MMSS: Graph-based Multi-modal Story-oriented Video Summarization and Retrieval[1]

Jia-Yu Pan, Hyungjeong Yang[2], Christos Faloutsos

August 2004

CMU-CS-04-114

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]Supported by the National Science Foundation (NSF) under Grants No. IIS-0121641, IIS-0083148, IIS-0113089, IIS-0209107, IIS-0205224, INT-0318547, SENSOR-0329549, EF-0331657, IIS-0326322, and by the Pennsylvania Infrastructure Technology Alliance (PITA) Grant No. 22-901-0001. Additional funding was provided by donations from Intel, and by a gift from Northrop-Grumman Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

[2]Supported by the Post-doctoral Fellowship Program of Korea Science and Engineering Foundation (KOSEF).

Keywords: story summarization, multi-modal video summarization, video retrieval, random walk, graph application

Abstract

We propose multi-modal story-oriented video summarization (*MMSS*) which, unlike previous works that use fine-tuned, domain-specific heuristics, provides a *domain-independent*, *graph-based* framework. *MMSS* uncovers correlations between information of different modalities and gives meaningful story-oriented news video summaries. *MMSS* can also be applied for video retrieval, achieving performance that matches the best traditional retrieval techniques (OKAPI and LSI), with no fine-tuned heuristics such as tf/idf.

1 Introduction and related works

As more and more video libraries [12] become available, video summarization is in great demands for accessing these video collections efficiently. Summarizing evolving news stories has broad applications ranging from media production (documentary) and education, to searching and indexing. Most previous work focuses on summarizing an *entire* video clip into a more compact movie, to facilitate browsing and content-based retrieval [11, 6]. For story-oriented summarization, research has been done mainly under the context of multi-document summarization [4] in the textual domain. Little work has been done on story-oriented video summarization using the multi-modal information in video clips.

Identifying footages of an evolving story from daily news programs is difficult. Broadcast news programs commonly shows a small icon beside an anchorperson to represent the story which the anchorperson is reporting at the time [2]. The same icon is usually reused later in the shots about the follow-up development of the story, as an aid for the viewers to link the current coverage with the past coverage. We call these icons "*logos*", and the associated stories "*logo stories*". The properties of logos make them a robust feature for linking separated footages of a story.

In this paper, we propose a method, *MMSS*, to generate multi-modal summary of a logo story. *MMSS* integrates multi-modal (visual/textual) information, treating it in an uniform, modality-independent fashion, with no need of parameter tuning. In fact, *MMSS* uncovers cross-modal correlations which gives not only good story summaries, but also video retrieval performance that matches the best finely tuned traditional information retrieval techniques.

The paper is organized as follows. Section 2 introduces the proposed method, *MMSS*. Sections 3 presents our experimental results on two applications, namely, story-oriented video summarization and video retrieval. Section 4 concludes the paper.

2 Proposed method: Video mining with *MMSS*

MMSS introduces a general framework for mining the cross-modal correlations among data of different modalities (frames/terms/logos) in video clips. The cross-modal correlations found by *MMSS* are then used for story-

winter olympics nagano ceremony night round competition result headline news superticker medal watch headline news superticker result

(a) "Winter Olympics"

money trie source lawyer house intern monica information counsel starr immunity learned starr

(b) "Lewinsky"

arab league secretarygeneral strike iraq reaction arab brent sadler report iraqi president saddam hussein kind sentiment attack

(c) "Iraq"

Figure 1: (News logos) Shots' keyframes which have logos and the transcript words associated with the shots.

oriented summarization and video retrieval.

The data set we used in this work is the TRECVID 2003 [10] data set. The data set is a collection of news programs. Each news program is segmented into shots, each of which is associated with a keyframe and a set of transcript words. For the words, we keep only the nouns and filter out the stop words.

A keyframe which contains a logo is called a "*logo frame*". Figure 1 shows the keyframes and the associated transcript words of three shots which have logo frames. In our experiments, logos are identified and extracted from the logo frames, using the off-the-shelf iconic matching algorithm [2, 3]. The iconic matching algorithm does not detect all the logo frames, due to the variations at the background of these frames. However, as we show later, the proposed *MMSS* method could identify the close relationship among the found logo frames and those that are missed, and successfully pulls up those missing ones.

Some news stories, such as "Winter Olympics", consist of footages that are loosely related, in terms of word usage and repeating scene occurrences. For example, shots of the "Winter Olympics", such as "speed skating" and "snow-boarding", may share only a few words or frames (one is an indoor sport, and the other is an outdoor one). Furthermore, as a story evolves, the usage of terms in the transcript changes. For such news stories, logos provide robust links to associate the shots of the same story.

Observation 1 *Logos provide robust visual hints and help track the shots of an evolving story.*

Our goal is to exploit the logos to facilitate video mining tasks. Particularly, we focus on the following two applications:

- (Story summarization) How do we generate high-quality textual and visual summaries of a logo story?
- (Video retrieval) How can we exploit the logos to retrieve the video clips that are relevant to a text query?

