Multimodal News Story Clustering With Pairwise Visual Near-Duplicate Constraint

Xiao Wu, *Student Member, IEEE*, Chong-Wah Ngo, *Member, IEEE*, and Alexander G. Hauptmann, *Member, IEEE*

***Abstract*—Story clustering is a critical step for news retrieval, topic mining, and summarization. Nonetheless, the task remains highly challenging owing to the fact that news topics exhibit clusters of varying densities, shapes, and sizes. Traditional algorithms are found to be ineffective in mining these types of clusters. This paper offers a new perspective by exploring the pairwise visual cues deriving from near-duplicate keyframes (NDK) for constraint-based clustering. We propose a constraint-driven co-clustering algorithm (CCC), which utilizes the near-duplicate constraints built on top of text, to mine topic-related stories and the outliers. With CCC, the duality between stories and their underlying multimodal features is exploited to transform features in low-dimensional space with normalized cut. The visual constraints are added directly to this new space, while the traditional DBSCAN is revisited to capitalize on the availability of constraints and the reduced dimensional space. We modify DBSCAN with two new characteristics for story clustering: 1) constraint-based centroid selection and 2) adaptive radius. Experiments on TRECVID-2004 corpus demonstrate that CCC with visual constraints is more capable of mining news topics of varying densities, shapes and sizes, compared with traditional k-means, DBSCAN, and spectral co-clustering algorithms.**

***Index Terms*—Multimedia topic detection and tracking, near-duplicate visual constraint, news story clustering, video data mining.**

I. Introduction

NEWS videos are broadcast everyday across different sources, times, languages, and countries. With the overwhelming volume of news videos available today, it becomes necessary to track the development of news stories from different channels, mine their dependencies and organize them in a semantic way [34]. Story clustering is a fundamental step for news browsing, retrieval, topic tracking and summarization, targeting for the tracking of stories, either supervised or unsupervised, into clusters of news topics according to their contents and themes. Current research efforts in managing multimedia documents are mainly devoted to the classification of stories to few generic but coarse topics such as finance, politics, and sports. Mining a finer granularity of clusters which can link together evolving and historical stories according to topics such as "September 11th terrorist attack" and "London bombing", nevertheless, appears more interesting and demanding. The stories under one topic can be further structured and threaded, for instance by the novelty and redundancy detection, to provide users an efficient means for browsing a query topic according to the storyline [34].

Effective mining of evolving stories across sources, nevertheless, imposes several challenges to the existing clustering algorithms. News stories record the gradual evolution of topics over time, intertwining with a variety of old and fresh developments. Under our observation, the feature distribution of stories, either within or across topics, often exhibits varying shapes, sizes and densities. For instance, the topic "Arkansas school shooting", which gradually evolves and develops into several subthemes, forms a cluster of complex shape with varying densities across different parts of the cluster. On the contrary, the topic "Yeltsin fired the whole cabinet" which is only mentioned few times appears as a small but compact cluster. The nature of this problem indeed poses a serious problem to the majority of clustering algorithms in the data mining field which normally assume clusters in the form of convex shape and uniform density.

In news videos, there exist a large number of near-duplicate keyframes (NDK) which frequently appear at different times/dates and across various broadcast sources. Near-duplicate keyframes, by definition, are a set of keyframes in which one is close to the exact duplicate of the other, but different in the capturing conditions (camera, camera parameter, view angle, etc.), acquisition times, rendering conditions, or editing operations [38], as shown in Fig. 1. These NDK pairs basically form pairwise equivalent constraints that are useful for bridging evolving news stories across times and sources. For example, the stories having the picture of two young suspects shown in Fig. 1(b) are highly related, and thus should be clustered together. A statistic in [38] found that there are approximately 10%–20% of NDK in TRECVID-2004 news corpus [31]. This indicates that there are abundant NDK useful for mining in news videos.

Generally speaking, pairwise constraints can be derived from textual information as well. However, the derivation of equivalent constraints from text is not as direct and easy as with NDK pairs, considering the amount of time required for identifying constraints and the level-of-subjectivity involved. Furthermore, different broadcast sources may use diverse keywords to describe a topic, and there is the basic difficulty of mining similar stories from different languages. In this paper, we treat visual

Manuscript received February 7, 2006; revised September 11, 2007. This work was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 118905) and a grant from City University of Hong Kong (Project No. 7002112). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Florent Masseglia.

X. Wu is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA and the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: wuxiao@cs.cityu.edu.hk).

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: cwngo@cs.cityu.edu.hk).

A. G. Hauptmann is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: alex@cs.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2007.911778

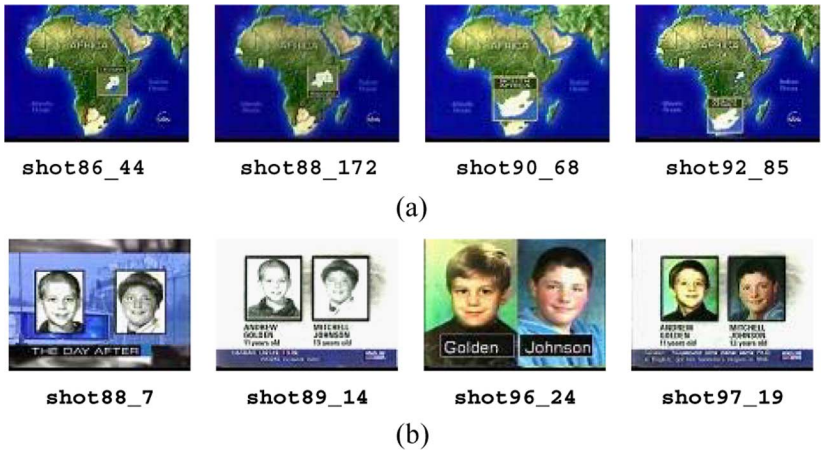


Fig. 1. Near-duplicate keyframe pairs across different stories of two topics (the label under each keyframe is its ID, e.g., shot88_7 denotes the keyframe of the seventh shot in video 88). (a) Keyframes of a map of Africa appear across different stories of the topic "Clinton visited Africa". (b) Keyframes of the "suspects" appear across different stories of the topic "Arkansas school shooting".

constraints as a "must-link" condition, which means that two stories sharing at least one pair of NDK are always placed into the same cluster. In some situations, a "must-link" constraint may not necessarily true, one might prefer to consider NDK as "soft-link", taking into account various factors such as the similarity level of the keyframes and the number of NDK links across two stories. Our approach, nonetheless, simply assumes the "must-link" constraints to avoid the uncertainty caused by defining "soft-link" heuristics, which allows us to more directly explore the potential of visual constraints in news story clustering.

In this paper, the main contribution is the novel idea of mining topic-related stories through clustering on the basis of visual constraints built on top of text. The constraints are the pairs of NDK serving as must-link constraints to provide visual cues for grouping topic-related stories. The constraints are exploited in several aspects during clustering. Firstly, we co-cluster stories, constraints and speech transcripts to explore the duality between stories and their underlying features in multiple modalities. Secondly, co-clustering reduces the feature dimensionality which potentially accelerates the forming of clusters with density-based algorithm such as DBSCAN [13]. Thirdly, cluster centroids are determined with the assistance of constraints, which effectively avoid initializing clusters using outliers or cluster borders. Finally, we further exploit the constraints to adapt the radius scale window, a feature not emphasized in DBSCAN, for growing clusters of varying densities. The visual constraints are found to be useful for bridging clusters with diverse within-densities, while capable of discriminating clusters of different topics by the effective means of mining cluster borders based on the density notion.

Fig. 2 depicts our proposed constraint-driven co-clustering (CCC) algorithm. Formally, a story is defined as a group shots depicting an event. Each shot is described by a representative keyframe. Initially, a weighted bipartite graph is formed by modeling the duality between the stories and their underlying multimodal features (textual and visual keywords). The NDK pairs take part by forming various visual feature nodes as shown in the figure, where each node acts as a visual keyword analogous to the representation of textual keyword. With this bipartite graph, a normalized cut algorithm under the co-clustering framework, as proposed in [10], is employed to generate a reduced dimensional matrix which encodes the relationship of stories and features. Subsequently, a constraint-based DBSCAN algorithm (CB-DBSCAN) takes this matrix as input and operates in a low-dimensional feature space. By further imposing the pairwise NDK constraints on the matrix, CB-DBSCAN can seamlessly drive the cluster initialization and parameter selection by exploring the underlying story distribution through constraint propagation. The algorithm, with the aid of the pairwise visual constraints, dynamically adjusts the radius scale to grow clusters of varying densities, while being capable of identifying outliers that do not form interesting news topics.

