Towards Auto-Documentary: Tracking the Evolution of News Stories*

Pinar Duygulu
CS Department
University of Bilkent, Turkey
duygulu@cs.bilkent.edu.tr

Jia-Yu Pan
CS Department
Carnegie Mellon University
jiayu@andrew.cmu.edu

David A. Forsyth
EECS Division
UC Berkeley, U.S.A.
daf@cs.berkeley.edu

ABSTRACT

News videos constitute an important source of information for tracking and documenting important events. In these videos, news stories are often accompanied by short video shots that tend to be repeated during the course of the event. Automatic detection of such repetitions is essential for creating auto-documentaries, for alleviating the limitation of traditional textual topic detection methods. In this paper, we propose novel methods for detecting and tracking the evolution of news over time. The proposed method exploits both visual cues and textual information to summarize evolving news stories. Experiments are carried on the TREC-VID data set consisting of 120 hours of news videos from two different channels.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene understanding—*Video Analysis*

General Terms

Algorithms, Experimentation

Keywords

News video analysis, auto-documentary, duplicate sequences, matching logos, graph-based multi-modal topic discovery

1. INTRODUCTION

News videos constitute an important source of information for tracking and documenting important events [1]. These videos record the evolution of a news story in time and contain valuable information for creating documentaries. Automated tracking of the evolution of a news story over the course of an event can help summarize the event into a documentary, and facilitate indexing and retrieval. The final results are useful in areas such as education and media production.

Most previous works consider the problem of event characterization on the text domain. However, for our problem of identifying and tracking stories in news videos, we have richer information than text streams. We would like to incorporate both visual and textual information to generate a more informative event summary.

In news videos, stories are often accompanied by short video sequences that tend to be used again and again during the course of the event. A particular video sequence can be re-used with some modifications either as a reminder of the story or due to a lack of video material for the current footage.

Human experts suggest that there are two common conventions are frequently used on news video productions: (a) the re-use of a particular shot sequence to remind a particular news event; and (b) showing a similar, if not the same, graphical icons as the symbol of a news event. We call the repeating shot sequences in news stories as *threads*. We also define *logos* as the graphical icons shown next to the anchor-person in news reports.

The tendency of news channels to re-use the same video sequences can be used to track news events. In this study, we propose an algorithm to detect and track news events by finding the *duplicate* video sequences and identifying the *matching logos*.

Furthermore, we propose a method of finding event topics according to both the visual cues from shot keyframes and textual information from shot transcripts. Topics found are then used for better event summarization. The observation is that, as the event evolves, more evidences are known and the materials presented in news stories change. This change could be changes of key terms in transcripts, as well as changes of visual cues (major players of the event change resulting in a change of the face information).

Particularly, we are interested in the following questions: *Which visual cues are effective for tracking news stories? How do we extract these visual cues automatically? How do we make smart use of the multi-modal (visual and textual) information in video clips?* Our experiments on the TREC-VID data sets give successful results on tracking news *threads*, which are the repetitive keyframe sequences, and *matching logos*. Event topics are identified automatically using both visual and textual information. The event of a

*This work is supported by the National Science Foundation under Cooperative Agreement No. IIS-0121641 and IIS-0205224, and the Advanced Research and Development Activity (ARDA) under contract number H98230-04-C-0406 and NBCHC040037.

MM'04, October 10-16, 2004, New York, New York, USA.

thread or a logo is characterized by *topics*, which is more robust than summarization by words co-occurring with shots of the thread or logo.

The paper is organized as follows: The next section describes the data set and features used in our study. We present the method for detecting duplicate video sequences in Section 3. Section 4 describes the proposed approach for automatic detection of repeating news stories. Logo images used by the channels to mark news stories are used as an alternative approach for tracking news stories as will be explained in Section 5. Section 6 presents results on how the topic clusters created from news transcripts can be used to compare the results obtained from the detection of duplicate video sequences. Finally, we conclude in Section 7 and discuss future lines of research.

2. DATA SET

In this study, the experiments are carried out on the data set provided by the content-based video retrieval track (TREC-VID) of the 2003 Text Retrieval Conference TREC [3]. The data set consists of 120 hours of broadcast news videos (241 thirty minutes programs) from ABC World News Tonight and CNN Headline News recorded by the Linguistic Data Consortium from late January through June 1998.

The common shot boundaries, defined by TREC-VID, are used as the basic units. One keyframe is extracted from each shot. In total, there are 43752 and 38346 shots from ABC and CNN videos, respectively. Each keyframe is described by a set of features. The average and standard deviation of HSV values obtained from a 5×5 grid (150 features) are used as the color features. The mean values of twelve oriented energy filters (aligned uniformly with 30 degree separation) extracted from a 3×3 grid (108 features) represent the texture information. Canny's edge detector is used to extract 71 edge features from a 3×3 grid. Schneiderman's face detector algorithm [4] is used to detect frontal faces. The size and position of the largest face are used as the face features (3 features). All the features are normalized to have zero mean and unit variance.

