Keyword-Propagation-Based Information Enriching and Noise Removal for Web News Videos

[1]Jun Zhang [2]Xiaoming Fan [3]Jianyong Wang [4]Lizhu Zhou
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
{[1]zhangjun03, [2]xmFan1983}@gmail.com
{[3]jianyong, [4]dcszlz}@tsinghua.edu.cn

ABSTRACT

The volume of Web videos have increased sharply through the past several years because of the evolvement of Web video sites. Enhanced algorithms on retrieval, classification and TDT (abbreviation of *Topic Detection and Tracking*) can bring lots of convenience to Web users as well as release tedious work from the administrators. Nevertheless, due to the the insufficiency of annotation keywords and the gap between video features and semantic concepts, it is still far away from satisfactory to implement them based on initial keywords and visual features. In this paper we utilize a keyword propagation algorithm based on manifold structure to enrich the keyword information and remove the noise for videos. Both text similarity and temporal similarity are employed to explore the relationship between any pair of videos and to construct the propagation model. We explore three applications, i.e., TDT, Retrieval and Classification based on a Web news video dataset obtained from a famous online video-distributing website, *YouKu*, and evaluate our approach. Experimental results demonstrate that they achieve satisfactory performance and always outperform the baseline methods.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Experimentation, Performance

Keywords

Information Enrichment, Noise Removal, Keyword Propagation

KDD'12, August 12–16, 2012, Beijing, China.

1. INTRODUCTION

Due to the fast advancement of multimedia technology and the great progress of Web 2.0, the volume of Web news videos on online video-distributing websites has been growing explosively through the past several years. Web users upload a large number of videos, share the videos of interests with and recommend them to friends all over the world. *YouTube* [5], one of the most popular online video-distributing websites, according to *YouTube Report 2009* [6] released by *Camscore* (a famous marketing research company in U.S.), has reached 100 million viewers. In January 2009, 14.8 billion online videos were watched by 147 million U.S. Internet users, with an average of 101 videos per person in this month alone. *YouTube* led the growth change, accounting for 91 percent of the incremental gain in the number of videos viewed versus December 2008, as it surpassed 100 million viewers for the first time. In China, there are many YouTube-style video-distributing websites emerged in the past several years, such as *YouKu* [4], *TuDou* [3], etc. *YouKu*, a dominant one of all the counterparts of *YouTube* in China, has increased by 20 times in terms of website traffic last year, and attracted an average of more than 10 million Internet users to visit videos of interests per day in the last three months [1]. Fig.1 shows the two video-distributing websites, i.e., *YouTube* and *YouKu*, and takes some hot news videos as examples, i.e., Beijing Olympic Games and China National Day.

In many cases, rather than surfing websites to read news documents, people prefer to watch Web news videos to know the progress of a hot event or topic. It is not satisfactory to recommend all new-coming unstructured videos to the users, because it is really daunting for the users to view from a video to another to select the hot ones of interests. Furthermore, once a video representing a hot event or a topic is encountered, the users always get curious to know the background, current progress, public opinions and potential trends. Thus tracking hot topics seems much promising to enhance the service of video recommendation. From the perspective of administrators by whom topic structures are manually generated, it is also difficult to do TDT on such a massive video database containing topics of many categories.

Sometimes, Web users submit a query with several keywords to search for the videos of interests. Satisfactory ranking results would come up with the best videos that the user wants most at the top of ranking list. Therefore, an effective and efficient information retrieval system can bring lots of convenience to Web users. Similarly, auto classification for

(a) YouTube (b) YouKu

(c) Beijing Olympics (d) China National Day

Figure 1: Famous video websites and some hot news videos. *YouKu* is a counterpart of *YouTube* in China

Web news videos definitely cuts down sharply the man-hours for administrators when they organize the video dataset and release human efforts for Web users when they try to watch videos of the same classes.