The remainder of this paper is organized as follows. In Section II, we give a brief description of related work. Section III describes the co-clustering algorithm which integrates multimodal (textual and visual) features to generate a reduced dimensional matrix. Section IV discusses the utilization of near-duplicate visual constraints for Constraint-Based DBSCAN (CB-DBSCAN). The detailed procedure of the CCC algorithm and the experimental results are presented in Section V and Section VI, respectively. Finally, Section VII concludes this paper.

II. Related Work

News story clustering is a fundamental step for topic novelty and redundancy mining [34]. Story clustering is normally studied under the theme of topic detection and tracking (TDT) [2] with textual features as the underlying cues. TDT investigates several aspects of news stories organization such as the text-based story link detection, topic detection (clustering), and topic tracking. In addition to text transcripts, news videos indeed provide richer visual information. Current approaches for story clustering or tracking in multimedia area mainly focus on fusing multimodalities such as textual and visual features. Duygulu *et al.* [12] presented the technique to mine and track the repeated sequence of shots. A hierarchical video content description and

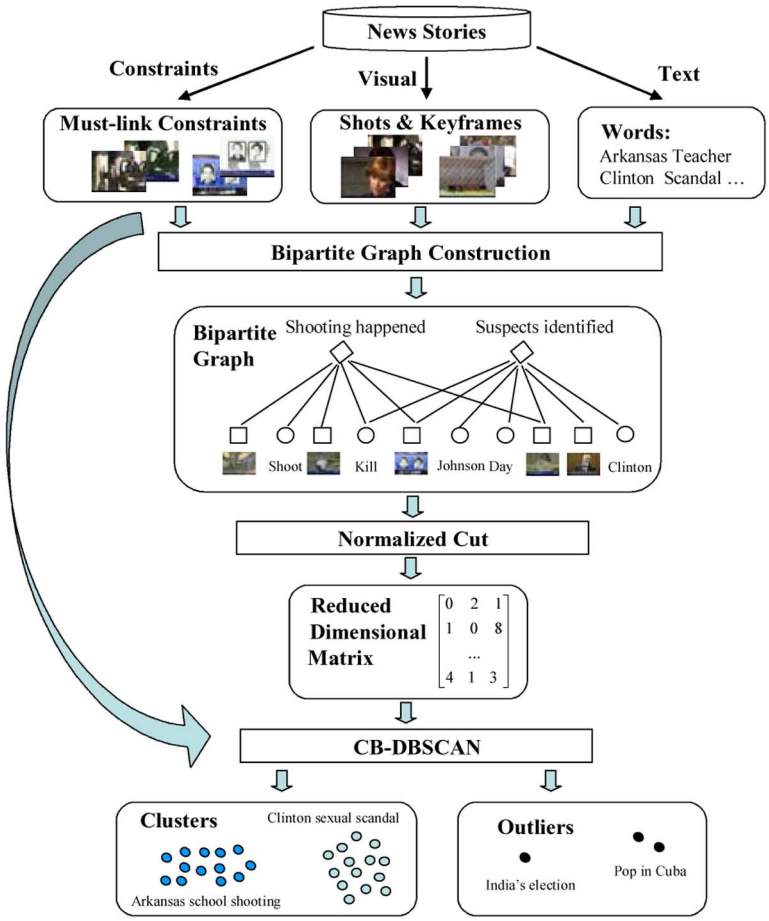


Fig. 2. Proposed constraint-driven co-clustering (CCC) algorithm.

summarization strategy was proposed in [40] with the support of a joint textual and visual similarity. Zhai *et al.* [37] linked news stories by combining keyframe matching and textual correlation. Recently, near-duplicate keyframes were also exploited in [8] to boost the performance of interactive video search. Hsu *et al.* [18] tracked four topics with visual duplicates and semantic concepts, and found that near-duplicates significantly improve tracking performance. Different from previous multimodality fusion, our work is built on the basis of visual constraints which are applied on top of text to improve the story clustering and mining.

Document clustering has been extensively studied and a large variety of algorithms have been proposed in the text and multimedia area, including hierarchical clustering [9], [29], [30], partitional clustering [11], [25], [28], density-based clustering [13], and spectral clustering [10], [23]. Recently, due to the demand for clustering inter-related heterogeneous objects (such as the documents and terms in a text corpus), co-clustering (CC) algorithms (e.g., [10], [26], [36], and [39]) were proposed to co-cluster two types of heterogeneous data. CC is a variance of traditional spectral clustering algorithms through the simultaneous exploitation of documents and their features under a bipartite graph representation. A consistent bipartite graph co-partitioning algorithm was also proposed in [14] to achieve a consistent partitioning and this has been applied to web image clustering [15]. Yu *et al.* [35] integrated partial grouping with normalized cuts for image segmentation. Furthermore, the clustering of evolving data streams (e.g., [1], [6]) has attracted a lot of research attention, which is more similar to new event detection [4] in the information retrieval area. In this paper, we focus on clustering of static data, instead of a dynamic data stream.

Considering the mining in multimedia environment, we employ and revisit the co-clustering framework [10] for two aspects. Firstly, CC is used to exploit the duality between stories and their multimodal features. Secondly, we adopt the proposed CB-DBSCAN algorithm to tailor clustering by exploring visual constraints. Under the notion of density, the algorithm partitions stories into dense regions (clusters) separated by regions with low density (outliers). The original DBSCAN algorithm [13] is excellent for identifying clusters with arbitrary shapes and sizes but not varying densities. Basically two parameters are required to define the notion of density: a maximum radius of the neighborhood (*Eps*) and a minimum number of points (*MinPts*) in an *Eps* neighborhood to decide the reachability of stories. These parameters are unknown but assumed fixed throughout the clustering. With the existence of constraints, CB-DBSCAN, in contrast, is capable of adapting *Eps* to cope with varying densities. More closely related work perhaps is OPTICS [3], which is also a variance of DBSCAN. OPTICS serves as a visualization or preprocessing tool prior to clustering for revealing the underlying cluster structure by observing the variation of *Eps* under fixed *MinPts*. Note that the change of *Eps* in OPTICS is data driven and passive, while it is constraint driven and adaptive in CB-DBSCAN. The difference allows CB-DBSCAN to

capitalize on the adaptive *Eps* for clustering, rather than merely utilizing *Eps* for visualization as in OPTICS.

Constraint-based clustering has been actively researched recently [7], [17], [20]–[22], [32]. Unlike global, cluster-level, or feature-level constraints, pairwise constraints provide instance-level information. They are easier to generate, but provide only weak "supervisory" hints. For instance, pairwise constraints only give information about pairs of instances and can be derived directly from class labels, but not vice versa. In [32], the must-link and cannot-link pairwise constraints were introduced into a k-means algorithm to improve the clustering performance. Later, spatial generalization was further considered in [32] so that pairwise constraints can also affect the neighboring data points. In [21] and [22], must-link equivalent and cannot-link constraints were incorporated into the expectation-maximization (EM) algorithm and posterior inference for model-based clustering. In addition, pairwise constraints were also utilized by [7], [17] to learn distance metrics for image and video retrieval. To the best of our knowledge, there are few cases to study the constraint driven mining under a density notion [27].

III. Co-Clustering With Multimodal Features

Different from traditional text documents, news videos provide rich visual information. For news story clustering, the pure textual method may overlook the interactions between visual and textual information. The employment of either textual or visual features may not be sufficient since either feature can appear differently over time in news videos. A robust approach should take into account both textual and visual features while exploiting the interaction of these features. CC just provides the right framework to explore the duality between news stories and their multimodal features (words and keyframes). CC is composed of two parts: normalized cut (NC) and k-means. The former partitions the bipartite graph constructed by the duality of stories and features, and produces a reduced dimensional matrix for k-means to perform k-way clustering.

A bipartite graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ is constructed, with $\boldsymbol{V} = \boldsymbol{S} \cup \boldsymbol{F}$ as the vertex set composed of stories ($\boldsymbol{S}$) and features ($\boldsymbol{F}$), and $\boldsymbol{E}$ as the edge set that specifics the association between $\boldsymbol{S}$ and $\boldsymbol{F}$. The feature $\boldsymbol{F}$ consists of words and keyframe clusters. The weight between the stories and features forms the affinity matrix $\boldsymbol{A}$ for graph partitioning.

