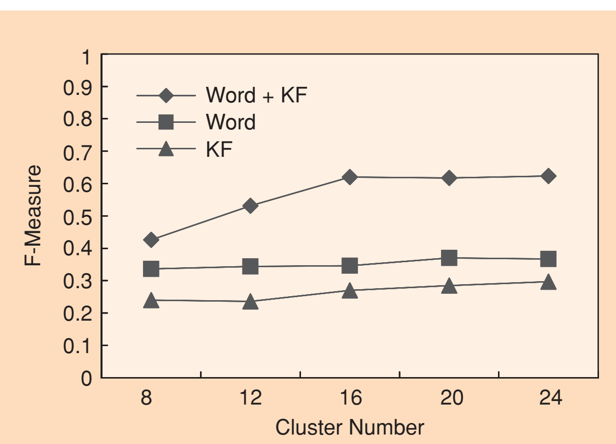
where $U = [u_2, u_3, \ldots, u_{l+1}]$ and $V = [v_2, v_3, \ldots, v_{l+1}]$.

5) Run the k-means algorithm on the l-dimensional data Z to obtain the desired k-way multipartition.

EXPERIMENT AND EVALUATION

We selected a week's videos, from 23–30 March 1998, from TRECVID-2004 [11], which include 14 videos from CNN and ABC news, as our test set. The story and shot boundary partitions are labeled by TRECVID. The textual concepts are a list of words extracted from speech transcripts by an automatic speech recognition (ASR) system at LIMSI [12], while the visual concepts are the set of representative keyframes extracted from video corpus. After data preprocessing, such as word stemming and stop-words removal, the data set consists of 174 news stories, 1,653 keyframes, and 3,180 words. To compute the story-keyframe association, we ran the k-means algorithm, based on the color histogram, to cluster the 1,653 keyframes into $g = 400$ groups. We built a ground truth table by manually labeling stories according to topic themes. To ensure the fairness of the comparison, the topics that have only one news story were omitted from the experiment.

We use the F-measure (FM) [13] as the metric for performance evaluation. The FM assesses the quality of clusters by comparing the detected clusters with the ground-truth clusters to find the maximal matching. The value of FM ranges 0–1. The higher the value of FM, the better the performance of the clustering algorithm.



[FIG4] Overall performance comparison of the coclustering.

We compared three coclustering approaches: with textual-visual concepts (WORD + KF), with textual concept only (WORD), and with visual concept only (KF). Figure 4 shows the performance in terms of the FM versus the number of clusters. Overall, the WORD + KF approach has a better performance than just the WORD or the KF approach. The WORD + KF approach correctly clusters most related evolving stories into the same-theme topic clusters. The WORD and KF approaches, in contrast, partition the related news stories into different clusters.

Table 1 lists the recall and precision of some ground-truth clusters, when $k = 16$. As indicated in the table, the WORD + KF approach has better precision and recall than the WORD and KF approaches. The topics such as "Arkansas school shooting," "Clinton visited Africa," and "Yeltsin fired the whole cabinet" are correctly clustered by integrating textual and visual concepts.

TOPIC STRUCTURE

After story coclustering, finding the dependencies among stories still remains a challenging task. In [7], Nallapati et al. threaded the news topics by first clustering stories into events and then constructing dependencies among them as a graph structure. In [14], the thread structure is formed with a chronologically ordered directed graph. The topic construction in [7] and [14] only utilizes textual features and ignores visual features, which carry useful cues for threading. In [3], Duygulu et al. presented a technique to detect and track the repeated sequence of shots based on textual and visual cues. Zhu et al. [15] also presented a hierarchical video content description and summarization strategy supported by a joint textual and visual similarity. Topic structure and story dependency, however, were not addressed in [3] and [15]. In this section, we present a novel threading approach for topic structure construction, by exploiting the novelty and redundancy of stories through both textual and visual information. The thread structure considers two aspects of relation: semantic and chronological dependencies.

NOVELTY/REDUNDANCY DETECTION

Previous attempts of novelty/redundancy detection [1], [9], [16]–[18] were based entirely on textual cues. We combine textual-visual information using the cosine distance for story similarity measurement. The cosine of the angle between a news story vector and each previously delivered news story vector determines the redundancy score for that news story. It is a pairwise measure, defined by

$$R_{cd}(S_i \mid S_1, \ldots, S_{i-1}) = \max_{1\le j\le i-1} R_{cd}(S_i \mid S_j) = \max_{1\le j\le i-1} \times \frac{\sum_{k=1}^{m} w_k(s_i)w_k(S_j) + \sum_{l=1}^{n} v_l(S_i)v_l(S_j)}{\sqrt{\sum_{k=1}^{m} w_k(S_i)^2 \sum_{k=1}^{m} w_k(S_j)^2}\sqrt{\sum_{l=1}^{n} v_l(S_i)^2 \sum_{k=1}^{n} v_l(S_j)^2}}.$$