3. DETECTING DUPLICATE SEQUENCES

Every time a piece of video is re-used, it may be slightly modified and the segmentation algorithm may partition it into different number of shots. Also, the keyframes selected from these shots may differ. Therefore, the same piece of video story may look like two different sequences. We define *duplicate sequences* as a pair of video sequences that share identical or very similar consecutive keyframes.

DEFINITION 1. ***(Duplicate sequence)*** *We denote a duplicate sequence as* $\{(s_1, \ldots, s_m), (t_1, \ldots, t_n)\}$*, where* s_i*'s are the shots of the first component, and* t_j*'s are those of the second.*

The sequences are allowed to have extra keyframes inserted, that is, a near-perfect match among occurrences of the duplicate sequence is sufficient. The relaxation on matching is to allow possible production variations.

In Figure 1, two *duplicate sequences* are shown. The lengths (number of shots) of the matching pair of sequences can be different due to the missing shots in one of the sequences as in (a). Similarly, the shots may be different as in (b), even though the sequences have the same length.

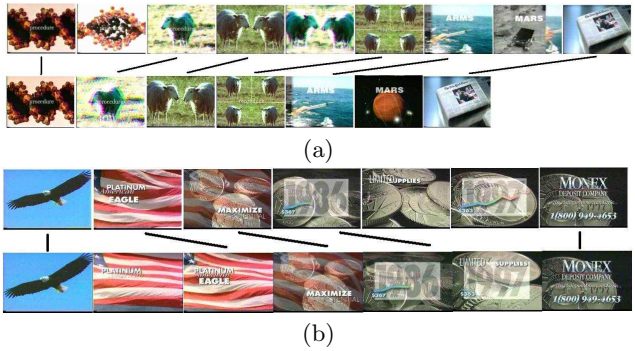


Figure 1: Examples of duplicate sequences. In (a) the 2^{nd} and 8^{th} keyframes of the top sequence are missing in the bottom sequence. In (b), the lengths of the sequences are same, but there are missing keyframes in both of the sequences. The keyframes are not always identical, e.g., the first and the second matching shots in (a).

In [5], visual features extracted from I-frames are used to detect repeating news videos. However, due to large amount of data, using I-frames is not feasible and this system works only for detecting identical video segments. Naphade and Huang [13] propose a HMM based method to detect the recurrent events in videos. Their model is mostly for finding the very frequent events which, in our case, may correspond to commercials among news stories and need to be removed.

In the following subsections, we will first explain the method to find *candidate repeating keyframes* (***CRKF***) by searching the identical or very similar keyframes using the feature similarity. Then, we describe a method to find the *duplicate sequences*.

We note that duplicate sequences are not all of news content. Commercials are also examples of duplicate sequences. To find news-related duplicate sequences, commercials are filtered out using our previously proposed method [12].

3.1 Finding *CRKF*s

Candidate repeating keyframes (*CRKF*s) are defined as the keyframes that have identical or very similar matching keyframes. In [6], similar news photographs are identified using iconic matching method which is adapted from [7]. However, in our case, there may be bigger differences between similar keyframes that may cause problems in iconic matching method (e.g., the text overlays, or large modifications due to montaging process). Therefore, we propose a method which can identify similar but not necessarily identical keyframes.

A candidate keyframe is defined to have a few duplicates or very similar images, and differ largely with the others. To detect this property, for each image in the data set, we find the most similar N images (Euclidean distance between feature vectors). There are 120 news videos in each of ABC and CNN data sets. We assume that a meaningful shot sequence (and nor will its keyframes) will not appear in all videos and choose N as 50.

To figure out the true nearest neighbors of a keyframe, we inspect the distances of the N=50 neighbors. Figure 2 shows the distances to the N=50 neighbor frames of some selected keyframes. If a frame reoccurs k times, then there

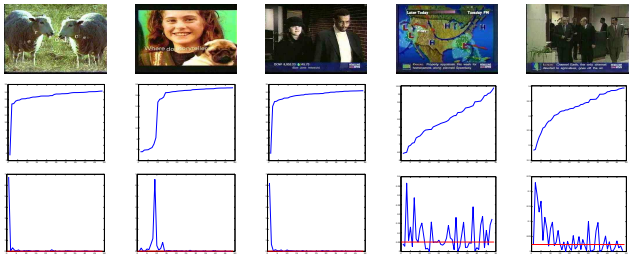


Figure 2: (Top) Keyframe images. (Middle) Distances to most similar 50 images. (Bottom) Derivatives. The horizontal lines (in red color) in the derivative figures are the derivative medians. A big jump in the diagram signifies a candidate frame for news: (a),(c) have only one duplicate, where (b) has 8 similar keyframes. Not chosen as candidates: (d) keyframe reoccurs too frequently. (e) a keyframe which does not have duplicates.