The second smallest eigenvector of the *Laplacian matrix* $\boldsymbol{L}$ is an approximate solution for bipartitioning with the minimum normalized cuts. In the bipartite case

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{D}_1 & -\boldsymbol{A} \\ -\boldsymbol{A}^{\boldsymbol{T}} & \boldsymbol{D}_2 \end{bmatrix} \text{ and } \boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_1 & \mathbf{o} \\ \mathbf{o} & \boldsymbol{D}_2 \end{bmatrix}$$

where $\boldsymbol{D}_1(i,i) = \sum_j \boldsymbol{A}_{ij}$ and $\boldsymbol{D}_2(j,j) = \sum_i \boldsymbol{A}_{ij}$. Let $\boldsymbol{u} = \boldsymbol{D}_1^{1/2}\boldsymbol{x}$ and $\boldsymbol{v} = \boldsymbol{D}_2^{1/2}\boldsymbol{y}$, the general eigenvalue problem $\boldsymbol{Lz} = \lambda \boldsymbol{Dz}$ can be written as

$$\boldsymbol{D}_1^{-1/2}\boldsymbol{A}\boldsymbol{D}_2^{-1/2}\boldsymbol{v} = (1-\lambda)\boldsymbol{u}$$
$$\boldsymbol{D}_2^{-1/2}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{D}_1^{-1/2}\boldsymbol{u} = (1-\lambda)\boldsymbol{v}.$$

By the singular value decomposition (SVD) of the normalized matrix $\boldsymbol{A}_n = \boldsymbol{D}_1^{-1/2}\boldsymbol{A}\boldsymbol{D}_2^{-1/2}$, the graph partitioning problem can be reduced to compute the left and right singular vectors of $\boldsymbol{A}_n$

$$\boldsymbol{A}_{\mathbf{n}}\boldsymbol{v}_2 = (1-\lambda)\boldsymbol{u}_2, \quad \boldsymbol{A}_{\mathbf{n}}^{\mathbf{T}}\boldsymbol{u}_2 = (1-\lambda)\boldsymbol{v}_2.$$

$(1-\lambda)$ is the singular value, while $\boldsymbol{u}$ and $\boldsymbol{v}$ are the corresponding left and right singular vectors, respectively.

In order to get the k partitions, the $l = \lceil \log_2 k \rceil$ singular vectors of $\boldsymbol{A}_n$, (i.e., $\boldsymbol{u}_2, \boldsymbol{u}_3, \ldots, \boldsymbol{u}_{l+1}$, and $\boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_{l+1}$) can be used to form a l-dimensional matrix:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{D}_1^{-1/2}\boldsymbol{U} \\ \boldsymbol{D}_2^{-1/2}\boldsymbol{V} \end{bmatrix}$$

where $\boldsymbol{U} = [\boldsymbol{u}_2, \boldsymbol{u}_3, \ldots, \boldsymbol{u}_{l+1}]$ and $\boldsymbol{V} = [\boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_{l+1}]$. With $\boldsymbol{Z}$, the story clusters and related feature clusters are formed by grouping the similar vectors (rows) of $\boldsymbol{Z}$. The partition of the first part of $\boldsymbol{Z}$ can group the related features together, while the second part of $\boldsymbol{Z}$ can cluster the stories. The clustering is performed by k-means on matrix $\boldsymbol{Z}$ to get k clusters in traditional co-clustering algorithms.

In our previous work [33], to construct the visual feature, keyframes are firstly clustered by k-means on color histograms and intuitively each cluster represents a visual concept. Nevertheless, such representation cannot fully explore the visual cues inherent in stories. This is simply because the color histograms alone are incapable of capturing the semantic aspects of a visual concept. Moreover, after normalized cuts, k-means is performed to obtain the k-way co-clustering. However, the k-means algorithm is known to be sensitive to outliers and often inaccurate in the initial guess of cluster centroids. In addition, the assumption (convex-shape cluster) and optimization (minimizing the sum of squared error) violate the nature and property of topic clusters.

Therefore, in the new approach, we replace the visual feature construction (color histograms) with the NDK constraints, and replace the noise-sensitive k-means step in the traditional CC with the newly proposed CB-DBSCAN. Here we derive visual features using information from NDK. Basically the same NDK form a cluster, while non-NDK form clusters of one keyframe each. A potential issue with such setting is whether non-NDK, which can potentially act as noise and the number is much larger than NDK, could affect the performance of co-clustering. Further post-processing such as capturing the relation between words and keyframes could be useful, although is not considered here.

The association between a story and a keyframe cluster is computed by weight W_{ij}

$$W_{ij} = kf_{ij} \times \log_2\left(\frac{N}{sf_i}\right)$$

where kf_{ij} is the frequency of keyframe cluster i in story j; N is the number of news stories; and sf_i is the story frequency of cluster i. For clusters formed by non-NDK, $kf_{ij} = sf_i = 1$, and thus $W_{ij} = \log_2 N$. With W_{ij}, the keyframe-by-story association is formed, and denoted as $\boldsymbol{A}_1$, in which rows correspond to keyframe clusters while columns refer to stories. The word-by-story association $\boldsymbol{A}_2$ is constructed in a similar way with typical *tf-idf* function. Initially a feature-by-story matrix is

formed with $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2]^T$. CC is then solved with normalized cut algorithm.

Furthermore, k-means is sensitive to noises and frequently miss-identifies the initial centroids. To reduce the randomness caused by k-means, we explicitly add NDK constraints to $\boldsymbol{X} = \boldsymbol{D}_2^{-1/2}\boldsymbol{V}$ (which is the second part of matrix $\boldsymbol{Z}$, i.e., story part) to guide the clustering. In this matrix, each row corresponds to a story and the pairwise constraints can be directly added onto the matrix. Instead of using k-means, the vectors in $\boldsymbol{X}$ are clustered with the constraint-based DBSCAN (CB-DBSCAN). Basically each vector is a story and represented as a point in the multidimensional space. Two points are connected with a must-link constraint if NDK are found. An interesting point to note is that, with constraints, we can effectively locate the cluster center and adapt the radius of DBSCAN to grow a cluster with varying densities—an important feature not found in the original DBSCAN algorithm. With this radius adaptation property, we can handle story clusters of different shapes, sizes, and densities, while discovering outliers.

IV. Story Clustering With Constraint-Based DBSCAN

DBSCAN [13] is a typical density-based clustering algorithm, which is designed to discover clusters of arbitrary shapes and sizes. The algorithm is popularly adopted in the data mining field. DBSCAN requires two critical parameters: *Eps* and *Minpts*. *Eps* defines the maximum radius of neighborhood, and *MinPts* is the minimum number of points in an *Eps* neighborhood. Both parameters are fixed. The algorithm starts by selecting an arbitrary point and retrieves all points that are density-reachable from this point as a cluster. With constraints, CB-DBSCAN can easily locate centers and adapt *Eps* to achieve better performance. However, in order to avoid the poor performance caused by improper seed initialization, we select the constrained core point in the densest part as the starting point instead of selecting arbitrary point. This is simply because *Eps* cannot be decided at points near the cluster border. In addition, when *Eps* can vary, all points can rival for different cluster assignments under different *Eps* settings. With the information of constraints, the clusters can be formed by expanding the points density-reachable from the initial points.

A. Constraint Propagation

Near-duplicate keyframes provide strong cues for linking related stories together. If two news stories have at least one pair of NDK, they can be regarded as having a must-link constraint, which indicates that they discuss the same topic. A must-link constraint between two news stories implies that they should be assigned to the same cluster. With these constraints, chunklets (groups of points with must-link constraints) are formed, indicating the stories in the chunklet must be in one cluster. In this paper, story, object, and point refer to the same thing. Note that $\boldsymbol{S}$ denotes the set of stories, while S_i means the ith story.

Definition: (*Must-link constraint*) There is a *must-link constraint* between story S_a and story S_b if there exists at least one near-duplicate keyframe pair between these two stories, formally, $S_a \longleftrightarrow S_b \Longleftrightarrow \{\exists KF_{ai} \in S_a, \exists KF_{bj} \in S_b, \text{ and } KF_{ai} \approx KF_{bj}\}$, where $\approx$ denotes that keyframes are near-duplicate, and KF_{ai} refers to the ith keyframe in the story S_a.