The weighting function used in our experiments is a *tf-idf* function [19] specified by the following formulas:

$$w_k(S_i) = \frac{tf_{wk,S_i}}{tf_{wk,S_i} + 0.5 + (1.5 * \frac{len(S_i)}{awl})} * \frac{\log \frac{g+0.5}{sf_{wk}}}{\log(g+1.0)}$$

$$v_l(S_i) = \begin{cases} \frac{tf_{vl,Si}}{tf_{vl,Si}+0.5+(1.5*\frac{kflen(S_i)}{akfl})} * \frac{\log \frac{g+0.5}{sf_{vl}}}{\log(g+1.0)} & \text{if } v_l \notin NDK \\ \alpha \frac{tf_{vl,Si}}{tf_{vl,Si}+0.5+(1.5*\frac{kflen(S_i)}{akfl})} * \frac{\log \frac{g+0.5}{sf_{vl}}}{\log(g+1.0)} & \text{if } v_l \in NDK, \end{cases}$$

where

- m= the number of words in the vocabulary
- n=the number of keyframes in the keyframe sets
- $w_k(S_i)$= the weight of word w_k in news story S_i
- $v_l(S_i)$= the weight of keyframe v_l in news story S_i
- $tf_{wk,Si}$= the frequency of word w_k in news story S_i
- $tf_{vl,Si}$= the frequency of keyframe v_l in news story S_i
- awl= the average number of words in news stories for the topic
- $akfl$ = the average number of keyframes in news stories for the topic
- sf_{wk}= the number of news stories for the topic that contain word w_k
- sf_{vl}= the number of news stories for the topic that contain keyframe v_l
- $len(S_i)$= the number of words in news story S_i
- $kflen(S_i)$= the number of keyframes in news story S_i
- g = the number of news stories for the topic
- α= the weight factor of NDK
- NDK= the set of NDK.

In news stories, the lengths of words and keyframes are different. Treating them in the same way is not appropriate. Usually, the number of words in a news story is much larger than keyframes. Moreover, each keyframe represents a shot, which can convey more information than a couple of words. If two news stories have a pair of similar shots (i.e., duplicate or near-duplicated keyframes), it seems that these two news stories are more similar to some extent than two news stories that have a pair of sentences (a couple of words) in common. Therefore, we introduce a parameter α to amplify the importance of keyframes. In our experiment, $\alpha = 3$.

Different people have different definitions of redundancy and different redundancy thresholds. In order to alleviate such inconsistency, similar to [9], we classify news stories into three classes in this article: redundant, evolving, and novel. If a news story contains no new information, it is regarded as redundant, as is a story which is a review or a near-duplicate of a previous story. News stories that have some new information and contain some redundant content are marked as evolving news stories. Evolving stories usually convey the gradual development of a theme. Finally, news stories in which most of the content is new, are marked as novel stories, indicating the emergence of new themes.

Besides cosine distance, other measures such as new word count [1] and set difference [1] can also be used. Our empirical results, nevertheless, show that cosine distance indeed gives the best performance, by investigating the similarity distribution of novel, evolving, and redundant stories.

TOPIC STRUCTURE GENERATION

The novelty/redundancy detection of news stories gives the closeness measure among news stories and provides the dependency pairs, which indicate the story development and dependency. Next, we describe the process of constructing the topic structure binary tree.

First, we define the story dependency pair as $(S_{\text{child}}, S_{\text{parent}})$ that denotes that the child story S_{child} is dependent on the parent story S_{parent}. The dependency pair set *D(T)* is the set of all story dependency pairs (S_i, S_j) in the topic T. Mathematically, this can be expressed as:

$$D = \begin{cases} \{(S_i, S_j) \mid S_j = \arg \max_{1\le j\le i-1 \&\& t(S_j)<t(S_i)} R(S_i, S_j) \\ \quad \text{if } S_i \text{ is redundant or evolving} \\ \{(S_i, S_0) \mid \text{if } S_i \text{ is a novel story}\}, \end{cases}$$