In a word, automatic TDT, Retrieval and Classification for Web news videos can benefit both administrators and Web users.

Actually, research efforts have been devoted to TDT, Retrieval and Classification on text-rich documents. Take TDT for example. The problem was firstly proposed by NIST in 1990s [2, 28, 29, 7], and encouraging results were reported [9, 12, 10]. TDT is a specialization of cluster analysis to facilitate the task of information analysis [25], thus models employed in information retrieval were borrowed to calculate similarity between two documents. Single-pass clustering [28] and hierarchical clustering [10] algorithms are among the most frequently used methods. For Web videos, however, there is much less text information to represent them, and it becomes useless to turn to traditional approaches in a straightforward way.

To survive this challenge in TDT, Retrieval and Classification, many researchers tried to extract semantic concepts from low-level visual features [21, 30, 14], but as we know, it is not a trival task to extract low-level features (e.g., color, texture, shape, etc.) from videos. Moreover, such low-level features are also far from sufficient to represent videos. Although some video search systems [17, 15] utilized not only text information but also other low-level features such as image content, audio, and have improved their performance, meanwhile they also required large amounts of data for training and consumed a large amount of time. In a word, they only achieved minor improvement because of the well-known gap between visual features and semantic concepts. Last but not least, most Web news videos are amateur-made, thus it would be less efficient to extract semantic features from them.

In this paper, given the title, tags, and creation time with any video, we firstly generate keywords with weights from title and tags to initially represent the videos' features. Nevertheless, the keywords of Web news videos are provided by the users with different backgrounds when they uploaded videos. These keywords are not enough to represent all items of interest in a video story. What is even worse, to attract more attention and pageviews or just to do advertisement, misleading or irrelevant keywords are sometimes intermingled with the normal ones by malicious users. As a result, the keywords are always insufficient, limited and noisy. Thus to overcome the difficulties stated above, we employ a manifold-based structure to propagate keywords between any pair of videos about a topic and re-rank them in order to filter out noisy keywords, and enrich keyword information such as involved people, places, and other items of interest.

The key point of our approach is that every video spreads its keywords and weights to their neighboring videos, then its neighbors spread its keywords further to their neighbors. The keyword propagation continues until a global stable state is achieved. Our main contributions are summarized as follows:

- we utilize a manifold-based keyword propagation model to enrich keyword information and remove noise for Web news videos. Two videos are considered as neighbors if and only if they emerge closely in time and share one common keywords at least, and are connected by an edge with a weight value. The manifold is composed of such video points and edges, within which keyword propagation is performed.
- Three applications (Topic Detection and Tracking, Retrieval, and Classification) for Web news videos are performed on the processed videos, and extensive experiments demonstrate that they all achieve significant improvements and outperform the baseline methods.

The rest of this paper is organized as follows. Some related works are discussed in Section 2, and we give some notations in Section 3. In Section 4 we describe the proposed algorithm and framework in details. Section 5 presents the three typical applications, and we analyze the experimental results in Section 6. Section 7 concludes this paper and points out the future work.

2. RELATED WORK

A manifold is a topological space that is typically endowed with a differentiable structure that allows one to measure distances by Euclidean distances. Data points on a manifold structure hold close connections to each other. Furthermore, if two data points were strongly connected to an intermediary data point respectively, they would be considered to have close relationship even if the Euclidean distance between them is large. In other words, the link structure of data points provides us with effective information to understand the environment of data points. Mathematically speaking, by an one-to-one map operation, any pair of data points in the high-dimensional space should be close to each other in a low-dimensional one.