Fig. 3. Propagating constraints to form a chunklet of stories.

Definition: (*Constrained chunklet*) A *constrained chunklet* CK_i is a nonempty subset of $\boldsymbol{S}$ satisfying the condition: $\forall\, p, q$: if $p \in CK_i$ and q has must-link constraint with p (i.e., $S_q \longleftrightarrow S_p$), then $q \in CK_i$;

The must-link constraint exhibits the symmetric property. If $S_a \longleftrightarrow S_b$, then $S_b \longleftrightarrow S_a$. A must-link constraint between two news stories implies that they should be assigned to the same cluster. The constraint chunklet define a transitive binary relation over the stories. Consequently, when making use of a set of constraints, we take a transitive closure over the constraints. More specifically, if story S_1 is linked to story S_2, and S_2 is linked to S_3 by near-duplicate keyframe pairs, (i.e., $S_1 \longleftrightarrow S_2$, $S_2 \longleftrightarrow S_3$), then these three stories are grouped as a chunklet in this case, which is demonstrated in Fig. 3. Based on the constraints and the transitive closure, we can propagate the constraints to form a set of chunklets. Note that the transitive closure is only available for forming the chunklet, for other cases, must-link constraints denote the constraints without transitive closure.

B. Center Selection

There are two kinds of points in a cluster, points inside the cluster (core points) and points on the border of the cluster (border points). In general, the neighborhood of a border point contains significantly fewer points than the neighborhood of a core point. To avoid initializing clusters starting from points that are likely to be outliers or lie along the cluster border, we choose the point near the center of the chunklet as the seed.

For each chunklet, we can get a virtual center (centroid), and then locate the medoid that is closest to the centroid and satisfies the core point condition as the starting point. By doing so, the medoid has higher possibility to be from the densest part. It is consistent with the fact that there is at least one core event in a topic, with other events inside the topic related to it and visual information of other stories is inherited from it. Other points in the cluster can be further retrieved by expanding the center.

Definition: (*Constraint neighborhood of a story* S_a) The *constraint neighborhood CN* of a story S_a is the set of stories that have must-link constraints with the story S_a. $CN(S_a) = \{S_b | S_a \longleftrightarrow S_b\}$.

Definition: (*Core object*) A story $o \in \boldsymbol{S}$ is called a *core object* w.r.t. k, if it contains at least $k-1$ constrained stories, formally: $\text{Core}(o) \Longleftrightarrow |CN(o)| \geq k-1$. This is also the core point condition.

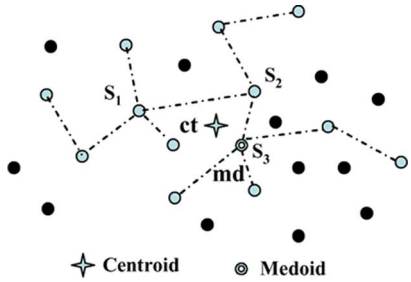


Fig. 4. Centroid and medoid.

Definition: (*Centroid of a chunklet*) For each chunklet CK_i, a *centroid* ct is the average value of all stories in this chunklet. Each story corresponds to one row in the l-dimensional matrix $\boldsymbol{X}$ after a normalized cut. $ct = (z_0, \ldots, z_{l-1})$:

$$z_i = \frac{\sum_{j=1}^{|CK_i|} x_i}{|CK_i|}, \text{ where } S_j \in CK_i, \quad S_j = (x_0, \ldots, x_{l-1}). \tag{1}$$

Definition: (*Distance of two stories*) The distance of two stories is based on the popular *Euclidean* distance, where $S_i = (x_0, \ldots, x_{l-1})$, and $S_j = (y_0, \ldots, y_{l-1})$

$$d(S_i, S_j) = \sqrt{\sum_{k=0}^{l-1} (x_k - y_k)^2}. \tag{2}$$

Definition: (*Medoid of a chunklet*) A *medoid md* is a real story in the chunklet which is the story nearest to the centroid and it satisfies the core point definition (i.e., the number of constrained neighborhood (*CN*) of this real object is no less than $k-1$). This strategy avoids the possibility that a medoid is chosen from an outlier or the border of cluster

$$md = \arg\min_{S_j \in CP} d(ct, S_j) \tag{3}$$

where $CP = \{S_i | S_i \in CK_i \wedge |CN(S_i)| \geq k-1\}$ is the core point set.

Fig. 4 shows an example about the centroid and medoid. The green points form a chunklet, and its centroid is labeled as ct, which does not correspond to a real story. There are three objects (S_1, S_2, and S_3) as the candidate real medoids because each point satisfies the core condition which has at least three constrained neighborhood objects when $k = 4$. The nearest point to the centroid will be selected as the medoid, that is, S_3 is the md. Note that a medoid must be a core object.

C. Radius Adaptation

Traditional DBSCAN uses a fixed radius to grow clusters. There is a simple heuristic to determine the parameters *Eps* and *MinPts* of the "thinnest", i.e., least dense cluster in the database. DBSCAN uses global values for *Eps* and *MinPts*, that is, the same values for all clusters. The density parameters of the "thinnest" cluster are good candidates for these global parameter values specifying the lowest density which is not considered to be noise [13]. However, in story clustering, the densities and shapes of clusters are different. Even in a cluster, the densities and radius scales can vary. The improper selection of the radius will result in poor clustering performance. If the radius is too small, most stories can simply become outliers. While if the radius is too large, a cluster may include outliers as well as stories from other clusters. Choosing an appropriate radius in DBSCAN becomes crucial. In CB-DBSCAN, the radius is adaptive under the guidance of constraints.

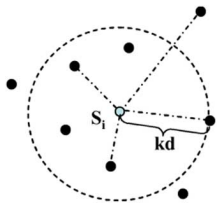


Fig. 5. k-distance of story S_i ($k = 4$).

Definition: (k-*distance of a story* p) For any positive integer k, the k-*distance* of a story p, denoted as $kd(p)$, is defined as the distance $d(p, o)$ between p and a story o that is constrained with p such that:

(i) there exists at least k stories o' with condition $o' \in \boldsymbol{S} | d(p, o') \geq d(p, o) \wedge p \longleftrightarrow o \wedge p \longleftrightarrow o'$;

(ii) there exists at most $k-1$ stories o' with condition $o' \in \boldsymbol{S} | d(p, o') < d(p, o) \wedge p \longleftrightarrow o \wedge p \longleftrightarrow o'$.

The definition of k-distance is stricter than the traditional definition in DBSCAN. It guarantees the points within the k-distance include at least k points and these k points must be in the same cluster with the object p.

Definition: (k-*distance neighborhood of a story* p) Given the k-distance of p, the k-*distance neighborhood* of p, denoted by $N_{kd}(p)$, is defined by: $N_{kd}(p) = \{q \in \boldsymbol{S} | d(p, q) \leq kd(p)\}$.

Fig. 5 gives an example of the k-distance of story S_i. S_i is a core object because it has four constrained stories. The constraints are labeled by dashed lines. When $k = 4$, the k-distance of S_i is the distance between S_i and the third nearest constrained story. All points within the range of the dashed circle are the k-distance neighborhood of S_i.

In CB-DBSCAN, with the assistance of constraints from NDK, the radius of clusters is easier to determine and can be dynamically adjusted. A point that has at least $k-1$ constrained neighborhood means that this point is a core center, and other points are closely related to it. So the k-distance can be treated as the candidate radius. However, the distances among these constrained neighbors are inconsistent. Some stories are closely connected, while others may be sparsely related. To tolerate constraint inconsistency, the radius is designed to vary with respect to the average density of the cluster and the current density of the core point. Fig. 6 shows an example that a point at the border of a cluster has $k-1$ constrained neighborhood. Its density is lower than the center of the cluster, but it still satisfies the definition of the core point, so this point will retrieve its density-reachable points. It has a large radius and will include some points from other clusters. To make the radius adaptive to the current density, we record the history of the radius (average k-distance of previous core points), and then compare the k-distance of the core point with the average radius. The smaller one is picked as the value of radius R, i.e.,

$$R = \min(kd(o), \text{Avg\_radius}). \tag{4}$$

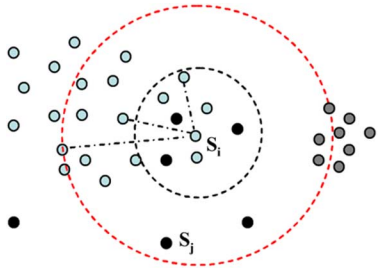


Fig. 6. Adapting radius to smaller circle at cluster border.