Definitions:
C: set of candidate repeating keyframes
$similar(c)$: set of similar keyframes of c
M: maximal length to look ahead for the next match
$seq(c, c')$: set of keyframes between keyframes c and c'
S, S': components of the found duplicate sequences
Algorithm:
for all $c_1 \in C$
  for all $c'_1 \in similar(c_1)$
    $S = \{c_1\}$; $S' = \{c'_1\}$
    $i = 1$
    /* look ahead sequentially */
    $\forall(c, c')$, $dist(c_i, c) \leq M$ if $c \in similar(c')$
      $c_{i+1} = c$; $c'_{i+1} = c'$;
      $S = S \cup \{seq(c_i, c_{i+1})\} \cup \{c_{i+1}\}$
      $S' = S' \cup \{seq(c'_i, c'_{i+1})\} \cup \{c'_{i+1}\}$
      $i = i + 1$; break;

Figure 3: Algorithm for detecting duplicate sequences.

would be a clear jump on similarity distance between the k and $k+1$ neighbors. In (a) and (c), the jump happens at k=1, indicates that the keyframes do not repeat. On the other hand, the keyframe in (b) repeats 8 times (jump at k=8). The keyframe shown in (d) is a common scene for weather news and repeats in almost all news programs. It is too frequent (there are more than 50 very similar images) and there is no obvious jump. Similarly, the keyframe in (e) is from a regular news story which doesn't have obvious jump either.

Intuitively, the jump shows that the keyframe in question has a well-formed cluster of similar keyframes, showing the keyframe is used repeatedly. The keyframes of Figure 2 (a)-(c) are defined as *CRKF*s, since they all have significant jumps in the diagrams.

To automatically detect a jump in keyframe similarity, we examine the first derivative of the similarity distances (Figure 2, bottom part), where a jump will cause a big derivative value. A jump is recognized if the ratio between a the largest derivative value and median value is larger than a threshold (for our experiments, the threshold is chosen as 100). This process chooses the images in Figures(a)-(c) as *CRKF*s.

3.2 From *CRKF*s to duplicate sequences

Due to news productions and keyframe selection, repeating video scenes do not necessary have identical sequence of keyframes. Certain keyframes will be inserted or deleted as the news event evolves, and keyframes in a sequence may not have matching counterparts. To find the entire sequence which covers a news story properly and prevent from being cut short, we need to allow gaps within matching sequences.

To detect matching sequences with gaps, *CRKF*s and their neighbors are used as the starting point. We said that a frame A *matches* another frame B, if A is a neighbor of B. A pair of possible matching sequences always starts from a pair of *CRKF*s. The matching sequences expand continuously by examining the next M keyframes following the starting keyframes to find the next matching keyframes. If a pair of such matching frames is found among the following M frames, the matching sequences are extended by inserting these two matching frames. The keyframes skipped during the expansion are also inserted into the sequences. This process repeats itself until no matching pairs within next M frames are found. This is performed for each candidate keyframe in the data set. The algorithm is given in Figure 3

Shorter footages, such as the teaser at the beginning of a news movie, or the preview in front of each commercial break, lack content and do not contain a lot of information To eliminate these sequences, only the matching sequences which have length longer than a threshold are chosen as *duplicate sequences*.

3.3 Detecting and removing commercials

In news videos, commercials are often mixed with news stories. For efficient retrieval and browsing of the news stories, detection and removal of commercials are essential [8, 9, 10, 11]. It is common to use black frames to detect commercials. However, such simple approaches will fail for videos from TV channels that do not use black frames to flag commercial breaks. Also, black frames used in other parts of the broadcast will cause false alarms. Furthermore, progress in digital technology obviates the need to insert black frames before commercials during production. An alternative makes use of shorter average shot lengths as in [10]. However, this approach depends strongly on the "high activity" rate which may not always distinguish commercials from regular broadcasts.

In this study, we detect and remove commercials using a combination of two methods that use distinctive characteristics of commercials [12]. In the first method, we exploit the fact that commercials tend to appear multiple times during various broadcasts. This observation suggests us to detect commercials as sequences that have duplicates. Commercials have longer sequences because of the rapid shot-breaks within. We use this fact to separate them from other duplicate sequences. The second method utilizes the fact that commercials also have distinctive color and audio characteristics. We note that the second method implicitly includes the idea of "black frame" detection.

Because the two methods capture different distinctive characteristics of commercials, they are orthogonal and complementary to each other. Therefore, combination of the two

CNN headline news 03/27/1998

Russian president boris yeltsin has nominated a new prime minister He announced today he wants acting prime minster sergei kiriyenko to take over the post permanently. The russian parliament's lower house now has one week vote on the nomination. Yeltsin is threatening to disband the duma if it doesn't approve the 35-year-old kiriyenko Yeltsin dismissed his entire cabinet monday without warning

CNN headline news 03/27/1998

And russian president boris yeltsin nominated acting prime minister sergei kiriyenko to take over the post permanently. Yeltsin is threatening to disband parliament if lawmakers don't approve his choice

Figure 4: A news story (top) and its preview (bottom).

methods yields even more accurate results. Experiments show over 90% recall and precision on a test set of 5 hours of ABC and CNN broadcast news data.