In addition, we want to perform the above two tasks in a principled way, that is, using the same framework for both tasks, integrating multi-modal sources easily, with no parameter tuning.

In the following subsections, we first describe our proposed method "*MMSS*", which is a graph-based method, using the versatile tool - *random walk with restarts*. Following that, we briefly review two of the best traditional textual retrieval techniques. As we show later, our proposed method achieves comparative (sometimes is even better) result on video retrieval, comparing with the two textual retrieval methods.

2.1 Graph G_{MMSS}

We integrate the information of shot-word co-occurrence and the logo information into a graph G_{MMSS}. The graph G_{MMSS} is a three-layer graph with 3 types of nodes and 2 types of edges. The 3 types of nodes are the *logo-node*, the *frame-node* and the *term-node*, each corresponds to a logo, a keyframe (shot), or a term, respectively. The 2 types of edges are the *term-occurrence edge* and the "*same-logo*" *edge*.

Figure 2 shows an example graph G_{MMSS} with 2 logo-nodes $\{l_1, l_2\}$, 5 frame-nodes $\{f_1, \ldots, f_5\}$, and 10 term-nodes $\{t_1, \ldots, t_{10}\}$. The term-occurrence edges are the solid lines, and the "same-logo" edges are the dotted lines. Let $O(n)$ be the corresponding object of a node n. For example, $O(l_i)$ is the corresponding logo of the logo-node l_i.

A logo-node l_i is connected to a frame-node f_j by a "same-logo" edge, if the logo $O(l_i)$ appears in the frame $O(f_j)$. A frame-node f_j is connected to

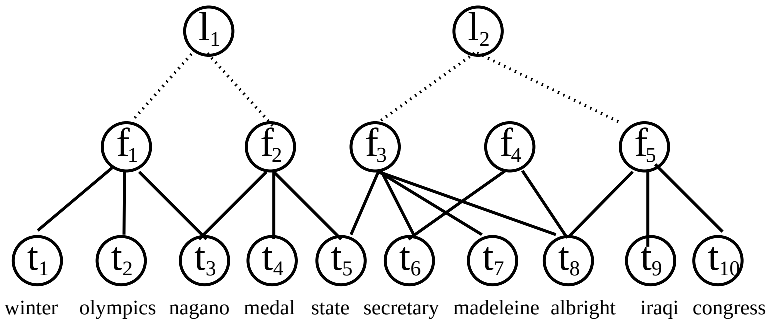

Figure 2: (The MMSS graph G_{MMSS}) Three types of nodes: logo-nodes l_i's, frame-nodes f_i's and term-nodes t_i's; and, two types of edges: "same-logo" edges (dotted) and the term-occurrence edges (solid). Each frame-node represents a video shot.

a term-node t_k by a term-occurrence edge, if the term $O(t_k)$ occurs in the shot whose keyframe is $O(f_j)$.

For logo story summarization and video retrieval, the essential part they share is to select objects pertaining to one (or a set of) query object(s). In logo story summarization, the query object is the logo(-node) of the story we want to summarize. The frames and terms forming the summary are selected based on their "relevance" to the query logo-node. As for video retrieval, we select video shots based on their "relevance" to the set of query terms. With the graph G_{MMSS}, we can turn the problem of computing "relevance" of an object with respect to the query objects, into a random walk on the graph G_{MMSS}, as we show next.

2.2 Random walk with restarts (RWR)

In this work, we propose to use *random walk with restarts* ("*RWR*") [7] to estimate the *relevance* of a node "v" with respect to the restart node "s". The "random walk with restarts" operates as follows: to compute the relevance of a node "v" for node "s", consider a random walker that starts from node "s". At every time-tick, the walker chooses randomly among the available edges, with one modification: before he makes a choice, he goes back to node "s" with probability c. Let $u_s(v)$ denote the stationary probability that our random walker will find himself at node "v". Then, $u_s(v)$ is what we want,

Input:
1. G_{MMSS}: a *MMSS* graph with N nodes (nodes are numbered from 1 to N).
2. $\mathcal{R}$: a set of restart nodes. ($|\mathcal{R}|$ is the number of nodes in $\mathcal{R}$)

Output:
$\overrightarrow{\mathbf{u}_{\mathcal{R}}}$: the RWR scores of all nodes with respect to $\mathcal{R}$

Steps:
1. Let $\mathbf{A}$ be the adjacency matrix of G_{MMSS}. Normalize the columns of $\mathbf{A}$ and make each column sum up to 1.
2. $\overrightarrow{\mathbf{v}_{\mathcal{R}}}$ is the N-by-1 restart vector, whose i-th element is $\frac{1}{|\mathcal{R}|}$, if node i is in $\mathcal{R}$; otherwise, the i-th element is 0.
3. Initialize $\overrightarrow{\mathbf{u}_{\mathcal{R}}}$=$\overrightarrow{\mathbf{v}_{\mathcal{R}}}$.
4. while($\overrightarrow{\mathbf{u}_{\mathcal{R}}}$ has not converged)
   4.1 $\overrightarrow{\mathbf{u}_{\mathcal{R}}} = (1\text{-c})\mathbf{A}\overrightarrow{\mathbf{u}_{\mathcal{R}}} + \text{c}\overrightarrow{\mathbf{v}_{\mathcal{R}}}$
5. Return the converged $\overrightarrow{\mathbf{u}_{\mathcal{R}}}$.