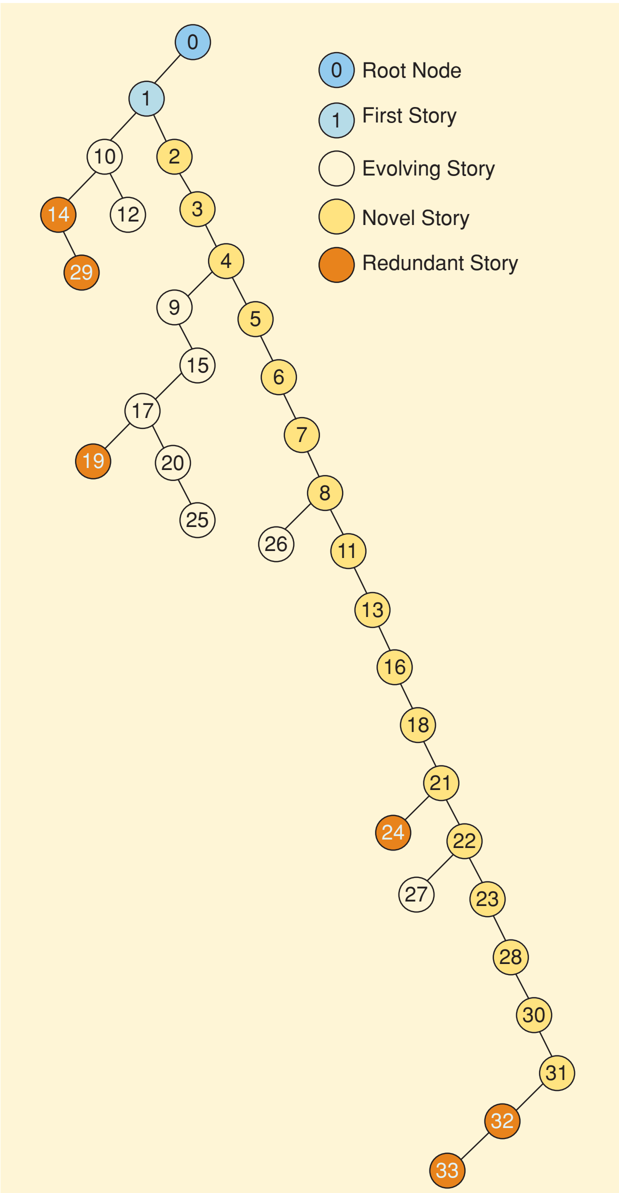
where $R(S_i, S_j)$ is the novelty/redundancy score, $t(S_i)$ is the publication time of story S_i, S_j is the most similar one to S_i, and S_0 is the root node (stories linked to it have no most similar story).

Note that the dependency pair has a direction. Story dependency pair (S_i, S_j) is not equal to (S_j, S_i). For (S_i, S_j), S_j happens before S_i, while for (S_j, S_i), story S_i happens before S_j. From the definition of the dependency pair, we can see that a story only has one parent and a parent can have more than one child. For example, for dependency pair (S_i, S_j), S_j is the most similar story to S_i among all the previous $i-1$ stories. There are no other dependency pairs with the same child story S_i. But for a story S_j in *D(T)*, it may have more than one dependency pair with S_j as the parent story, such as (S_i, S_j), (S_n, S_j).

The topic structure is represented by a binary tree $<S, E>$, where S is the set of news stories and E is the set of edges. We build up the topic structure based on the dependency pairs. To illustrate the process, assume we are given a set of news stories $S = \{S_1, S_2, \ldots, S_n\}$ on a given topic T, and these news stories are ordered by their time of publication. Each news story is compared with all previously delivered news stories to find the most similar news story using the novelty/redundancy detection method described previously. Next, n story dependency pairs are formed based on their publication orders, resulting in each story having one dependency pair.

Based on the story dependency pair set $D(T)$ of a topic T, a topic structure binary tree can be constructed by using the parent/child relation of dependency pairs. The previously delivered news story S_{parent} that is most similar to the current news story S_{child} is regarded as the parent of S_{child}. S_{child} is linked to the left branch of S_{parent}. If the left branch is occupied, then S_{child} is linked to the right branch of the left child of S_{parent} until the right branch is empty. If a story is novel, its parent is set as the root node. The first story of a topic is the left child of the root node. The main process to generate the topic structure binary tree is listed as follows.

Algorithm: Topic Structure Generation
Input: the list of news stories based on their chronological orders
Output: the topic structure binary tree



[FIG5] The topic structure binary tree for the "Arkansas school shooting" (we ignore the redundancy score attached with each edge).