The semi-supervised learning algorithm which employs relationship among data points in [8, 31, 27] is based on the concept of manifold, and achieves significant improvement. Such a learning algorithm satisfies the assumption of local and global consistency, thus the points on the same manifold structure can spread their messages to the neighboring points and reach a stable state. As a result, the idea of propagation model on a manifold has been widely used in several

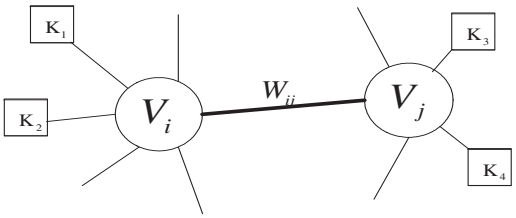


Figure 2: Keyword Propagation Model

research fields, among which are image retrieval, document classification [32, 24, 13, 22], and so on.

Take [31] for example to get more details. As Zhou et al. stated, since it is possible for a data point to spread its labels to those with long distances in space by many intermediary neighbors on the same manifold, the authors designed a classifying function, and generated a toy dataset with a pattern of two intertwining moons in order to evaluate the effectiveness. Classification results on the toy dataset demonstrate that the manifold-based algorithm outperforms the SVM algorithm. Similary in [32], the authors proposed a method to rank on data manifolds, which also achieved higher performance than other baseline methods. This related work encouraged us to develop a manifold-based algorithm for applications of Web news videos. More exhaustive details can be found in the following sections.

3. NOTATIONS

Before we present our keyword propagation model, we first introduce some notations.

Given a news video set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\} \subset R^c$ and a keyword set $\mathcal{T} = \{t_1, t_2, \ldots, t_c\}$, any video $V_i (1 \leq i \leq n)$ has its own keyword set $K_i \subseteq \mathcal{T}$. Let $\mathcal{F}$ denote the set of matrices of $n \times c$ with nonnegative entries. The keyword distribution matrix is defined as an $n \times c$ matrix $F = [F_1^T, F_2^T, \ldots, F_n^T]^T \in \mathcal{F}$, where F_j means the keyword distribution state of the jth video, and each entry of the matrix denotes the corresponding *weight* value. Namely, F can be viewed as a vectorial function $F : \mathcal{V} \to R^c$, which maps a vector F_i to each video v_i. F_{ij} is the weight of the ith video (i.e., v_i) with respect to the jth keyword (i.e., t_j), and it must be a nonnegative number.

Furthermore, $\sum_{k=1}^{c} F_{ik} = 1$, that means the sum of probabilities of the ith video with respect to each keyword should be full. Most of F_{ij}'s are zero because of the sparseness property of F. Each video only has a few keywords of a large keyword set. We define a matrix $Y \in \mathcal{F}$ with $Y_{ij} = w_{ij}$ if v_i has the jth keyword with weight value w_{ij} (defined in Section 4.1) and $Y_{ij} = 0$ otherwise. Generally speaking, Y is consistent with the initial keyword distribution.

As we stated in Section 2, data points on a manifold structure can spread their keywords to the neighboring points. Fig. 2 shows an example of keyword propagation. v_i and v_j are two Web news videos. w_{ij} is the similarity between video i and video j, and it can be regarded as the keyword transition probability from v_i to v_j. In the process of propagation, the model will propagate keywords from v_i to v_j with probability w_{ij}. Excitingly, the keywords of v_i can be propagated to v_k though v_j exactly. The algorithm for keyword propagation will be described in details in the next section.

4. ALGORITHM AND FRAMEWORK

4.1 Pre-Processing

First of all, we tokenize words (word segmentation is performed when dealing with Chinese texts), perform part-of-speech tagging, recognize named entities, remove stop words, and filter out adjective, adverb, etc. Finally a keyword vector is created for each video. To generate a weight for a keyword with respect to each video, we collected 584 days worth of video data from website Youku as a training corpus and compute *video frequency* (a counterpart of *document frequency*) for each keyword, then a weight is computed as Equation 1 which is described in [26].

$$w(t,v) = \frac{tf(t,v)log((|V|+1)/(vf(t)+0.5))}{\sqrt{\sum_{v' \in V} (tf(t,v')log((|V|+1)/(vf(t)+0.5)))^2}} \quad (1)$$

where $tf(t,v)$ represents how many times term t appears in video v, $vf(t)$ represents video frequency of term t in corpus, and $|V|$ is equal to the number of videos in the dataset, that is $2,654,470$ in our experiments. Note that the vector is normalized so that it is of unit length. More detail can be found in [26].