Figure 3: Algorithm RWR: $\vec{u}_{\mathcal{R}} = \text{RWR}(G_{MMSS}, \mathcal{R})$

the relevance of "v" with respect to "s", and we call it the *RWR score* of "v" (with respect to "s").

The intuition is that if the random walker who restarts (with probability c) from s has high chance of finding himself at node v, then node v is close (relevant) to s. Figure 3 gives the algorithm of RWR.

Definition 1 (RWR score) *The RWR score of node v with respect to the restart node s, indicating the relevance of v to s, is the stationary probability $u_s(v)$ of a random walk with restarts, as defined above.*

The stationary probability of *RWR* is dependent on the restart nodes, as opposed to the plain random walk (with or without damping) that the final stationary probability is independent to any node. Since we want the relevance of a node to be dependent on the query nodes, RWR fits our need better.

We summarize a logo story by choosing a set of keyframes which show the major scenes and people involved in the story, and a set of words that describes the story. To use RWR to summarize a logo story $O(l_i)$, we set the restart node s at the logo-node s=l_i. The frame(-node)s and term(-node)s

Input: 1. G_{MMSS}: a *MMSS* graph with N nodes (nodes are numbered from 1 to N). 2. $O(l)$: the logo story to be summarized. Let its logo-node be l. 3. p_F (p_T): number of frames (terms) to be selected for the summary.
Output: 1. $\mathcal{F}_l$: a set of p_F frame-nodes 2. $\mathcal{T}_l$: a set of p_T term-nodes
Steps: 1. Let $\mathcal{R} = \{l\}$. 2. Do $\vec{\mathbf{u}_l} = \mathrm{RWR}(G_{MMSS}, \mathcal{R})$ to obtain the RWR scores of all nodes with respect to the logo node l. 3. $\mathcal{F}_l$ = the set of p_F frame-nodes having the highest RWR scores among all frame-nodes. 3. $\mathcal{T}_l$ = the set of p_T term-nodes having the highest RWR scores among all term-nodes.

Figure 4: Algorithm: $[\mathcal{F}_l, \mathcal{T}_l]$ = Algo-VSum(G_{MMSS}, $O(l)$, p_F, p_T)

with the highest RWR scores are then selected as the story summary. The algorithm for story-oriented summarization, *Algo-VSum*, is given in Figure 4.

Similarly, for video retrieval, the restart nodes are set at the term-nodes corresponding to the query terms. The query result is the set of shots (frame-nodes) with the highest RWR scores. The algorithm for video retrieval, *Algo-VIR*, is given in Figure 5.

The computation of the stationary probability is very interesting and important. We use matrix notation, for compactness. We want to find the most related terms to the set of query nodes $\mathcal{Q}$. To do that, we do an RWR restarting randomly from any node in $\mathcal{Q}$, and compute the stationary probability vector $\vec{\mathbf{u}_\mathcal{Q}} = (u_\mathcal{Q}(1), \ldots, u_\mathcal{Q}(N))$, where N is the number of nodes in the G_{MMSS} graph. Here, we label each node in the G_{MMSS} graph sequentially from 1 to N.

The estimation of vector $\vec{\mathbf{u}_\mathcal{Q}}$ can be implemented efficiently by matrix multiplication. Let $\mathbf{A}$ be the N-by-N adjacency matrix of the G_{MMSS} graph. We normalize each column of A, so that each column sums up to 1, to make it a valid random-walk transition matrix.

Input: 1. G_{MMSS}: a *MMSS* graph with N nodes (nodes are numbered from 1 to N). 2. $\mathcal{Q}$: a set of query terms. 3. p: number of shots to be retrieved.
Output: $\mathcal{F}_{\mathcal{Q}}$: the set of p retrieved shots/frame-nodes. (Note: A frame-node represents a shot.)
Steps: 1. Let $\mathcal{R}$ be the set of query term-nodes corresponding to $\mathcal{Q}$. 2. Do $\vec{\mathbf{u}}_{\mathcal{R}} = \text{RWR}(G_{MMSS}, \mathcal{R})$ to obtain the RWR scores of all nodes with respect to the query $\mathcal{Q}$. 3. $\mathcal{F}_{\mathcal{Q}}$ is the set of p frame-nodes having the highest RWR scores among all the frame-nodes.