4. TRACING NEWS STORIES: "THREADS"

The evolution of news stories can be tracked by finding the repeating news video scenes. We represent a scene as a sequence of keyframes. and observe two production effects on repeating news scenes. First, parts of the scenes for important events are collected and shown as preview at the beginning of a program (e.g., Figure 4). Second, and more interestingly, the same video scene will be re-used in related news stories that continue over a period (e.g., Figure 5). Tracking those re-used sequences could provide meaningful summaries, as well as more effective retrieval where related stories can be extracted all at once.

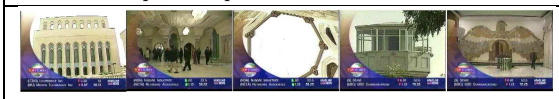
We call the repeating news scenes **threads**. Similar to commercials, we define threads as a subclass of duplicate sequences. That is, a thread is a duplicate sequence which is (a) *not commercial* and (b) *at least 20 keyframes apart between its components*. In our data set, 907 sequences in CNN and 430 sequences in ABC are detected as thread components.

The histogram of thread component lengths (ranging from 1 to 7) is shown in Figure 6. CNN tends to have longer thread components than ABC. Having a large amount of single-frame thread component in ABC may due to: (a) it commonly re-use only a small part of previous material, or (b) the order of the sequences are changed when being re-used.

The separation between thread components varies from 1 to 25000 shots. The average number of shots in for an half hour CNN news video is around 333. This means that thread components which are separated by more than 333 keyframes are shown in different days. Shorter separations usually correspond to previews (e.g., Figure 4), while larger ones correspond to stories which repeat on different days and are more interesting for our concern. Figure 5) shows a thread which is one week apart, whose thread component has length two (2 keyframes).

CNN headline news 02/04/1998

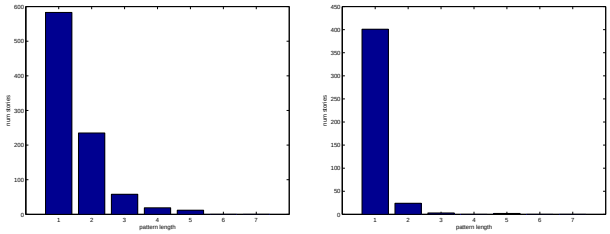
white house says time is running out for iraq to avoid military strike. administration officials are reacting coolly to baghdad's latest offer to open presidential Palaces to international weapons inspectors



CNN headline news 02/11/1998

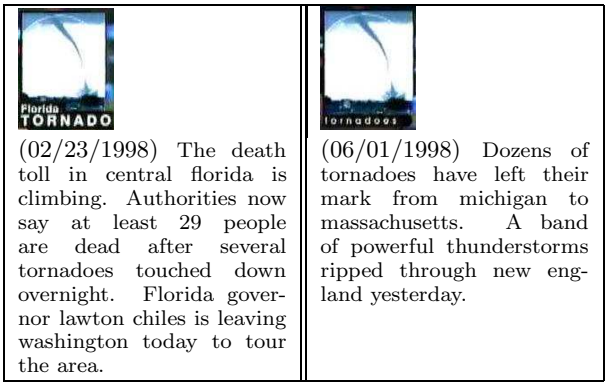
iraq is again offering to allow a limited number of u.n. weapons' inspectors into eighth presidential sites. The plan is giving inspectors two months to search the areas. The united states is demanding full access by u.n. weapons' inspectors to all sites.

Figure 5: Re-used news scene on different days.



(a) CNN (b) ABC

Figure 6: Lengths of the sequences that has duplicates.



(02/23/1998) The death toll in central florida is climbing. Authorities now say at least 29 people are dead after several tornadoes touched down overnight. Florida governor lawton chiles is leaving washington today to tour the area.

(06/01/1998) Dozens of tornadoes have left their mark from michigan to massachusetts. A band of powerful thunderstorms ripped through new england yesterday.

Figure 7: Similar logos are used on different days to present stories about tornadoes.

5. LINKING NEWS BY LOGOS

Another helpful visual cue for finding related news stories is the re-use of *logos* - the small graphics or picture that appears behind the anchor person on the screen. The same logo is repeatedly used to link related stories and show the evolution of a story. Figure 7 shows a logo which is used in different news stories about tornadoes on different days. We are especially interested in finding the repeating logos which appear in programs on different days.

Figure 8: Anchor-logo frames. (First two rows) correct detection results. (Last row) false positives.

We make use of the iconic matching method [6, 7] to find matching logo sequences. We perform iconic matching only on the *anchor-logo frames* in the news reports. *Anchor-logo frames* are the frames that have both the anchor person and a logo side-by-side. In our experiments, we use only the CNN news whose logos appears at the right of the anchor person. Regions in anchor-logo frames which correspond to logos are then cropped and fed to the iconic matching process to find matching logos.