4.2 Manifold-based Propagation Model

As we stated in Section 1, it is essential to filter out noisy keywords and enrich relevant ones to the video, thus we use a manifold-based keyword propagation method inspired mainly from work [31] and [32]. But before diving into more details of the algorithm, we present two prior assumptions that are critically necessary.

As we know, it is a characteristic of news reporting that stories about the same topic often occur in clumps close in time [7], and gradually fade away. Thus it seems that:

1. videos on the same category (event or topic) are likely to share some common keywords, and
2. neighboring news videos that emerge closely in time are more likely about the same topic.

Based on the two assumptions, besides creation time, only keywords obtained from titles and tags are utilized here. Firstly, we need to form a weighted graph based on Web news videos, and to assign a positive similarity score to the edge connecting each pairwise, which is used to construct affinity matrix in our algorithm. To simplify the algorithm, we connect two points with an edge if they are neighbors. It ensures enough connection for each point while preserving the sparse property of the weighted graph.

Given the notations in Section 3, we present the detailed algorithm for the task of keyword propagation in Algorithm 1. Note the algorithm was first proposed in [31] and [32], and has been used for image retrieval in [13] and [24].

In step 1, we establish a connected graph and the distances in Equation 4 are used to calculate the distances between videos. In the second step, we set the diagonal elements of affinity matrix to zero to avoid *self-reinforcement*. The affinity matrix is symmetrically normalized in step 3 and the algorithm really begins to spread keywords among all videos in step 4. In practice, all videos spread their keywords to their neighbors via the linked network. The spread process

Algorithm 1 Manifold-based Keyword Propagation

1. Sort the pair-wise distances ($d(v_i, v_j) = ||v_i - v_j||$) among videos in ascending order. Repeat connecting two videos with an edge according to the order until a connected graph is obtained.
2. Form the affinity matrix W defined by $W_{ij} = exp[-d^2(v_i, v_j)/(2\sigma^2)]$ if there is an edge linking v_i and v_j. Let $W_{ii} = 0$.
3. Symmetrically normalize W by $S = D^{-1/2}WD^{1/2}$ in which D is the diagonal matrix with (i, i)-element equal to the sum of the ith row of W.
4. Iterate $F(t+1) = \alpha SF(t)+(1-\alpha)Y$ until convergence, where α is a parameter in $[0, 1)$ and $Y = F(0)$.

is not stopped until a global state is achieved. By the way, the parameter α and the number of iterations are very important for the algorithm's convergence. Some details will be discussed in Section 6.5.

As we can see from step (2), $W_{ij} \in (0, 1]$, which implies that W is a positive and irreducible matrix, so is S. Since $0 < \alpha < 1$ and the eigenvalue of S in $[-1, 1]$, according to *Perron-Frobenius Theory* [23] and comprehensive discussions in [18, 31, 32, 13], we come to a conclusion that $\{F(i)\}$ converges to

$$F^* = (1 - \alpha)(I - \alpha S)^{-1}Y \tag{2}$$

Although F^* can be expressed in a closed form, we can expand it by Taylor expansion method:

$$\begin{aligned} F^* &= (I + \alpha S + \alpha^2 S^2 + \cdots)Y \\ &= Y + \alpha SY + \alpha S(\alpha SY) + \cdots \end{aligned} \tag{3}$$