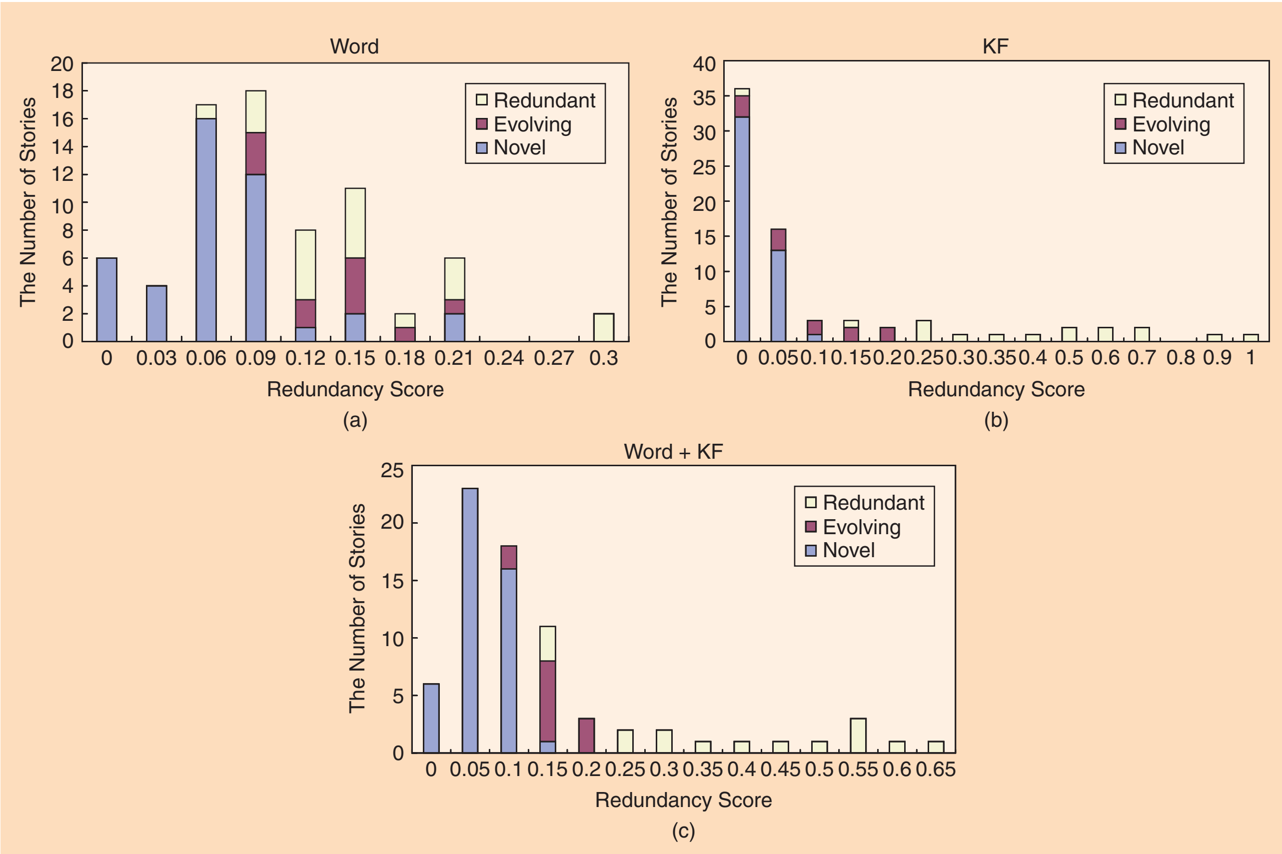
Procedure:

1) Order each news story S_i by their publication time.
2) Calculate the novelty/redundancy measure to find the previously delivered news story S_j with the largest redundancy score, and set the story dependency pair (S_i, S_j). If news story S_i is novel, set the dependency pair (S_i, S_j), where $j = 0$.
3) Create a new node for news story S_i and locate the parent node S_j.
4) If the left branch of the parent node S_j is null, add the new node as the left child of S_j.
5) If S_i has already had a left child, the pointer is set to its left child, say S_k. If the right branch of S_k is empty, then the new node S_i is directly linked to the right branch of S_k. Otherwise, the pointer is continuously moved to its right branch until the right branch is empty. Then, the story S_i is linked as its right child.
6) Repeat steps 2–5 for each news story.

Figure 5 shows the topic structure of the "Arkansas school shooting." Let us see how this binary tree is constructed step by step: When the first story S_1 of the "Arkansas school shooting" arrives, it is a novel story, and the root node S_0 is empty. S_1 is linked to the left branch of S_0. The second story S_2 is also a novel story, which has the dependency pair (S_2, S_0). However, the left child of S_0 has been occupied by S_1, so it checks the right branch of S_1 to see whether its right branch is empty. S_1 has noright child, so the story S_2 is linked as its right child. Since all the first eight stories are novel, they have been linked to the rightmost path. For story S_9, it is an evolving story, which has the dependency pair (S_9, S_4). It first locates the parent node S_4. Since the left branch of S_4 is empty, story S_9 is added as the left child of S_4. When all stories have been processed, the topic structure binary tree is constructed and looks like that shown in Figure 5.

The topic structure offers several unique features to facilitate browsing and autodocumentary.

- *First story*: The first story of the topic is the left child of the root node, i.e., S_1.
- *Story dependency*: A story S_i being dependent on story S_j implies that among the previously delivered $i-1$ news stories, story S_i is most related to S_j. For example, the news story S_{19} is most related to story S_{17} among the previously delivered 18 news stories.
- *Novel story*: The first story and the nodes in its rightmost path are novel stories. For example, the first story and the stories with shallow background color in Figure 5 are novel stories.
- *Peripheral story*: The nodes in the rightmost path without the left child are peripheral (isolated) stories. In Figure 5, stories S_2, S_3, S_5–S_7, S_{11}, S_{13}, S_{16}, S_{18}, S_{23}, S_{28}, and S_{30} are peripheral stories. A peripheral node indicates that there is no news story similar to it, and this news story has no further development, such as node S_{23}.
- *Redundant story*: The nodes that have edges with high redundancy scores are redundant stories.
- *Evolving story*: Stories that do not belong to either novel or redundant stories are evolving stories.



[FIG6] The story distribution based on the cosine distance.

Through this topic structure, viewers can easily find the dependencies among these news stories, which can facilitate content summarization.