5.1 Detecting Anchor-Logo Frames

To detect the anchor-logo frames, we first prepare a training set which has 354 frames with logo (labeled manually) as positive examples, and 1000 frames without logos (chosen randomly) as negative examples. We then build a nearest neighbor classifier to find the anchor-logo frames in a test set. The test set is consisted of 44 anchor-logo images and 935 images without logos. All 44 anchor-logo frames are detected correctly as logo images and 917 images are detected correctly as non-logo images. Overall, over 98% accuracy is obtained in detecting the the anchor-logo frames.

Figure 8 shows some of the images detected as anchor-logo frames. We note that the nearest neighbor classifier can be easily built with high accuracy for video data of a previously unseen channel. This is due to the observation that a news channel always produce similar anchor-logo frames of one particular look, which makes such a simple classifier sufficient to identify them accurately.

5.2 Identify Repeating Logos: Iconic Matching

After having a set of *anchor-logo frames*, logos are cut-off from the predefined upper-right corner of these frames. The logos are re-sampled to the size of 128-by-128 to facilitate the iconic matching steps given in [6]. From each logo, we compute 3 sets of the 2-Dimensional Haar coefficients, one for each of the RGB channels. The RGB values are in the interval [0,255]. We select 64 coefficients which located at the upper-left corner of the transform domain as features and form the feature vector of a logo. The selected coefficients are the overall averages and the low frequency coefficients of the three channels.

Finding repeating logos is a similarity search based on the feature vectors of the logos. We consider two logos are matched (hence the logo repeats), if more than 40 coefficients in their feature vectors have differences smaller than some thresholds (5 for the first three overall averages, and 3 for the rest of the coefficients).

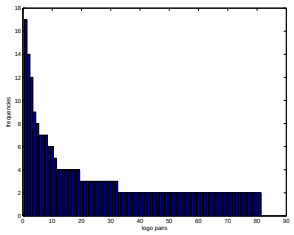


Figure 9: Repeat frequency of logos.

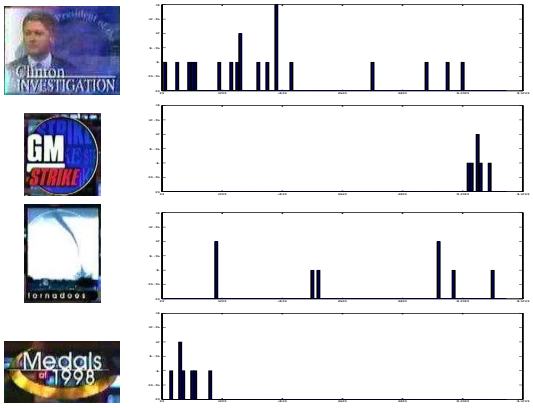


Figure 10: Time spans of the selected logos. Some events span shortly (e.g., `GM strike` or `Medals`), while some have longer periods (e.g., `Clinton investigation`).

For our data set, 660 images are predicted as *anchor-logo frames*, of which 267 images have repeating logos. The number of distinct logos is 81. Figure 9 shows the histogram of the repeat frequencies. Most logos repeat only once, while three logos repeat over 10 times. Each repeating logo usually corresponds to footages about the evolution of a particular news story.

The time period between the re-use of a logo is different for different stories. A news story, such as the `Clinton Investigation`, may span a long period, while it could important only for a few days as the stories `GM strike` and `Medals` (Figure 10).

6. AUTOMATIC EVENT SUMMARIZATION

After we found the news threads and logos, we would like to summarize them automatically. The straightforward way to come out with a summary is to take the transcripts of all thread shots and process the transcript words using some textual techniques. However, the pure textual method may overlook the interactions between visual and textual information, i.e., the visual content determines 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 shot is selected. *Can we develop an method which consider both visual and textual information at the same time for summarizing stories related to threads and logos? How effective and consistent the method is?* Using visual information may help generate better summary by linking additional information. For example, a frame of Kofi Annan

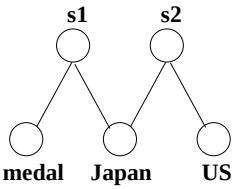


Figure 11: The graph shown is $G = (V_S \cup V_W, E)$, where the shot-nodes $V_S = \{s_1, s_2\}$ and the word-nodes $V_W = \{\texttt{medal}, \texttt{Japan}, \texttt{US}\}$. The shot s_1 is associated with the words `medal` and `Japan`, while s_2 is associated with the words `Japan` and `US`.

may appear in shots of a about United Nation as well as in some shots about Iraq, and therefore pulling in information on Annan's role on the Middle East situation, besides his role as the UN secretary-general.

6.1 Identify topics for event summary

We propose to summarize an event by the topics to which the event is related. By using topics rather than transcript words, we can achieve more robust summarization. As an example, for a thread about "Clinton investigation", we could successfully assign words like "`whitewater`", "`jones`", even these words do not appear in the transcripts of the associated thread shots.