From Equation 3, F^* can be regarded as the sum of infinite terms. The first term is the initial (keyword, weight) pairs for each video, the second term is to spread the (keyword, weight) pairs of each video to its neighbors, the third term is to spread the pairs further, and so on. Parameter α specifies the relative amount of the information from its neighbors and its initial information. The spread process is repeated until a global stable state is reached, and all the videos have their own keywords and corresponding weights. It is possible that a video receives dozens of keywords, but in our experiments only the top K dominant words (with high weights) would be reserved to represent the features to speed the propagation of keywords. And the last keywords in sorted keywords list with low weights are removed. Therefore the weight of some important keywords will be increased due to that the neighbors spread these keywords, in contrast, some noisy keywords will be removed due to low weight. Although F^* has a concise expression in Equation 2, for large scale problems, the iteration algorithm expressed in Equation 3 is preferable in considering the computational efficiency. In practice, however, we make some modifications as follows.

First, it is not necessary to calculate all distances between any pair of videos and ensure a totally connected graph. Actually, videos with few keywords in common in different class are far away in space, the graph constructed may contains many strong connected components. Therefore, if two videos are far away in space, keyword propagation seems to bring no effect to change the top K dominant keywords and their relative orders. That is to say, it does no harm to performance if keywords coming last in the keyword ranking list are removed, or even if no propagation is performed for them. Consequently, it may lead to reducible matrices W and S, which results in an iterative process that converges slowly, or even a state of oscillation around a stable dominant value.

Second, instead of the closed form without iterations, we choose the iterative method stated in Equation 3 to work out a final result. Although it is not necessarily equal to the stable F^*, we can get a satisfactory one by giving a constant number of iterations or an acceptable error threshold as the termination condition.

Third, an intuitive distance function incorporating time and keyword information is defined to generate the matrix W. Since it is a characteristic of news reporting that videos about the same event come out in the form of a burst, only those pairs that the creation time gap does not exceed a limit (e.g., N days) are taken into consideration. Meanwhile, σ^2 in step (2) of Algorithm 1 is described as a human-specified parameter, thus we incorporate it into the distance metric to reduce it. Given a weight threshold w_d,

$$d(v_i, v_j) = \sqrt{2}\sigma/(w_C(i, j) - w_d) \tag{4}$$

if the sum of the dot product of common keywords ($C = \{t|t \in v_1 \cap v_2\}$), $w_C(i, j)$, is larger than the given threshold w_d, otherwise $d(v_i, v_j) = \infty$.

5. TYPICAL APPLICATIONS

In this section we will explore three typical applications of the keyword propagation framework proposed in preceding section, i.e., topic detection and tracking, retrieval, and classification. We will evaluate the effectiveness of the proposed model based on the experimental results over a Web news videos dataset obtained from an online video-distributing website, *YouKu*, which contains $2,654,473$ news videos. For any video, only the title, tags, and the creation time are utilized. We will describe the experimental results and discuss them in the following sections.

5.1 Topic Detection and Tracking

5.1.1 Story, Event, and Topich

In [11], the authors described the concept of story, event, TDT and so on. TDT mainly process news streams (either in the form of documents or of videos) and group news reports into different events and topics. The basic units are stories (More detail in [11]). Typically, a news document text or a Web news video is treated as a single story, for example, Figures 1(c) (Beijing Olympics) and 1(d) (China National Day) can be treated as stories. An *event* identifies something (non-trivial) happening in a certain place at a certain time [28], and each event should contain one or more stories that describe it. Obviously, all stories in an event always are close in time and emerge in the form of a burst. Finally, one or more directly related events form a broad *topic*, and it is allowed that there are relatively long time gaps between any two topically similar events. And for simplicity, we say a topic is the *super-unit* of multiple