EXPERIMENTS

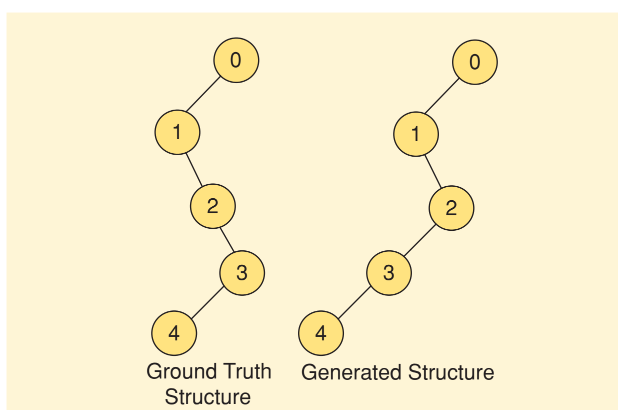
PERFORMANCE OF NOVELTY/REDUNDANCY DETECTION

We collected story distributions based on the redundancy score calculated by the redundancy measure proposed in the previous section. Figure 6 gives an example of story distributions of the Cosine Distance measure of textual concepts (WORD), visual concepts (KF) and the combination of textual and visual concepts (WORD + KF). From Figure 6(a), we can see that the evolving, redundant, and novel stories interleave together. There are no obvious decision boundaries among them. In Figure 6(b), the evolving stories are almost evenly scattered across a wide range. However, the story distribution in Figure 6(c) is obviously superior to the previous two cases. The novel, evolving and redundant stories are mostly dispersed without serious overlap, which indicates that the measure integrating textual and visual concepts can better identify the stories than the measure using either textual or visual concepts.

Generally speaking, choosing a proper threshold is a difficult problem. Improper thresholds can lead to drastic performance variation. In Figure 6(a), it is difficult to find the proper thresholds to achieve good performance. However, in Figure 6(c), the boundary is easier to identify, which can help in proper threshold selection.

EVALUATION OF TOPIC STRUCTURE

To analyze the performance of the novelty/redundancy detection scheme, three untrained nonexpert assessors were asked to watch our test stories and to judge one topic at a time. News stories in each topic were ordered chronologically. The assessors were requested to label the news stories with a judgment (redundant, evolving, or novel) and to identify the dependencies of news stories. The results of the assessors provided us with a "ground truth" with which we could compare our own results. In this article, we assume that stories of the same topic and the set of NDK are given.



[FIG7] An example of the ground truth topic structure and the generated topic structure.

[TABLE2] THE PERFORMANCE COMPARISON OF THE STORY DEPENDENCY.

ID	TOPIC	#	COSINE WORD	COSINE KF	COSINE WORD + KF	SET DIFFERENCE WORD	SET DIFFERENCE KF	SET DIFFERENCE WORD + KF
1	YELTSIN	6	1	1	1	0.5	0.833	0.833
2	CLINTON	19	0.842	0.789	0.842	0.684	0.789	0.737
3	SCANDAL	7	1	0.714	0.857	0.714	0.857	0.714
4	ARKANSAS	33	0.545	0.818	0.939	0.485	0.818	0.606
5	IRAQ	5	1	1	1	1	1	1
6	HOSPITAL	5	0.6	0.8	1	0.4	0.8	0.6
	AVERAGE		0.831	0.854	0.94	0.631	0.85	0.748
	Σ(DM * #)/Σ#		0.733	0.826	0.92	0.587	0.826	0.693

The performance of the topic structure is examined by comparing the generated dependency pairs and the ground truth dependency pairs labeled manually. An algorithm which has a better dependency pairs of stories should generate a topic structure that is closer to human perception, (i.e., the ground truth).

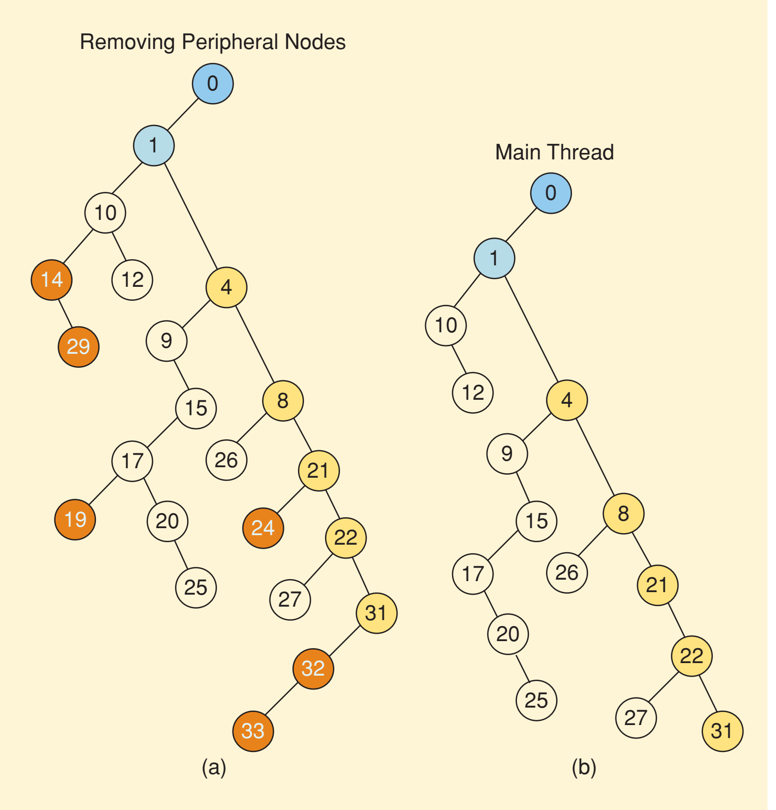
We define dependency matching (DM) to evaluate the performance between the generated dependency pairs $D'(T)$ generated and the ground truth dependency pairs *D(T)* for a specific topic T, which is the number of correctly matched dependency pairs divided by the total number of dependency pairs:

$$\mathrm{DM} = |D(T) \cap D'(T)|/L,$$

where $L = |D(T)| = |D'(T)|$ is the number of stories in the topic. The value of DM ranges from zero to one. The higher the value of DM is, the better the performance of the topic structure.

Figure 7 shows an example of the ground truth topic structure and the generated topic structure for a sample topic. There are four news stories in the topic and the ground truth dependency pairs are (S_1, S_0), (S_2, S_0), (S_3, S_0) and (S_4, S_3). The generated topic structure has dependency pairs (S_1, S_0), (S_2, S_0), (S_3, S_2) and (S_4, S_3). The story S_3 should be novel, bu t this story is wrongly detected as an evolving story and is depe ndent on the story S_2. Other dependency pairs (S_1, S_0), (S_2, S_0), and (S_4, S_3) are correctly identified. So the value of DM is 0.75, i.e., (3 / 4).

Table 2 lists the performance comparison of the story dependency, in which the last row denotes the weighted average measure. From this table, we can see that the cosine distance with textual and visual concepts has the best performance and the DM value is comparatively high, which indicates that the topic structure obtained by this method is close to the real topic structure.



[FIG8] **Main thread generation for the "Arkansas school shooting."**

AUTODOCUMENTARY

An attractive application that can be developed, using threading, is to provide a documentary that is as compact as possible, which presents the most critical features required to summarize a topic. In this section, we present a method beneficial for autodocumentary by extracting the main thread of a topic structure. In a topic, there usually exist many redundant and peripheral news stories (as shown in Figure 5). To generate a documentary, these stories should be automatically detected and removed from the main thread topic structure. Redundant news stories have minor development and carry duplicate information to users and peripheral stories contain fresh information, but without further evolution. They usually depict isolated but not very important themes which can be pruned when other interesting and evolving themes exist.

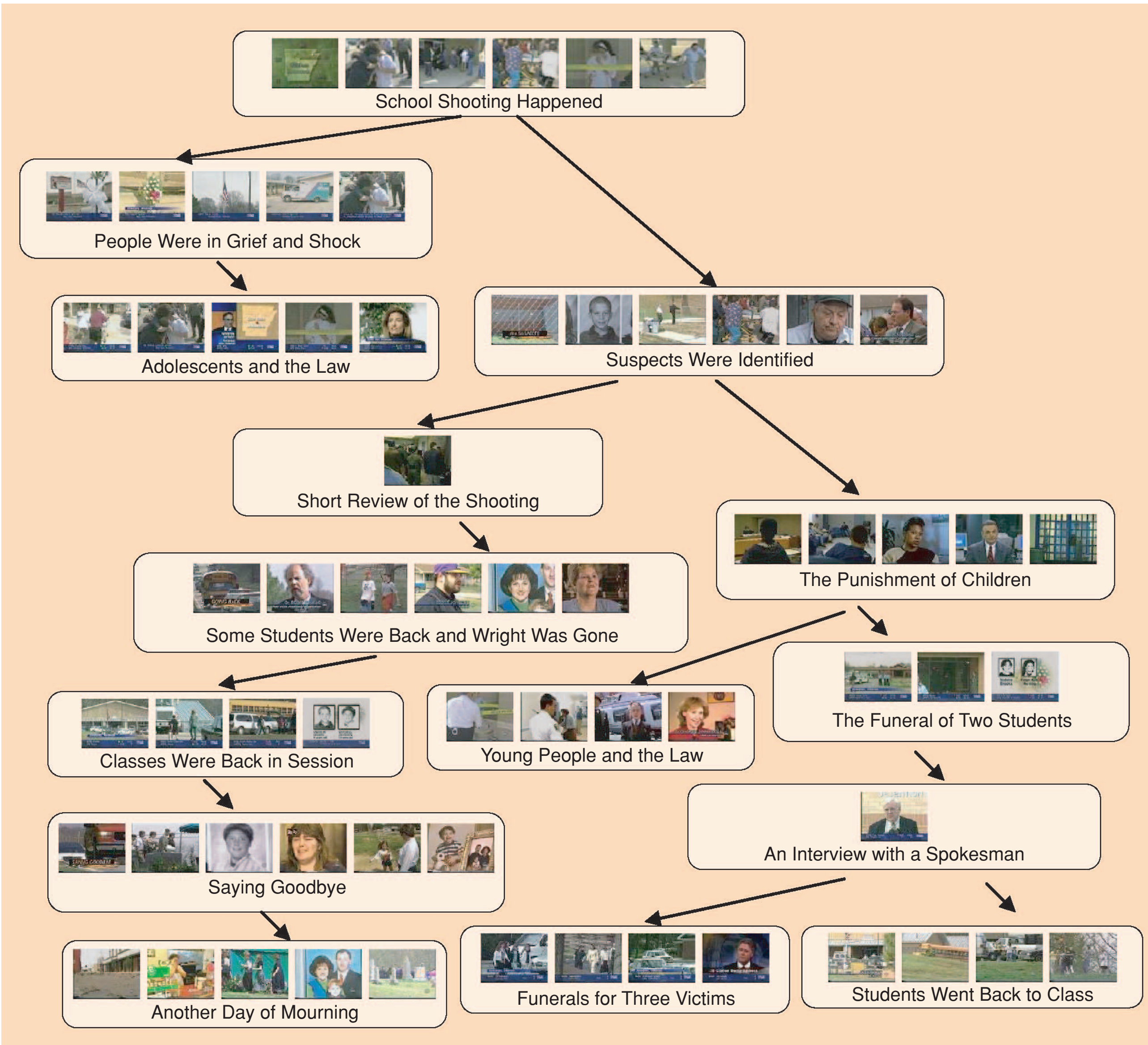
Given a topic structure, the main thread can be generated as follows:

- *Removing peripheral stories*: To give viewers a concise view of the main thread of a topic, we first remove the peripheral nodes in the topic structure tree.
- Removing redundant stories: We remove the redundant nodes to obtain a clear main

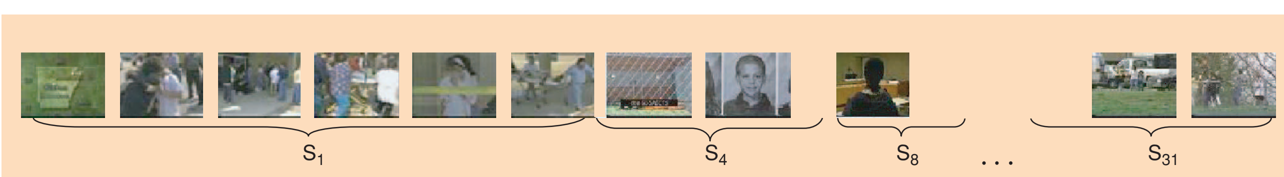
thread (which is used later for autodocumentary). Since redundant nodes can be inner nodes, removing them may cause to the reorganization of the topic structure. The structure can still keep the original properties after the reorganization since the dependencies between the parent and children have the transitive property.

Figure 8(a) shows the result for the topic structure in Figure 5, after removal of the peripheral nodes. Figure 8(b) shows the result after also removing the redundant nodes. Figure 9 shows a more informative graphical view of the main thread of the "Arkansas School Shooting."

Once these steps have been done, and a main thread has been created, a news summary can then be generated based on the depth-first, breadth-first traversal or time-order traversal of the main thread to assemble the autodocumentary. Figure 10



[FIG9] A graphical view of the main thread of the "Arkansas school shooting."



[FIG10] The final documentary based on the time-order traversal of the "Arkansas school shooting."

shows the final documentary based on the time-order traversal of the "Arkansas School Shooting."

CONCLUSIONS AND FUTURE WORK

We have presented a coclustering approach in integrating textual and visual concepts for grouping time-evolving news stories. Based on a redundancy measure, a topic structure binary tree can be modeled to represent the dependencies among news stories, which provides several unique features to facilitate browsing and autodocumentary. This allows the main thread to be further extracted to present the highlights of the topic. News summaries can then be directly generated by assembling the news stories in the main thread.

Our method proves to be quite encouraging, yet more work still remains to be done. For example, further analysis on how many clusters to be selected needs to be done, along with investigation into using other information (e.g., audio and motion), to improve the results. Although redundant stories (i.e., interstory redundancy) have been removed, there exist redundant shots and sentences (i.e., intrastory redundancy) in the summary. News video editing optimization is a topic that can be explored to reduce the intrastory redundancy of the main thread, while still preserving the coherence of audio and visual modalities.

In brief, with the overwhelming volume of news videos available today, it becomes challenging to track the development of news stories from different channels, identify their dependencies, and organize them in a semantic way. Topic threading and autodocumentary of news stories appears as one promising solution to help users rapidly browse and easily track the evolution of news topics.

ACKNOWLEDGMENTS

The work described in this article was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 1072/02E) and a grant from City University of Hong Kong (Project No. 7001804).