We consider information from both keyframes and transcripts to discover topics. An evolving story may use certain words repeatedly in the transcripts of related footages, while the keyframes of these footages may differ. For example, the many shots of the Winter Olympic Games may have different keyframes, but words such as `medal`, `gold` and `olympic` may appear in all these shots. The situation may reverse, with the word usage gradually changes, while the keyframes stay intact. This happens usually when certain video scenes are presented as reminder to the previous development of a story. For example, the picture of President Clinton with Monica Lewinsky may appear again and again, even as the transcripts in the shots have changed to focus on the new findings from the investigation. By taking both visual and textual information into account, we hope to discover topics that better describe the news events.

We build a bipartite graph $G = (V, E)$, where the nodes $V = V_S \cup V_W$, the shot-nodes $V_S = \{s_1, \ldots, s_N\}$ is a set of nodes of shots, and the word-nodes $V_W = \{w_1, \ldots, w_M\}$ is a set of nodes of words in the vocabulary. (N is the total number of shots in the data set, and M is the size of the transcript vocabulary.) An edge (s_i, w_j) is included in the edge set E, if the word w_j appears in the transcript of the shot s_i. For example, if the data set has N=2 shots, where the first shot is about 1998 Nagoya Winter Olympic Games with words `medal` and `Japan`, and the second is about economy with words `Japan` and `US`. The vocabulary is $\{\texttt{medal}, \texttt{Japan}, \texttt{US}\}$ (M=3). The corresponding graph G is shown in Figure 11.

We fix the number of topics K that we want to discover from the bipartite graph G and apply the spectral graph partitioning technique [14] to partition G into K subgraphs. The spectral technique partitions the graph such that each subgraph has greater internal association that external association. Each subgraph is considered to be a topic characterized by both the keyframes of the shot-nodes and the words belong to this subgraph. For example, the topic of "interest rate" may have keyframes of "Federal reserve" and transcript words like "`Washington`" and "`crisis`" (Figure 12). The topic label assigned to a shot is the label of the subgraph to which the shot belongs.

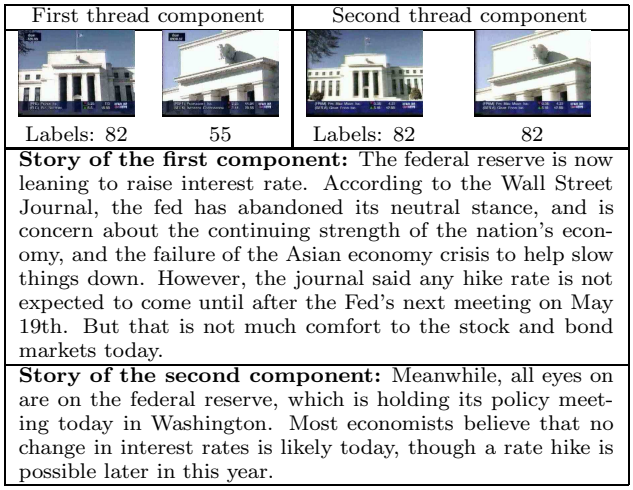


Figure 12: Topics assigned to the thread "Federal reserve on interest rate". Total number of topics is set at K=90.

To summarize a thread $T = \{(s_1, \ldots, s_m), (t_1, \ldots, t_n)\}$, where s_i's and t_j's are the shots of the two components, we first look up the topic labels of the shots and have the topic label sequences $\mathcal{C}(T) = \{(c_1, \ldots, c_m), (d_1, \ldots, d_n)\}$. Note that the labels c_i's and d_j's could duplicate, since two shots can have the same topic label. Let the most frequent label *shared* by the 2 thread components be e^*. We would summarize the thread T by the words of topic e^*.

Similarly, for a logo $L = (s_1, \ldots, s_m)$, where s_i's are the associated shots. We look up the topic labels of s_i's and have a sequence $\mathcal{C}(L) = (c_1, \ldots, c_m)$ of topic labels c_i's. Let the most frequent label in $\mathcal{C}(L)$ be c^*. We would describe the story of the logo L by the words of topic c^*.

Figure 12 shows the result on the thread "Federal Reserve's decision on interest rate". The words automatically chosen to describe this thread are "`income economy company price consumer bond reserve motor investment bank bathroom chrysler credit insurance cost steel communication airline telephone microsoft strength`" (from topic 82), which reflect the story content quite well.

Figure 13 shows the result on the logo "Clinton investigation". The words automatically chosen to describe this logo contains words form cluster 35, which includes the names of the major players involved such as "`monica`", "`lewinksy`", "`paula`" and "`starr`". Other words also reflect the story content very well. Other topics associated with this logo also have related words about the story, giving a hint that the entire story contains events of multiple aspects.