Figure 5: Algorithm: $\mathcal{F}_{\mathcal{Q}} = \text{Algo-VIR}(G_{MMSS}, \mathcal{Q}, p)$

Let $\vec{\mathbf{v}}_{\mathcal{Q}}$ be a N-by-1 vector with all its N elements zero, except for the entries that corresponds to the query nodes $\mathcal{Q}$, which are set to $\frac{1}{|\mathcal{Q}|}$ ($|\mathcal{Q}|$ is the number of query nodes). We call $\vec{\mathbf{v}}_{\mathcal{Q}}$ the "*restart vector*" for the query $\mathcal{Q}$. Now we can formalize the definition of the "relevance" (RWR score) of a node (Definition 1). Note that the set $\mathcal{Q}$ is equivalent to the set $\mathcal{R}$ of restart nodes in Figure 3 (Algorithm RWR).

Definition 2 (Stationary vector $\vec{\mathbf{u}}_{\mathcal{Q}}$) *Let $\mathcal{Q}$ be a set of query nodes from which the RWR restarts with probability c. RWR randomly picks one node from $\mathcal{Q}$ when it restarts. Let $\mathbf{A}$ be the column-normalized transition matrix. Then, the N-by-1 stationary probability vector $\vec{\mathbf{u}}_{\mathcal{Q}}$, satisfies the equation:*

$$\vec{\mathbf{u}}_{\mathcal{Q}} = (1-c)\mathbf{A}\vec{\mathbf{u}}_{\mathcal{Q}} + c\vec{\mathbf{v}}_{\mathcal{Q}}. \tag{1}$$

We can easily show that

$$\vec{\mathbf{u}}_{\mathcal{Q}} = c\left(\mathbf{I} - (1-c)\mathbf{A}\right)^{-1}\vec{\mathbf{v}}_{\mathcal{Q}}, \tag{2}$$

where $\mathbf{I}$ is the N-by-N identity matrix.

We can compute the stationary probability vector $\vec{\mathbf{u}}_{\mathcal{Q}}$ by inverting the sparse matrix $(\mathbf{I} - (1-c)\mathbf{A})$. By exploiting the sparseness of the matrix, the matrix inversion can be performed efficiently [5, 8].

Symbol	Description
N	number of documents
V	number of distinct words (vocabulary size)
R	number of the top singular vectors kept for LSI
$\lvert\mathbf{D}\rvert$	length of document $\mathbf{D}$
$\mathcal{V}$	word vocabulary $\{w_1, \ldots, w_V\}$
i	index to documents
j	index to words
$\mathbf{D_i}$	the i-th document in the document collection
$\mathbf{Q}$	the query (set of words)
$\mathbf{T}$	OKAPI similarity matrix (equation 5)
$\mathbf{T}_R$	LSI similarity matrix (equation 6)
$\overrightarrow{\mathbf{q}}$	V-by-1 query vector
$\mathbf{sim}(\overrightarrow{\mathbf{q}})$	N-by-1 document similarity to query $\overrightarrow{\mathbf{q}}$ (from OKAPI)
$\mathbf{sim}_R(\overrightarrow{\mathbf{q}})$	N-by-1 document similarity to query $\overrightarrow{\mathbf{q}}$ (from LSI)

Table 1: Symbols used in this paper

2.3 Textual retrieval techniques: OKAPI and LSI

Besides story-oriented summarization, *MMSS* is versatile and can be applied to other tasks such as video retrieval. In this paper, we compared the performance of *MMSS* on video retrieval with two traditional information retrieval techniques, *OKAPI* and *LSI*, which use only textual information. OKAPI and LSI treat each video shot as a document of transcript words. The three methods assign scores to video shots with respect to a set of query words, and return the shots with high scores as the retrieval result of the query. We compare the shots retrieved by the three methods, and examine their relevance to the query. Here, we briefly introduce the methods OKAPI and LSI. Table 1 summaries the symbols we use in this paper.

The OKAPI method The OKAPI method [9] is reported to be one of the best methods for measuring document-to-query similarity. The OKAPI method defines the similarity between a query $\mathbf{Q}$ and a document $\mathbf{D}$ as

$$sim(\mathbf{Q}, \mathbf{D}) = \sum_{qw \in \mathbf{Q}} \frac{tf(qw, \mathbf{D}) \log\left(\frac{N - df(qw) + 0.5}{df(qw) + 0.5}\right)}{0.5 + 1.5\frac{|\mathbf{D}|}{avg\_dl} + tf(qw, \mathbf{D})}. \quad (3)$$

In Equation 3, N is the total number of documents in the collection; $tf(qw, \mathbf{D})$ is the frequency of the term qw in the document $\mathbf{D}$; $df(qw)$ is the document frequency of the term qw (the number of documents in the collection containing the term qw); $|\mathbf{D}|$ is the length of the document $\mathbf{D}$; and avg_dl is the average length of all the N documents. The similarity function of a document and a query has been finely tuned to achieve outstanding performance.