Fig. 6 gives an example of the radius adaptation. We can see that although story S_i is at the border of the cluster, it still satisfies the core point condition. When S_i has the chance to expand the cluster, the radius will be determined by the k-distance of S_i and the average radius of precious core points, whichever is smaller. In this example, the red circle is with the k-distance as the radius, while the black circle is with the average radius. So the radius will adapt to the smaller one, which effectively avoids inducing false points into the cluster. Without adaptation, S_i would retrieve points within the red circle, which includes the stories from other clusters (gray points) and outliers, because of the large k-distance.

D. Cluster Formation

In CB-DBSCAN, a cluster is defined to be a set of stories density-reachable from a seed story. While outlier is the set of stories in $\boldsymbol{S}$ not belonging to any cluster or belonging to clusters having less than k stories.

Definition: (*Directly density-reachable*) A point p is *directly density-reachable* from a point q w.r.t. R and k if

(i) $d(p,q) \leq R$;

(ii) $|CN(q)| \geq k-1$ (core point condition).

Definition: (*Density-reachable*) A point p is density-reachable from a point q if there is a chain of points $p_1,\ldots,p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Definition: (*Density-connected*) A point p is *density-connected* to a point q, if there is a point o such that both p and q are density-reachable from o.

Definition: (*Cluster*) A *cluster* C_i is a nonempty subset of $\boldsymbol{S}$ satisfying the following conditions:

(i) $\forall\, p,q$: if $p \in C_i$ and q is density-reachable from p, then $q \in C_i$. (Maximality)

(ii). $\forall\, p,q \in C_i$: p is density-connected to q. (Connectivity).

Definition: (*Outlier*) Let $C_1,\ldots,C_m$ be the clusters of the data set $\boldsymbol{S}$, $i = 1,\ldots,m$. Then we define the *outlier* as the set of stories in the dataset $\boldsymbol{S}$ not belonging to any cluster C_i or belonging to clusters but the number of the clusters is less than k, i.e., outlier $= \{p \in \boldsymbol{S} | \forall\, i : p \notin C_i || p \in C_i \text{ but } |C_i| < k\}$.

To find a cluster, initially a point in the dense region is selected from the data set satisfying the core point condition as a seed. Secondly, we retrieve all points that are density-reachable from the seed. Our algorithm starts from a medoid which is closest to the centroid and satisfies the core point condition. The radius is decided by the current k-distance of the core point and the average radius, whichever is smaller. At each iteration, the radius adaptation is based on the average density and the density of the current core point. The points within the radius will be included in this cluster unless the point has been assigned to other chunklets. If the constrained neighborhood of a newly added point is greater than $k-1$, this point is another core point. It will continue to expand the cluster until there are no more core points. This procedure yields a cluster.

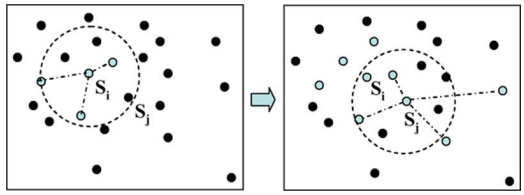


Fig. 7. Sprawling a cluster from object S_i to S_j.

Fig. 7 demonstrates an example to sprawl a cluster. At first, it starts from the medoid S_i. Based on its k-distance ($k = 4$), the points within the black circle are included into the cluster (labeled as green points). The point S_j is a newly added point and it satisfies the core point condition. The radius is recalculated based on the history of the k-distances and the k-distance of the current core point. Then, a new radius is achieved and another circle is formed. New points falling into the radius are regarded as the elements of the cluster. The cluster can further sprawl until there are no more core points.

ALGORITHM I
CONSTRAINT-DRIVEN CO-CLUSTERING ALGORITHM (CCC)

Algorithm: Constraint-driven Co-Clustering algorithm (CCC)

Input: Story boundary, word list, keyframe list and near-duplicate keyframe list

Output: Clusters and outliers

Procedure:

I. Construct bipartite graph $\boldsymbol{G}$ and generate reduced data matrix $\boldsymbol{X}$ with normalized cut.

II. Identify chunklets by propagating pairwise constraints that come from near-duplicate keyframe information with transitive closure.

III. For each chunklet, calculate the centroid and locate the medoid *md* which satisfies the core point condition.

IV. For un-clustered chunklets, select a chunklet with minimum k-distance $kd(md)$ to sprawl the cluster. For each iteration during sprawling:

a) Adaptively adjust the radius with **(4)**;

b) Retrieve density-connected points of a core object as stories in this cluster.

V. CCC Algorithm

The CCC algorithm under density notion is listed in Algorithm I. In Phase-I, a bipartite graph $\boldsymbol{G}$ is modeled to represent the relationship between stories and their textual and visual features. Near-duplicate keyframe constraints are first used to construct the association between the stories and keyframes. After a normalized cut, a reduced dimensional matrix on stories ($\boldsymbol{X}$) is formed as the inputs of later phases. In this matrix $\boldsymbol{X}$, each row refers to one story.

By propagating the pairwise constraints with transitive closure, stories with constraints from NDK are grouped into chunklets in Phase-II. Pairwise constraints are further utilized to drive the cluster initialization and parameter selection of CB-DBSCAN. One critical consideration is to avoid initializing clusters starting from points that are likely to be outliers or lie along cluster border. Phase-III guarantees the correctness of the initialization problem starting from a dense region. For each chunklet, a centroid is calculated with the existing stories in this chunklet, and then a medoid *md* nearest to the centroid and satisfies the core point condition is located as the seed of the chunklet.

To guarantee the robustness of CCC, we prioritize the order of forming clusters. CB-DBSCAN is incremental in that the points are drawn in the order of the confidence hinted by must-link constraints. In Phase-IV, clustering is started by identifying the chunklet in which the centroid has the smallest k-distance. We actually utilize this property to initialize clusters with high density (i.e., low radius) in an incremental fashion, which avoids the case that a cluster with a large radius will overlook smaller radius clusters due to improper radius selection.

From the core point, the points density-reachable from it are retrieved. In CB-DBSCAN, *Eps* and *Minpts* are set to R and k respectively. They are then used to crawl all density connected points of the core point to form a cluster. Traditionally, DBSCAN is excellent for identifying clusters with different shapes but not densities. In our case, the radius of a cluster is varied depending on R. To cope with the varying radius scales, the radius for each core point is adapted based on (4). This special feature enables our proposed CCC to cope with clusters of various densities and shapes. The unclustered points and the clusters having less than k points are regarded as outliers.

The time complexity of the CCC algorithm depends on the complexity of normalized cuts and CB-DBSCAN. For the normalized cut, solving a standard eigenvector problem takes $O(m^3)$ where m is the number of nodes (stories and their textual and visual features) in the bipartite graph. However, due to the fact that the eigen systems are very sparse and only the few top eigenvectors are needed, the complexity can be reduced to $O(m^2)$ [28]. The complexity of CB-DBSCAN is nearly the same as DBSCAN due to their structural equivalence. Similar to DBSCAN, its complexity is dominated by the run-time of k-distance and k-distance neighborhood which must be performed for each story in the database. To retrieve the k-distance neighborhood of a story, a scan through the whole database has to be performed. So the computational complexity of CB-DBSCAN is $O(n^2)$ where n is the number of stories. Therefore, the resultant complexity of CCC is $O(m^2) + O(n^2)$ where m and n are the number of nodes (stories and their features) and stories respectively. Furthermore, with the aid of normalized cut, CCC is potentially more efficient than traditional DBSCAN since it operates in a considerably lower dimensional space $O(\log(n'))$ where n' is the number of clusters, instead of the full dimensional space $O(m')$ as DBSCAN where m' is the total number of textual and visual features in the dataset.