6.2 Measuring Coherence

We design a metric which we called *coherence* to measure the goodness of our summarization of a thread or a logo. In-

Figure 13: Logo "Clinton Investigation". The number of topics is set at $K = 50$. The most common topic is topic 35, which includes the following words: "brian monica lewinsky lawyer whitewater counsel jury investigation paula starr relationship reporter ginsburg deposition vernon affair oprah winfrey cattle source intern white deputy lindsey immunity aide adviser subject testimony subpoena courthouse privilege conversation mcdougal showdown turkey". Some words from other topics : topic 44 - "president clinton investigator scandal assault", topic 6 - "bill official campaign jones lawsuit", topic 12 - "court supreme document evidence".

tuitively, the coherence measures the degree of homogeneity of the topic labels assigned to a thread or a logo.

DEFINITION 2. ***(Logo topic coherence)*** *Let $L = (s_1, \ldots, s_m)$ be a logo associated with m shots (s_i's). The topic labels assigned to the shots in L are $\mathcal{C}(L)=(c_1, \ldots, c_m)$. Let c^* be the most frequent label in $\mathcal{C}(L)$. The logo topic coherence H_{logo} is defined as*

$$H_{logo} = \frac{\sum_{i=1}^{m} I(c_i == c^*)}{m},$$

where the function $I(p)$=1, when the predicate p is true, and $I(p)$=0, otherwise. Note that the range of H_{logo} is $[\frac{1}{m}, 1]$.

DEFINITION 3. ***(Thread topic coherence)*** *We consider the pairwise coherence between thread components. Let $T = \{(s_1, \ldots, s_m), (t_1, \ldots, t_n)\}$ be a thread consisting of two thread components of shots s_i's and t_j's. The topic labels assigned to the shots in T are $\mathcal{C}(T)=\{(c_1, \ldots, c_m), (d_1, \ldots, d_n)\}$. Let e^* be the most frequent label **shared** among labels c_i's and d_j's. The thread topic coherence H_{pair} is defined as*

$$H_{pair} = \frac{\sum_{i=1}^{m} I(c_i == e^*) + \sum_{i=1}^{n} I(d_i == e^*)}{n+m},$$

where the function $I(p)$=1, when the predicate p is true, and $I(p)$=0, otherwise. Note that the range of H_{pair} is $[0, 1]$. H_{pair}=0 when e^* does not exist.

Table 1 reports the average of the coherence values of all 81 logos we collected from the CNN set. The base value shown in Table 1 is overall mean logo coherence $\sum_{i=1}^{81} \frac{1}{m_i}$, where m_i is the number of shots of the i-th logo. The base value indicates the worst coherence the data set could get. The proposed method gives at least half ($H_{logo} > 0.5$, in average) of the shots in a logo the same topic label. The fact that logo shots share topic labels indicates that logos are indeed an useful handle to identify shots of the same story.

Table 1: (Logo topic coherence) The base coherence value is 0.398 (the worst possible coherence value). *Random avg* and *std* correspond to the mean and standard deviation of coherence values when topics are randomly assigned.

	K=50	K=60	K=70	K=80	K=90
H_{logo}	0.548	0.510	0.506	0.506	0.497
random (avg)	0.429	0.422	0.419	0.415	0.413
random (std)	0.008	0.010	0.008	0.007	0.005

Table 2: Thread topic coherences and thread component coherences.

	K=50	K=60	K=70	K=80	K=90
H_{pair}	0.122	0.100	0.112	0.132	0.090
H_{thread}	0.846	0.832	0.825	0.826	0.821

As expected, having K=50 topics gives the highest coherence, since it has the least diversity on labels. However, the coherence value remains stable as K increases, which is good, and indicates the performance would not decay much for any reasonable selected K.

We also compare the results with the coherence value assuming the topics are randomly assigned. The difference between H_{logo} value and that of random assignment is more than 3 times the standard deviation, showing that the topic assignment by the proposed method is statistically significantly better than random topic assignment.

Table 2 reports the average thread topic coherence of all 335 threads we collected from the CNN set. In the table, we also show the *thread component coherence* (denoted as H_{thread}), which is the coherence value of the shots in a thread component. H_{thread} is defined similarly as H_{logo}, where the thread component (a list of shots) is viewed as same as a logo shot-sequence (also a list of shots). The thread component coherence H_{thread} is above 82%, which indicates a great degree of coherence among shots in a thread component.

The proposed summarization method assigns the same topic label to shots associated with the pair of thread components only about one-tenth of the time ($H_{pair} \approx 0.1$). This shows that a great deal of difference exists in transcript words as an event evolves. This may due to our graph partitioning algorithm which provides a hard clustering among the words. However, as shown in Figure 13, although different topics are assigned, these topics are in fact reasonable, providing different viewpoints to the same story. We are currently extending our work to soft partitioning algorithm to try to improve the coherence degree and to achieve a more robust summarization.