AUTHORS

Xiao Wu received the B.Eng. and M.S. degrees in computer science from Yunnan University, Kunming, China, in 1999 and 2002, respectively. He is currently a Ph.D. candidate in the Department of Computer Science at the City University of Hong Kong. He was with the Institute of Software, Chinese Academy of Sciences, Beijing, between January 2001 and July 2002. From 2003 to 2004, he was with the Department of Computer Science of the City University of Hong Kong as a research assistant. His research interests include multimedia information retrieval, video computing and mobile computing.

Chong-Wah Ngo received his Ph.D. in computer science from Hong Kong University of Science and Technology, and his M.Sc. and B.Sc., both in computer engineering, from Nanyang Technological University of Singapore. Before joining City University of Hong Kong as assistant professor in Computer Science department in 2002, he was a postdoctoral scholar in Beckman Institute of University of Illinois in Urbana-Champaign. He was also a visiting researcher of Microsoft Research Asia in 2002. His research interest is in video computing, multimedia information retrieval, data mining, and pattern recognition.

Qing Li holds a Ph.D. degree in computer science from the University of Southern California. Before joining City University of Hong Kong in 1998, He taught at the Hong Kong Polytechnic University, the Hong Kong University of Science and Technology, and the Australian National University. His research areas are multimedia and mobile data management, data warehousing and mining, workflow management, and Web services. He has published over 190 papers in refereed journals and conference proceedings in these areas. He has been an organizer/coorganizer and as a program committee member for numerous international conferences. He currently serves on the editorial boards of *World Wide Web* and *The Journal of Web Engineering*. He is a Senior Member of the IEEE and chair of the Hong Kong Web Society. In addition, he is a councilor of the Database Society of Chinese Computer Federation and a Steering Committee member of DASFAA, ICWL, and the international WISE Society.

REFERENCES

[1] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proc. ACM SIGIR'03*, pp. 314–321.

[2] D.Q. Zhang, and S.F. Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *Proc. ACM Multimedia Conf.*, Oct. 2004, pp. 877–884.

[3] P. Duygulu, J.-Y. Pan, and D.A. Forsyth, "Towards auto-documentary: tracking the evolution of news stories," in *Proc. ACM Multimedia Conf.*, Oct. 2004, pp. 820–827.

[4] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Multimedia Conf.*, Oct. 2004, pp. 869–876.

[5] J. Allan, Ed *Topic Detection and Tracking: Event-Based Information Organization*. Norwell, MA: Kluwer, 2002.

[6] I.S. Dhillon, "Coclustering documents and words using bipartite spectral graph partitioning," in *Proc. ACM SIGKDD'01*, pp. 269–274.

[7] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event threading within news topics," in *Proc. CIKM'04*, USA, Nov. 2004, pp. 446–453.

[8] D.-Q. Zhang, C.-Y. Lin, S.-F. Chang, and J.R. Smith, "Semantic video clustering across sources using bipartite spectral clustering," in *Proc. Int. Conf. Multimedia Expo*, Jun. 2004, , pp. 117–120.

[9] Y. Zhang, J. Callan, and T. Minka, "Novelty and redundancy detection in adaptive filtering," in *Proc. ACM SIGIR'02,* pp. 81–88.

[10] X. Wu, C.-W. Ngo, and Q. Li, "Co-clustering of time-evolving story with transcript and keyframe," in *Proc. Int. Conf. Multimedia and Expo*, 2005, pp. 117–120.

[11] TRECVID, 2004 [Online]. Available: http://www-nlpir.nist.gov/projects/trecvid/

[12] J.L. Gauvain, L. Lamel, and G. Adda, "The LIMSI broadcast news transcription system," *Speech Commun.*, vol. 37 no. 1-2, pp. 89–108, 2002.

[13] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques" in *Proc. KDD Workshop on Text Mining, 2000*.

[14] I. Ide, H. Mo, and N. Katayama, "Threading news video topics," in *Proc. Int. Workshop Multimedia Information Retrival*, USA, 2003, pp. 239–246.

[15] X. Zhu, J. Fan, A.K. Elmagarmid, and X. Wu, "Hierarchical video content description and summarization using unified semantic and visual similarity," *Multimedia Syst.*, vol 9, no. 1, 2003, pp. 31-53.

[16] E. Gabrilovich, S. Dumais, and E. Horvitz, "Newsjunkie: Providing personalized newsfeeds via analysis of information novelty," in *Proc. WWW'04*, USA, 2004, pp. 482–490.

[17] Q. Ma, A. Nadamoto, and K. Tanaka, "Complementary information retrieval for cross-media news content," in *Proc. MMDB'04*, USA, Nov. 2004, pp. 45–54.

[18] S. Morinaga and K. Yamanishi, "Tracking dynamics of topic trends using a finite mixture modal," in *Proc. ACM SIGKDD 04,* USA, Aug. 2004, pp. 811–816.

[19] G. Salton, *Automatic Text Processing*. Reading, MA: Addison-Wesley, 1989.