VI. Experiments

A. Dataset and Evaluation

We select one month's interval (1998-03-02 to 1998-03-31) from TRECVID-2004 corpus [31] as our test set, which includes 52 CNN and ABC news videos. The common story and shot boundaries, defined by TRECVID, are used as the basic units of analysis. The text features are derived from a list of words extracted from speech transcripts by an automatic speech recognition system (ASR) at LIMSI [16], while the visual concepts consist of the set of representative keyframes extracted from video corpus. Note that each shot is represented by one keyframe. The representative keyframes of shots are given. The set of duplicate and near-duplicate keyframes is labeled manually. The keyframes with anchor person has been removed and they are not treated as must-link constraints. After data preprocessing such as word stemming and stop-word removal, the data set is comprised of 805 news stories, 7006 keyframes, and 7028 words. There are 1833 near-duplicate keyframes in our test set, which form 627 groups. In each group, the near-duplicate keyframes are very similar but varied in terms of viewpoints, lighting conditions, and editing operations. Without the official annotation of topics, we built a ground-truth table by manually labeling stories according to topic themes. Two nonexpert assessors are asked to watch news stories ordered chronologically and to annotate the topic themes. They are requested to give a judgment of the topic theme for each news story. To ensure the fairness of comparison, the topics having less than four news stories are regarded as outliers. In total, there are 29 topics plus 241 outlier stories.

In broadcast news, in addition to the traditional event-related topics, such as "Arkansas school shooting" and "Yeltsin fired the whole cabinet", there are some general nonevent topics, such as "Weather", "Sports" (e.g., "Ice", "Basketball"), "Health", and "Stocks". Different from specific event-related topics, nonevent topics are usually general topics with coarse granularity, in which stories under these topics are loosely connected and cover different subthemes. General topics are not further detailed because they contain few repeating stories and the duration of stories is short. For instead, further labeling the topic "Basketball" into few more specific topics according to game only results in few stories with very short duration (few seconds). Out of 29 topics, there are eight general topics included in the experiments. In Table II, the first ten topics are specific topics while the last five topics are general ones. We do not consider commercials as topics in the experiments.

We use *F-measure* and *Entropy* [30] as the metric for performance evaluation. F-measure assesses the quality of clusters by comparing the detected clusters with the ground-truth clusters.

TABLE I
PERFORMANCE COMPARISON

	K-MEANS	DBSCAN	CC_W	CC_W_KF	CK-MEANS	CCC
F-Measure	0.348	0.339	0.338	0.392	0.426	0.562
Entropy	4.603	2.812	2.355	1.791	1.162	0.825

TABLE II
PERFORMANCE COMPARISON (PRECISION) OF DIFFERENT CLUSTERING METHODS (#: NUMBER OF STORIES)

ID	Topic	#	K-MEANS	DBSCAN	CC_W	CC_W_KF	CK-MEANS	CCC
1	Iraq nuclear weapons	19	8 / 80	18 / 307	10 / 19	8 / 80	15 / 38	10 / 12
2	Clinton sexual scandal	58	33 / 63	49 / 307	33 / 58	33 / 63	44 / 98	45 / 57
3	Clinton visited Africa	21	7 / 43	17 / 307	8 / 21	7 / 43	14 / 98	9 / 12
4	Dianna accident	7	4 / 63	7 / 307	3 / 7	4 / 63	7 / 98	3 / 3
5	Yeltsin fired the cabinet	8	5 / 80	8 / 300	5 / 8	5 / 80	8 / 15	7 / 7
6	Arkansas school shooting	37	15 / 80	34 / 307	20 / 37	15 / 80	28 / 68	30 / 55
7	El Nino	38	19 / 25	7 / 7	22 / 38	19 / 25	17 / 20	13 / 13
8	Army sexual misconduct	14	11 / 28	14 / 307	5 / 14	11 / 28	14 / 98	10 /13
9	Titanic	13	8 / 94	4 / 78	6 / 13	8 / 94	7 / 13	7 / 8
10	Hospital killed patients	6	5 / 46	6 / 307	5 / 6	5 / 46	4 / 57	5 / 5
11	Ice	25	5 / 56	10 / 78	10 / 25	5 / 56	23 / 107	23 / 23
12	Health	32	15 / 21	32 / 300	20 / 32	15 / 21	15 / 18	29 / 71
13	Basketball	78	24 / 56	45 / 78	38 / 78	24 / 56	62 / 107	71 / 109
14	Weather	28	26 / 26	26 / 35	25 / 28	26 / 26	4 / 57	27 / 28
15	Stock	31	13 / 31	26 / 37	13 / 31	13 / 31	17 / 41	13 / 14

Let $\boldsymbol{\mathcal{G}}$ be the set of ground-truth clusters and $\boldsymbol{\mathcal{D}}$ be the detected ones. The F-measure FM is

$$\mathbf{F}M = \frac{1}{\boldsymbol{H}} \sum_{\boldsymbol{C}_i \in \boldsymbol{\mathcal{G}}} |\boldsymbol{C}_i| \max\ \boldsymbol{C}_j \in \boldsymbol{\mathcal{D}}\{F(\boldsymbol{C}_i, \boldsymbol{C}_j)\}$$

$$F(\boldsymbol{C}_i, \boldsymbol{C}_j) = \frac{2 \times \text{Recall}(\boldsymbol{C}_i, \boldsymbol{C}_j) \times \text{Precision}(\boldsymbol{C}_i, \boldsymbol{C}_j)}{\text{Recall}(\boldsymbol{C}_i, \boldsymbol{C}_j) + \text{Precision}(\boldsymbol{C}_i, \boldsymbol{C}_j)}$$

where $\text{Recall}(\boldsymbol{C}_i, \boldsymbol{C}_j) = |\boldsymbol{C}_i \cap \boldsymbol{C}_j|/\boldsymbol{C}_i$ and $\text{Precision}(\boldsymbol{C}_i, \boldsymbol{C}_j) = |\boldsymbol{C}_i \cap \boldsymbol{C}_j|/\boldsymbol{C}_j$. The term $\boldsymbol{H} = \sum_{\boldsymbol{C}_i \in \boldsymbol{\mathcal{G}}} |\boldsymbol{C}_i|$ is a normalized constant that indicates the sum of stories from each cluster $\boldsymbol{C}_i$. The value of FM ranges from 0 to 1. The higher FM is, the better the clustering performance is.

Entropy measures the homogeneity of clusters. The higher the homogeneity of a cluster, the lower the entropy is, and vice versa. For every cluster j in the clustering result $\boldsymbol{\mathcal{C}}$, we compute p_{ij}, the probability that a member of cluster j belongs to ground-truth cluster i. The entropy of each cluster j is calculated using the standard formula $E_j = -\sum_i p_{ij} \log(p_{ij})$, where the sum is taken over all ground-truth clusters. The total entropy for a set of clusters is calculated as the sum of entropies for each cluster weighted by the size of each cluster

$$\boldsymbol{E}C = \sum_{j=1}^{m} \left(\frac{\boldsymbol{N}_j}{\boldsymbol{N}} \times \boldsymbol{E}_j \right)$$

where N_j is the size of cluster j, m is the number of clusters, and N is the total number of data objects.

Basically, we would like to maximize the F-measure, and minimize the entropy of clusters to achieve high quality clustering.

B. Performance Comparison

To evaluate the performance, we compare our algorithm (CCC) against traditional text-based k-means algorithm on the original full-dimensional word-by-story matrix (K-MEANS), traditional DBSCAN on text features (DBSCAN), the spectral clustering algorithm (co-clustering) with only textual information (CC_W), with the combination of textual and visual features (CC_W_KF), and constraint-based k-means (CK-MEANS) [32]. First, the effect of k-means and DBSCAN to the story clustering is examined on the textual area. Then, by comparing the performance of CC_W and CC_W_KF, the improvement of co-clustering is studied when visual information is integrated. Finally, the contribution of constraints is evaluated by investigating CK-MEANS and CB-DBSCAN in CCC. For K-MEANS and co-clustering-based algorithms, the number of clusters is preset to 32, which is approximately equal to the number of topics (29 topics) in our dataset. Due to the sensitivity of k-means, the results might be heavily affected by the initial points. To alleviate this problem, the experiments are repeated with five times and the average performance is reported.

In the CCC algorithm, k (for the k-distance) is the only parameter that must to be set manually. The parameter k impacts the setting of CB-DBSCAN in Phase-IV. According to the analysis in [5], the range of k depends on the allowable size of a cluster. In principle, the lower bound of k should not be less than the minimum allowable size of a cluster, while the upper bound cannot exceed its maximum size. In our case, the upper bound is hard to determine, but the lower bound can be set in the range of 4–6. Thus we can simply set $k = 4$. Specifically, it means a news topic should be composed of at least four stories to make it an interesting subject for reading. For the Phase-I of CCC, we compute $l = \lceil \log_2 32 \rceil = 5$ singular vectors. With the normalized cut, a matrix $\boldsymbol{X}$ with each row representing a 5-D feature vector is formed, where each row corresponds to one story.