7. DISCUSSIONS AND FUTURE WORK

The tendency to re-use the same video material allowed us to detect and track important news stories by detecting repeating visual patterns (duplicate video sequences and logos). The duplicate video sequences are detected with a heuristic pattern matching algorithm and same logos are detected using the iconic matching method.

Every time a piece of video is re-used, it may be slightly modified. For example, the re-used video could be cut shorter or have its frames re-ordered. The idea of duplicate sequences can deal with modifications such as cutting, but falls short to frame reordering. Instead of duplicate sequences, detection of duplicate "bag of keyframes" could solve such problems.

News threads and commercials are subclasses of duplicate sequences. To find the news threads, all possible duplicate sequences are examined and those of commercials or teasers/previews are filtered out.

Commercials are distinguished from the repeating news stories by the sequence length and whether the neighboring shots are commercial or not. Including the audio and transcripts will help to identify them better, since the audio and transcripts are also duplicated in commercials, which is not the case for news stories.

The evolution of news stories is important for creating documentaries automatically. With the proposed methods, it is possible to automatically track the stories with similar visual or semantic content inside a single TV channel. Same news story may also be presented in different channels in various forms with different visual and rhetoric styles. This may represent the perspectives of different TV channels, or even the perspectives of different regions or countries. Capturing the use of similar materials may provide valuable information to detect differences in production perspectives.

In this work, we only consider the association between shots and transcript words, and from which we found meaning topic clusters. By using multiple topic clusters, we can characterize the content of a news event (Figure 13). However, using multiple topics on characterizing news events limits the topic coherence of logos - outperforms the random topic assignment only by 0.1 coherence value (Table 1). We expect that by taking into account the similarity between the visual content of shots, as well as the similarity among the transcript words, we could find topic clusters which better describe the news events, and achieve larger improvement in the coherence metric over the random baseline. Although we show that the number of topic clusters, K, does not affect the coherence much (Table 1 and 2), being able to detect the right value of K is desirable and is left to the future work.

There has been much work on clustering text for finding topics, such as latent semantic indexing [15]. Most of them are pure textual methods. Our proposed method finds topics based on both visual and textual association. In the future, we would like to compare our result with the results from pure textual approaches, to gain deeper insights on how visual cues help find topics.

This is our first attempt to automatically generate event documentary. Many issues remain open, for example, how to determine the parameter values and what is the appropriate evaluation metric, just to name a few. We plan to address these problems in future work.

8. REFERENCES

[1] H. Wactlar, M. Christel, Y. Gong and A. Hauptmann, "Lessons Learned from the Creation and Deployment of a Terabyte Digital Video Library", IEEE Computer, vol. 32, no. 2, pp. 66-73, February 1999.

[2] Topic detection and tracking (TDT), http://www.nist.gov/speech/tests/tdt/

[3] TRECVID 2003, http://www-nlpir.nist.gov/projects/tv2003/

[4] H. Schneiderman, T. Kanade, "Object Detection Using the Statistics of Parts", International Journal of Computer Vision, 2002.

[5] F. Yamagishi, S. Satoh, T. Hamada, M. Sakauchi, "Identical Video Segment Detection for Large-Scale Broadcast Video Archives", International Workshop on Content-Based Multimedia Indexing (CBMI'03), pp. 135-142, Rennes, France, Sept. 22-24, 2003.

[6] J. Edwards, R. White, D. Forsyth, "Words and Pictures in the News", HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data, Edmonton, Canada, 31 May 2003.

[7] C. E. Jacobs, A. Finkelstein, D. H. Salesin, "Fast Multiresolution Image Querying", Proc. SIGGRAPH-95, pp. 277-285, 1995.

[8] R. Lienhart, C. Kuhmunch, W. Effelsberg, "On the detection and Recognition of Television Commercials", In proceedings of IEEE International Conference on Multimdeia Computing and Systems, 1997.

[9] A. Hauptmann, M. Witbrock, "Story Segmentation and Detection of Commercials in Broadcast News Video ", Advances in Digital Libraries Conference (ADL'98), Santa Barbara, CA, April 22 - 24, 1998

[10] S. Marlow, D. A. Sadlier, K. McGeough, N. O'Connor, N. Murphy, "Audio and Video Processing for Automatic TV Advertisement Detection", Proceedings of ISSC, 2001.

[11] L. Agnihotri, N. Dimitrova, T. McGee, S. Jeannin, D. Schaffer, J. Nesvadba, "Evolvable visual commercial detector", CVPR 2003.

[12] P. Duygulu, M.-Y. Chen, A. Hauptmann, "Comparison and Combination of Two Novel Commercial Detection Methods", Proceedings of the International Conference on Multimedia and Expo (ICME2004), Taipei, Taiwan, 2004.

[13] M.R. Naphade, T.S. Huang, "Discovering recurrent events in video using unsupervised methods", ICIP 2002.

[14] I. S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning", Proceedings of the Seventh ACM SIGKDD Conference, August 2001.

[15] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman, "Indexing by latent semantic analysis", Journal of the Society for Information Science, 41(6), 391-407.