For a query $\mathbf{Q}$, we need to evaluate $sim(\mathbf{Q}, \mathbf{D_i})$ for every document $\mathbf{D_i}$ in the collection, so that we can rank the documents by their similarity to the query $\mathbf{Q}$. We can present this overall evaluation in matrix form. Let $\mathcal{V} = \{w_1, \ldots, w_V\}$ be the vocabulary of all the V possible transcript words. Let i be the index to documents and j be the index to words. Define the *OKAPI similarity matrix* $\mathbf{T}$ as a N-by-V matrix whose (i,j)-element $T_{i,j}$ is

$$T_{i,j} = \frac{tf(w_j, \mathbf{D_i}) \log\left(\frac{N - df(w_j) + 0.5}{df(w_j) + 0.5}\right)}{0.5 + 1.5\frac{|\mathbf{D_i}|}{avg\_dl} + tf(w_j, \mathbf{D_i})}. \tag{4}$$

A query is represented as a V-by-1 vector $\overrightarrow{\mathbf{q}}$, with its j-element be the frequency of word w_j in the query $\mathcal{Q}$. Elements in $\overrightarrow{\mathbf{q}}$ are mostly zero, except those corresponding to the words that appear in the query. Let $\mathbf{sim}(\overrightarrow{\mathbf{q}})$ be a N-by-1 vector of the OKAPI similarity scores of all the documents to a query $\overrightarrow{\mathbf{q}}$. $\mathbf{sim}(\overrightarrow{\mathbf{q}})$ is defined as

$$\mathbf{sim}(\overrightarrow{\mathbf{q}}) = \mathbf{T}\overrightarrow{\mathbf{q}}. \tag{5}$$

Notice that the i-th element of $\mathbf{sim}(\overrightarrow{\mathbf{q}})$ is the OKAPI similarity score, $sim(\mathbf{Q}, \mathbf{D_i})$, between document $\mathbf{D_i}$ and the query $\mathbf{Q}$.

The LSI method Latent semantics indexing (LSI) [1] has shown great success in information retrieval applications. In this work, we also compare our video retrieval result with the result of LSI. We apply LSI to construct a N-by-V matrix $\mathbf{T}_R$ from $\mathbf{T}$ (Equation 4), where R is the number of singular vectors kept by the LSI. Specifically, singular value decomposition (SVD) is applied on $\mathbf{T}$ and decomposes $\mathbf{T}$=$\mathbf{U}\Lambda\mathbf{V^T}$. Let $\mathbf{U}_R$ be the N-by-R matrix consisted of the first R columns of $\mathbf{U}$. Similarly, $\mathbf{V}_R$ is the V-by-R matrix consisted of the first R columns of $\mathbf{V}$. Let Λ_R be the top-left R-by-R submatrix of matrix Λ. The *LSI similarity matrix* $\mathbf{T}_R$ is defined as

$$\mathbf{T}_R = \mathbf{U}_R \Lambda_R \mathbf{V}_R^T. \tag{6}$$

The LSI similarity scores of all documents to the query $\overrightarrow{\mathbf{q}}$ can be computed as

$$\mathbf{sim}_R(\overrightarrow{\mathbf{q}}) = \mathbf{T}_R\overrightarrow{\mathbf{q}}. \tag{7}$$

In our experiments, each shot is treated as a document - a document of the transcript words in the shot. We remove stop words from the transcript and keep only terms that are nouns. Given a query, the documents/shots which have the highest OKAPI or LSI similarity scores are retrieved as the query result.

3 Experimental Results

The experiments are designed to answer the following questions: (a) For story summarization, how informative are the shots (keyframes) and the terms that *MMSS* chooses? (b) For video retrieval by textual query, how well does *MMSS* do, comparing to the existing successful textual retrieval methods, like OKAPI and LSI?

Specifically, the problems of story summarization and video retrieval are

Problem 1 (Story summarization) *Given a logo, find the best shots and/or terms for it.*

Problem 2 (Video retrieval) *Given a query word, find relevant video shots.*

We should emphasize that OKAPI and LSI can only answer queries of the form "given a query, find relevant video shots". Our *MMSS* method, being modality-independent, can answer other types of queries. For example, queries such as "given a shot (whose keyframe does not have a logo), find the best logo for it", which can be done by computing the RWR scores of the logo-nodes (restarting from the node of the given shot); or "given a logo, find other related logos", which can be done by computing RWR scores on all logo-nodes (restarting from the query logo-node).

In our experiments, we follow the guidelines from [7] and set the restart probability c=0.65 for our 3-layer G_{MMSS} graph.

3.1 Story summarization

MMSS summarizes a logo story using the frames and terms which have the highest RWR scores (with respect to the logo story). Figures 6 shows the top 30 frames selected by *MMSS* for the logo "Iraq" (Figure 1 (c)). The top 7 frames are the logo frames detected by the iconic matching. These frames are ranked high, simply because they are connected directly to the restart

Figure 6: (Visual summary of logo "Iraq") Frames are sorted (highest score first).