The size of the clusters (topics) varies from a couple of stories (e.g., "Diana accident") to over 50 stories (e.g., "Clinton sexual

scandal"). The density of stories within the topics is also inconsistent. For example, the stories in topics such as "Weather" are relatively tightly bunched, while the stories in topics such as "Arkansas school shooting" are spread over a wide range, from the school shooting itself to related problems, such as legal issues related to the shooting, American culture, and adolescent education. In addition, some terms (e.g., Clinton) appear across different topics like "Clinton visited Africa", "Clinton sexual scandal". Moreover, there are a large number of outliers (241 stories) in the dataset. These characteristics make story clustering a challenging problem.

Table I shows the performance comparison of six tested approaches in clustering 805 stories into 29 news topics. Overall, CCC outperforms other approaches in terms of both FM and entropy, and achieves the comparatively best performance. Table II shows the detailed results (precision) of 15 news topics. In this table, the recall is not listed because it can be calculated using the first part in each field (correctly detected stories) divided by the number of stories in each topic. For example, the precision of K-MEANS for topic 1 ("Iraq nuclear weapons") is 8/80, so its recall is 8/19 since there are 19 stories in this topic. We first examine the performance of the classic text-based k-means (K-MEANS), text-based DBSCAN (DBSCAN) and co-clustering on text feature (CC_W) to verify the effect of DBSCAN and normalized cut. From Table I, we can see that although they have similar FM values, DBSCAN and CC_W have better entropy, that is, the clusters with DBSCAN and CC_W have better homogeneity. It can be seen from Table II that K-MEANS groups some topics together and forms a big cluster, which consists of 533 stories, while at the same time regards many unrelated stories as clusters. With proper settings, DBSCAN can group density-connected stories into clusters, which improves the homogeneity. However, by applying the same parameters for all clusters, DBSCAN is not effective for clusters with varying densities and sizes. CC_W, with the help of the normalized cut, groups stories into relatively smaller clusters, which improves the homogeneity of clusters. When there is a lot of noise in the story features, it is not difficult to predict that the methods based on k-means perform poorly due to both the sensitivity to the noise and cluster initialization since co-clustering is composed of normalized cut and k-means.

We look at the improvement of co-clustering algorithm when visual information is integrated. Basically the approach with multimodality features (CC_W_KF) outperforms the text-only approach (CC_W). When constructing the relationship between the news stories and their features, NDK provide useful information to link the stories with similar visual shots. CC_W_KF makes the effective use of the interaction between stories, words and visual cues to explore story clustering, which has better FM and Entropy. For topics having near-duplicate keyframes, such as "Basketball", "El Nino", generally, clustering algorithms that combine textual and visual features have better performance than approaches with only textual features. However, the problem due to the deficiency of k-means remains. Some topics like "Iraq nuclear weapons", "Clinton sexual scandal", and "Arkansas school shooting" are still mixed together.

The effect of constraints is also examined by comparing CK-MEANS with K-MEANS. The must-link constraints are especially useful for news story clustering and mining, which provide the capability to connect stories having similar visual duplicates into clusters. Other stories under the same topic are further grouped by identifying their nearest neighbor. Constraints from visual duplicates substantially improve the performance compared to approaches without constraints.

Finally, we tested the performance of our algorithm CCC in which near-duplicate keyframes act as constraints and the density notion is integrated. The experiment is performed on the reduced dimensional matrix derived from normalized cut. So it has the advantage of speed efficiency since they operate in a lower dimensional space (five in our case). The performance is improved from $FM = 0.392$ (by CC_W_KF) to $FM = 0.562$ (by CCC), and from Entropy $= 1.791$ to 0.825. According to these findings, constraints are particularly useful for clustering topics with specific themes (e.g., "Clinton sexual scandal", "Arkansas school shooting"). In these topics, which are usually reported in a limited time frame, NDK frequently serve as a reminder to recapitulate and refresh memory of the topic over time. With the assistance of constraints, a set of chunklets is formed which represent relatively dense topics, and usually cluster centers are roughly within these regions, which effectively alleviates the randomness of cluster initialization caused by k-means. CCC can always select proper initial centers and adaptive radius for these topics to attain better precision and recall than CC_W_F. Most of the hot topics like "Clinton sexual scandal" and "Arkansas school shooting" achieve better performance using CCC than with other approaches. The case that a lot of important topics are mixed together is effectively eliminated. For general topics like "Health", although recall is improved, the precision drops. These topics discuss a wide range of subtopics and many subtopics indeed stay apart in feature space although connected by NDK constrains. In such situation, CCC may include stories from other clusters, although the overall recall is actually boosted. In other cases, CCC usually achieves better performance.

C. Result Discussion

Fig. 8 gives the detailed sprawling process of topic "Clinton sexual scandal". For the iterations not shown in this figure, it means that there is no new story being found. Because the constraints provide useful information for the cluster initialization, the cluster centers and dense parts are easier to locate. For the first few iterations, the most closely related stories are retrieved. For example, in the first round, 21 stories are correctly included while two stories are wrongly detected. Among these 21 stories, 17 stories have constraints, and four stories have no NDK constraints. From Fig. 8, we can see that the density-connected stories are retrieved from the center to the border, and the number of retrieved stories decreases when the core point moves to the border. The radius adaptation scheme effectively reduces the chance of including outliers or stories belonging to other clusters into the topic. Finally, the cluster consists of 46 correctly detected stories (including five stories within the chunklet formed by constraint propagation) plus 11

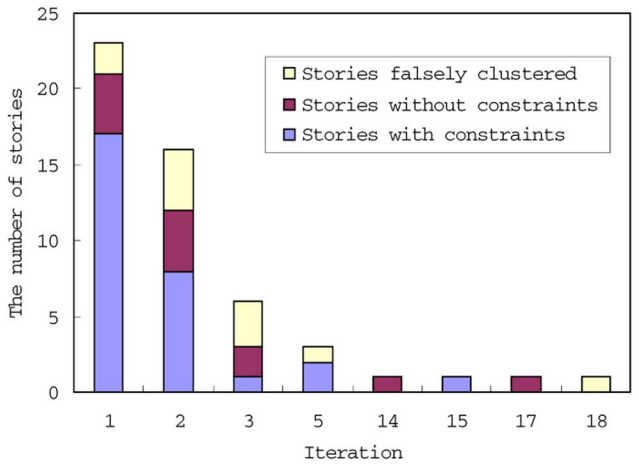


Fig. 8. Sprawling density-connected stories for topic "Clinton sexual scandal".

falsely detected stories, which is considerably better than the clusters detected by other approaches.

Another merit of CCC over other approaches is that it can deal with outliers, while others do not. In the experiments, CCC detected 106 out of 241 outliers. For other approaches, these outliers are scattered into different clusters. Outliers frequently appear in news videos due to the fact that there are large numbers of completely new stories that are not continuously reported. They are relatively sparsely distributed and not tightly related to other stories. Outliers usually demonstrate less textual and visual similarity than stories belonging to clusters. With the assistance of constraints, CB-DBSCAN detects the clusters using the notion of density. The chance of including outliers into the clusters is greatly reduced with the characteristics of CCC such as the density constraint and radius adaptation. Furthermore, outliers cannot initialize the cluster formation. They have no must-link constraints with other stories. Even if there exist NDK within them, the core point requirement cannot be satisfied. Other approaches do not have the ability of outlier detection. Stories are evenly treated and each story is allocated to the nearest cluster. Therefore, outliers cannot be well distinguished, and inliers and outliers are often mixed together.

In this paper, the near-duplicate keyframes are manually labeled. However, it is impossible to be handled as the scale of stories get larger. Example approaches to automatic part-based NDK representation, retrieval and detection can be found in [12], [19], [24], and [38]. Considering the fact that NDK detection can be erroneous, we also test the capability of our algorithm in tolerating the potential errors caused by NDK detection. An experiment is conducted by using the automatic NDK detector in [24]. The F-measure of CCC is 0.429. With reference to Table I, the performance is 23.7% lower than the one with manual labeling. Nevertheless, the result is still better than K-MEANS (23.3% improvement) and DBSCAN (26.5% improvement), which use manual labeling.