Figure 7: (Visual summary of the logo story "Winter Olympics") Frames are sorted (highest score first).

logo-node in the graph G_{MMSS}. Interestingly, *MMSS* found extra "Iraq" logo frames (e.g., the logo frame ranked 16) missed by the iconic matching.

MMSS selects informative frames about the logo story, where faces of the major players could be easily seen. For example, Kofi Annan appears in the frames ranked 9-th and 20-th. In addition, frames which contain *overlaid text* are also selected, as shown in the frames ranked 26-th and 28-th - the "Crisis in the Gulf"- on which the current developments in Iraq are listed. We emphasize that the information of overlaid text is important and may not be available to the textual retrieval methods, for they are rarely fully mentioned by the anchorperson and therefore, are not in the transcript. Other logos pertaining to the logo "Iraq" are also detected and selected, for example, the "Yeltsin" logo at rank 14 and the "Canada-Iraq" logo at rank 29.

Figure 7 shows the top 15 frames selected by *MMSS* for the logo story "Winter Olympics" (Figure 1 (a)). All top 15 frames are pertinent to the topic "Winter Olympics". The selected frames are very informative, where

Figure 8: (Visual summary of logo "Lewinsky") Frames are sorted (highest score first).

scenes of major activities are shown (e.g., frames ranked 7-th and 8-th). Frames with informative overlaid text are also selected by *MMSS*. For example, the frame ranked 4-th gives the athletes' names, countries and finishing times on the final result of Women's 7.5 kilometers biathlon. The top three frames are the logo frames detected by the iconic matching. Logo frames missed by the iconic matching are again found by *MMSS* (e.g, the logo frame at rank 5), as in the case of logo story "Iraq".

Figure 8 shows the top 15 frames selected by *MMSS* for the logo "Lewinsky" (Figure 1 (b)). Faces of the major players are shown in the selected frames. For example, Starr is shown in the frames at rank 5, 11 and 12. Frames which contain overlaid text are selected (rank 14), as well as other logos pertaining to the query logo (rank 15, logo "Clinton investigation").

Observation 2 (Visual summary by *MMSS*) *MMSS summarizes logo stories by selecting relevant frames from the news video collections. Specifically, MMSS selects frames*

- *of persons, objects, activities which are significant to the story;*

Logo story	Summarizing terms
"Winter Olympics"	winter medal gold state skier headline news result superticker olympics competition nagano ceremony watch night round game team sport weather photo woman that today canada bronze year home storm coverage
"Lewinsky"	house lawyer intern ginsburg starr bill whitewater counsel immunity president clinton monica source information money trie learned iraq today state agreement country client weapon force nation inspection courthouse germany support
"Iraq"	iraq minister annan kofi effort baghdad report president arab strike defense sudan iraqi today weapon secretary talk school window problem there desk peter student system damage apart arnett albright secretarygeneral

Table 2: (Textual summary by *MMSS*) Terms are sorted (highest score first).

- *with meaningful overlaid text;*
- *which contain the logos but are missed by the "iconic matching" technique;*
- *of other relevant logos.*

Besides ranking frames for summarization, *MMSS* also ranks and selects relevant terms at the same time. Table 2 shows the terms selected by *MMSS* for summarizing three logo stories in Figure 1, namely "Winter Olympics", "Lewinsky" and "Iraq". The selected terms are meaningful and convey the content of the logo stories. Together with the selected frames in Figures 6, 7, and 8, we found that *MMSS* successfully provides multi-modal (frames and terms) summaries of logo stories.

3.2 Video retrieval

In the task of video retrieval, we are given a query (a set of terms), the goal is to retrieve shots which are most relevant to the query. In other words, we want to rank all the shots by their relevance to the set of query words. The queries used in our experiments are: {`‘‘lewinsky’’`, `‘‘clinton’’`}, {`‘‘lewinsky’’`}, {`‘‘clinton’’`}, {`‘‘olympics’’`}, {`‘‘annan’’`, `‘‘iraq’’`}, {`‘‘annan’’`}, and {`‘‘iraq’’`}.

Figure 9: Keyframes of the top 10 shots retrieved by *MMSS* on query {``‘‘lewinsky’’``, ``‘‘clinton’’``}. Frames are sorted (highest score first).

Since the data set we use does not have ground truth for any query, we do not report the standard precision and recall measures. Instead, we inspect the result by human judgment. We leave the precision/recall experiments to the future work.

We notice that a shot which contains many query words does not necessarily has meaningful content about the query. A “teaser” which gives an overview of all the stories that will be covered in a news program is such an example. In a teaser shot, many keywords about many stories are mentioned by the anchor, however, no detail about any story is provided there. Besides, a teaser is usually accompanied with the anchor shots and does not have informative scene shots. Traditional textual retrieval methods are likely to retrieve teaser-style shots, for they are full of keywords. On the other hand, *MMSS* is unbiased to the teasers, as we show next.