VII. Conclusion

In news videos, topic-related stories often use near-duplicate keyframes as reminders to bridge old and new developments of a topic. Near-duplicate keyframes which act as visual-based constraints provide clear hints for story similarity measurement and topic mining. In this paper, we proposed a CCC algorithm by exploring near-duplicate visual constraints under the density notion to mine news topics of varying densities, shapes and sizes. Constraints are utilized to construct the association of stories and keyframes, initialize cluster centers, and dynamically adjust the retrieval radius to sprawl density-reachable stories, which alleviate the burden of parameter selection and provide important clues for grouping stories. Experiments on TRECVID-2004 corpus demonstrate good performance of CCC with the guidance of visual constraints. Visual constraints commonly exist in the context of videos across different sources, times, languages and countries. The advantage is more conspicuous when text information is noisy or unavailable. This work can also be applied to other tasks including video retrieval and mining, video content/copyright management and topic summarization. When a specific video story (shot) is detected, other stories (shots) that having related constraints can also be identified, which can potentially improve the performance of retrieval and mining.

In this paper, we treat pairwise constraints as strict "must-link" restrictions to be satisfied. To allow for the fact that constraints may also be outliers and noisy due to errors in automatic NDK detection, in the future we will explore the case where we relax "hard" NDK constraints to be "soft" constraints in a probabilistic manner. Moreover, due to the emerging popularity of Web 2.0 which includes multimedia data, there are large numbers of duplicate web video clips shared by users on the Internet. Another interesting topic worth exploring is how to exploit constraints from the textual (e.g., title, tags, categories) and visual (NDK) information to mine and discover duplicate clips.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data stream," in *Proc. Conf. Very Large Data Bases*, Germany, 2003.

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

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Conf.*, 1999, pp. 46–50.

[4] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *Proc. ACM Int. Conf. Information Retrieval*, Toronto, Canada, Jul. 2003, pp. 330–337.

[5] M. M. Breungi, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Conf.*, 2000, pp. 93–104.

[6] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proc. SIAM Conf. Data Mining*, 2006, pp. 326–337.

[7] H. Chang and D. Y. Yeung, "Locally linear metric adaptation for semi-supervised clustering," in *Proc. Int. Conf. Machine Learning*, 2004, pp. 153–160.

[8] S.-F. Chang *et al.*, "Columbia University TRECVID-2005 video search and high-level feature extraction," in *Proc. TRECVID*, Gaithersburg, MD, 2005.

[9] D. R. Cutting, D. R. Karger, and J. O. W. T. PedersenJ, "Scatter/gather: A clustering based approach to browsing large document collections," in *Proc. ACM Int. Conf. Information Retrieval*, 1992, pp. 318–329.

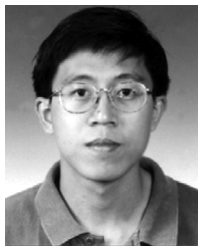
[10] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 2001, pp. 269–274.

[11] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, Washington, DC, Aug. 24–27, 2003, pp. 89–98.

[12] P. Duygulu, J.-Y. Pan, and D. A. Forsyth, "Towards auto-documentary: Tracking the evolution of news stories," in *Proc. ACM Conf. Multimedia*, Oct. 2004, pp. 820–827.
[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial database with noises," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 1996, pp. 291–316.
[14] B. Gao, T.-Y. Liu, X. Zheng, Q. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 2005, pp. 41–50.
[15] B. Gao, T.-Y. Liu, T. Qin, X. Zheng, Q. Cheng, and W.-Y. Ma, "Web image clustering by consistent utilization of visual features and surrounding texts," in *Proc. ACM Conf. Multimedia*, 2005, pp. 112–121.
[16] 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.
[17] T. Hertz, N. Shental, B.-H. Aharon, and D. Weinshall, "Enhancing image and video retrieval: Learning via equivalence constraints," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2003, pp. 668–784.
[18] W. H. Hsu and S.-F. Chang, "Topic tracking across broadcast news videos with visual duplicates and semantic concepts," in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, GA, Oct. 2006, pp. 141–144.
[19] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Conf. Multimedia*, Oct. 2004, pp. 869–876.
[20] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *Proc. Int. Conf. Machine Learning*, 2002, pp. 307–314.
[21] T. Lange, M. H. C. Law, A. K. Jain, and J. Buhmann, "Learning with constrained and unlabelled data," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2005, pp. 731–738.
[22] M. H. C. Law, A. Topchy, and A. K. Jain, "Model-based clustering with probability constraints," in *Proc. SIAM Conf. Data Mining*, 2005, pp. 641–645.
[23] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.
[24] C.-W. Ngo, W.-L. Zhao, and Y.-G. Jiang, "Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation," in *Proc. ACM Conf. Multimedia*, 2006, pp. 845–854.
[25] P. Pantel and D. Lin, "Document clustering with committees," in *Proc. ACM Int. Conf. Information Retrieval*, 2002, pp. 199–206.
[26] G. Qiu, "Image and feature co-clustering," in *Proc. IEEE Int. Conf. Pattern Recognition*, 2004, pp. 991–994.
[27] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing Gaussian mixture models with EM using equivalence constraints," in *Proc. Neural Information Processing Systems*, Istanbul, Turkey, 2003.
[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE. Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[29] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck," in *Proc. ACM Int. Conf. Information Retrieval*, 2000, pp. 208–215.
[30] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. KDD Workshop on Text Mining*, 2000.
[31] TRECVID 2004 [Online]. Available: http://www-nlpir.nist.gov/projects/trecvid/
[32] K. Wagstaff, S. Rogers, and S. Schroedl, "Constrained K-means clustering with background knowledge," in *Proc. Int. Conf. Machine Learning*, 2001, pp. 577–584.
[33] X. Wu, C.-W. Ngo, and Q. Li, "Co-clustering of time-evolving news story with transcript and keyframe," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Amsterdam, The Netherlands, 2005, pp. 117–120.
[34] X. Wu, C.-W. Ngo, and Q. Li, "Threading and autodocumenting news videos," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 59–68, Mar. 2006.
[35] S. X. Yu and J. Shi, "Segmentation given partial grouping constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 173–183, Feb. 2004.
[36] H. Zha, C. Ding, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proc. ACM Conf. Information and Know. Management*, 2001, pp. 25–32.
[37] Y. Zhai and M. Shah, "Tracking news stories across different sources," in *Proc. ACM Conf. Multimedia*, Singapore, Nov. 6–12, 2005, pp. 2–10.
[38] D.-Q. Zhang and S.-F. Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *Proc. ACM Conf. Multimedia*, Oct. 2004, pp. 877–884.
[39] D.-Q. Zhang, C.-Y. Lin, S.-F. Chang, and J. R. Smith, "Semantic video clustering across sources using bipartite spectral clustering," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jun. 2004, pp. 117–120.
[40] X. Zhu *et al.*, "Hierarchical video content description and summarization using unified semantic and visual similarity," *Multimedia Syst.*, vol. 9, no. 1, pp. 31–53, 2003.

Xiao Wu (S'05) received the B.Eng. and M.S. degrees in computer science from Yunnan University, Yunnan, China, in 1999 and 2002, respectively. He is currently pursuing the Ph.D. degree in the Department of Computer Science, City University of Hong Kong.

From 2006 to 2007, he was with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, as a Visiting Scholar. He was with Institute of Software, Chinese Academy of Sciences, Beijing, China, from 2001 to 2002, and was with City University of Hong Kong as a Research Assistant between 2003 and 2004. His research interests include multimedia information retrieval and video processing.



Chong-Wah Ngo (M'02) received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University of Singapore and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 2000.

Before joining City University of Hong Kong in 2002, he was with the Beckman Institute, University of Illinois at Urbana-Champaign. He was also a Visiting Researcher with Microsoft Research Asia in 2002. His research interests include video computing and multimedia information retrieval.

Alexander G. Hauptmann (M'92) received the B.A. and M.A. degrees in psychology from Johns Hopkins University, Baltimore, MD. He studied computer science at the Technische Universität Berlin, Berlin, Germany, from 1982 to 1984, and received the Ph.D. degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, PA, in 1991.

He is a Senior Systems Scientist in the Computer Science Department of CMU and also a faculty member with CMU's Language Technologies Institute. His research interests have led him to pursue and combine several different areas: man-machine communication, natural language processing, speech understanding and synthesis, and machine learning. He worked on speech and machine translation at CMU from 1984–1994, when he joined the Informedia project for digital video analysis and retrieval and led the development and evaluation of the news-on-demand applications.