Figure 9 shows the shots retrieved by *MMSS* for the query {``‘‘lewinsky’’``, ``‘‘clinton’’``}. The frontal view of the major players related to the query is at the top of the list, for example, Starr at rank 1 and Monica at rank 4. Other shots at the top of the list are the shots of related logo stories - shots of the logo stories “Clinton investigation” (at rank 3, 7 and 10) and “Jordan” (at rank 5).

In addition, *MMSS* avoids the news “teasers” while OKAPI and LSI rank the teaser shots with high scores. For example, in Figure 10, the rank 10 shot chosen by OKAPI is a teaser shot - indicated by the words “In the next

Figure 10: Keyframes of the top 10 shots retrieved by OKAPI on query {`‘‘lewinsky’’`, `‘‘clinton’’`}. Frames are sorted (highest score first).

Figure 11: Keyframes of the top 10 shots retrieved by LSI on query {`‘‘lewinsky’’`, `‘‘clinton’’`}. Frames are sorted (highest score first).

30 minutes” at the background of the keyframe. In Figure 11, the shots at rank 5, 8, and 10 chosen by LSI are teaser shots: rank 5th shot has a small symbol (“top stories”) at the bottom left of the keyframe, and shots at rank 8 and 10 have keyframes of the lottery numbers, which are usually shown right before the opening teaser of a news program. Results of other queries yield similar observations and are not shown here.

Method	Transcript term (histogram)
MMSS	clinton lewinsky president monica attorney today house jury starr bill washington relationship story whitewater lawyer daughter jones counsel intern investigation immunity conversation headline minute ginsburg affair question judge mother office
OKAPI	clinton(16) lewinsky(11) monica(6) president(6) service(2) minute(2) report(2) blitzer(1) wolf(1) conversation(1) controversy(1) lewis(1) testimony(1) agent(1) nature(1) officer(1) claim(1) exchange(1) immunity(1) intern(1) house(1) affair(1) attorney(1) question(1) relationship(1) whitewater(1) headline(1) time(1) office(1)
LSI (using 50 singular vectors)	clinton(20) president(20) mandela(1) friend(1) congress(1) lady(1) visit(1) administration(1) relationship(1) washington(1)

Table 3: Query result summary of the query {`‘‘lewinsky’’`, `‘‘clinton’’`}. Numbers in the parentheses are the counts of the terms in the top 30 shots retrieved by OKAPI or LSI. Terms are sorted (highest RWR scores or frequencies first).

Observation 3 (OKAPI and LSI are biased to teaser shots) *Textual retrieval methods such as OKAPI and LSI prefer teaser shots, such as, the "headlines preview" at the beginning of news programs, due to the many keywords the news anchors mentioned in those shots. Unfortunately, these teasers do not contain major shots of story content.*

Besides selecting relevant shots/keyframes for the user query, *MMSS* also provide a list of keywords ranked by their relevance (RWR scores) to the query. The result can be viewed as a textual summary of the retrieved shots. On the other hand, the OKAPI and LSI methods do not have a straightforward way to generate such a query result summary. For comparison, we collect the word-count histogram of the transcript words in the shots retrieved by the OKAPI or LSI, and consider such a histogram as the query result summary. These histograms are compared with the query result summary given by *MMSS*. We note that in our experiments, the transcript words are "filtered", where stop words are removed and only nouns are kept.

Table 3 compares the textual summaries of the retrieval results by *MMSS*, OKAPI and LSI. The query is {`‘‘lewinsky’’`, `‘‘clinton’’`}. *MMSS* reports the top 30 terms, ordered by their RWR scores. Each of the OKAPI and LSI (using the first 50 singular vectors) reports the histogram of the terms

Method	Transcript term (histogram)
MMSS	olympics winter sponsor headline sport dial moscow news network medal drug disappointment rumor region game gold technology underwhelming quarter hold pair champion reason later michael entertainment next force panel dennis
OKAPI	olympics(20) winter(6) sponsor(3) headline(3) sport(3) dial(2) news(2) underwhelming(1) disappointment(1) snowboarding(1) moscow(1) network(1) rumor(1) next(1) michael(1) pair(1) medal(1) gold(1) drug(1) technology(1) hold(1) later(1) entertainment(1) reason(1) champion(1) region(1) force(1) dennis(1) persian(1) panel(1)
LSI (using 50 singular vectors)	sport(20) news(3) headline(2) article(1) winter(1) bull(1) month(1)

Table 4: Query result summary of the query {`‘‘olympics’’`}. Numbers in the parentheses are the counts of the terms in the top 30 shots retrieved by OKAPI or LSI. Terms are sorted (highest RWR scores or frequencies first).

in the top 30 retrieved shots. Terms of the highest RWR scores (*MMSS*) or frequencies (OKAPI/LSI) are listed first.

For the query {`‘‘lewinsky’’`, `‘‘clinton’’`}, *MMSS* selects relevant terms such as `‘‘monica’’` and `‘‘whitewater’’`, as OKAPI does. For this particular query, *MMSS* also successfully retrieves terms like `‘‘starr’’` and `‘‘jones’’`, which OKAPI and LSI miss. Other queries yield similar observations: For example, Table 4 compares the query result summaries on the query {`‘‘olympics’’`}; Table 5 compares the query result summaries on the query {`‘‘iraq’’`}. In fact, *MMSS* picks up more meaningful terms, because it is not restricted to select terms that are in the 30 retrieved shots, and can consider all possible terms. In general, without sophisticated parameter tuning, *MMSS* gives query result summary (at least) as good as those given by OKAPI and LSI.

We want to emphasize that *MMSS ranks terms and frames/shots independently*, as opposed to the textual retrieval methods which rank the shots/frames first, and then collect the terms in the top ranked shots as "relevant". The ability of *MMSS* to rank terms and frames independently produces a more meaningful query result summary.

Method	Transcript term (histogram)
MMSS	iraq weapon president inspector clinton saddam standoff site minister security annan nation state strike action today resolution council inspection consequence defense baghdad attack agreement access month gulf people unscom secretarygeneral
OKAPI	iraq(20) consequence(2) upheaval(1) spirit(1) cooperation(1) apart(1) stage(1) thursday(1) reaction(1) demand(1) conflict(1) agreement(1) minister(1) unscom(1) school(1) saddam(1) inspector(1) resolution(1) people(1) month(1) president(1)
LSI (using 50 singular vectors)	iraq(17) weapon(12) inspector(11) site(10) access(5) council(4) security(4) annan(3) month(3) secretarygeneral(2) member(2) nation(2) action(2) standoff(2) president(2) official(2) butler(1) monitor(1) wait(1) cooperation(1) biological(1) survavers(1) jacques(1) pressurer(1) mapping(1) production(1) arriving(1) size(1) kofi(1) chemical(1) building(1) richard(1) minister(1) chief(1) unscom(1) finding(1) say(1) commission(1) arnett(1) peter(1) baghdad(1) strike(1) mission(1) team(1) saddam(1) agency(1) russian(1) resolution(1) part(1) work(1) today(1) report(1) number(1) news(1) washington(1) morning(1)

Table 5: Query result summary of the query {`‘‘iraq’’`}. Numbers in the parentheses are the counts of the terms in the top 30 shots retrieved by OKAPI or LSI. Terms are sorted (highest RWR scores or frequencies first).

4 Conclusions

We propose *MMSS* for multi-modal story-oriented video summarization and video retrieval. Logos (Figure 1) extracted from shot keyframes provide a robust hint to group shots of each logo story. *MMSS* integrates both the textual and logo information in a graph. The random work with restarts (RWR) is used to obtain a story-specific relevance ranking (RWR scores, Definition 1) on the terms and shot keyframes. Our experiments on the TRECVID 2003 data set show that *MMSS* is effective and gives meaningful multi-modal story-oriented summaries. Moreover, *MMSS* matches the performance of the best textual retrieval methods on video retrieval. In fact, *MMSS* sometimes does better, because it avoids the news “teasers” (Observation 3). Unlike the textual retrieval methods, *MMSS* achieves these with no sophisticated parameter tuning.

Finally, we note that *MMSS* can discover correlations between objects of different modalities. Besides answering queries of video summarization

(Problem 1) and retrieval (Problem 2), *MMSS* is general and can answer other queries such as “given a shot whose keyframe does not contain a logo, find the best logo for it”, or “given a logo, find other related logos.”

References

[1] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[2] P. Duygulu, J.-Y. Pan, and D. A. Forsyth. Towards auto-documentary: Tracking the evolution of news stories. In *Proceedings of the ACM Multimedia Conference*, October 2004.

[3] J. Edwards, R. White, and D. Forsyth. Words and pictures in the news. In *HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, May 2003.

[4] J. Goldstein, V. O. Mittal, J. Carbonell, and J. Callan. Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the Ninth International Conference on Information Knowledge Management (CIKM-00)*, November 2000.

[5] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical Report 2003-35, Stanford University, June 2003.

[6] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Automatic video summarization by graph modeling. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, 2003.

[7] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *Proceedings of the 10th ACM SIGKDD Conference*, August 2004.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipe in C.* Cambridge University Press, 1992.

[9] S. Robertson and S. Walker. Okapi/Keenbow at trec-8. In *Proceedings of The Eighth Text REtrieval Conference (TREC 8)*, pages 151–162, 1998.

[10] A. F. Smeaton, W. Kraaij, and P. Over. TRECVID 2003 - an introduction. In *Proceedings of TREC 2003*, 2003.

[11] M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques. In *Proceedings of CVPR 1997*, pages 775–781, June 17-19 1997.

[12] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer*, 32(2):66–73, February 1999.