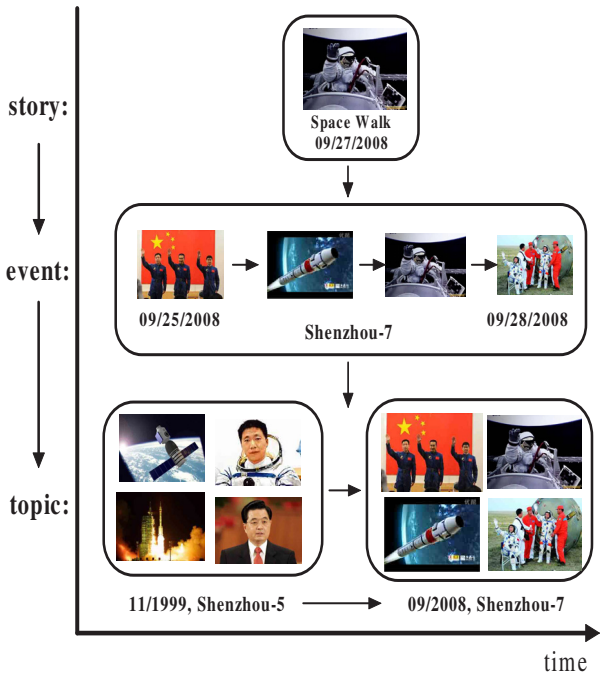


Figure 3: Shenzhou-Series Manned Spacecraft of China

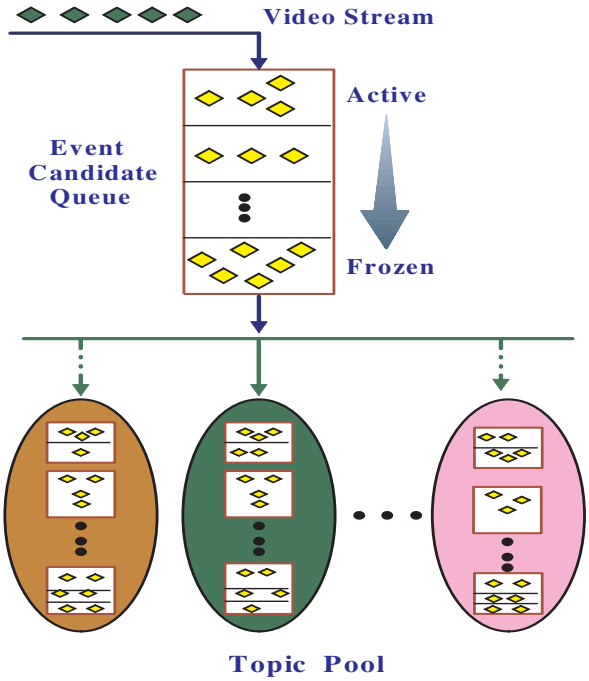


Figure 4: Procedure of Topic Detection and Tracking

events, and an event is the *super-unit* of multiple stories. As a good example, Figure 3 illustrates the three concepts.

News stories discussing the same event tend to be temporally proximate, thus instead of the whole collection, only *local* events are selected to compare with the current story to do *New Event Detection*, which is different from *Topic Tracking* requiring to compare the new event with all historical topics.

5.1.2 Procedure

As we mentioned in Section 5.1.1, an event contains one or more stories close in time, and a topic contains several events. Events are represented as keyword vectors defined as the arithmetic average of keyword vectors of all stories within them, and only the top K dominant keywords are reserved for each event, so are the topics. Cosine similarity in [26] is employed here to calculate the similarities between any two *unit*s. Since the vectors are all normalized, it can be formulated as:

$$similarity(u_1, u_2) = \sum_{t \in u_1 \cap u_2} w(t, u_1) \times w(t, u_2) \quad (5)$$

The procedure of TDT for Web news videos is illustrated in Figure 4. And we explain it more formally step by step in the following:

1. Once a new video arrives (either an original video or an enriched video), calculate the similarities between the video and every event in the *Event Candidate Queue*. If the similarity between the current video and the most similar event is below a given threshold, it indicates that the story is *new* and a new event is detected; otherwise the video will be assigned to the event. With regard to the former case, a new event will be created and pushed into the queue, while in the latter case, we re-calculate the representative keywords for the updated event, and place it at the head of the queue where it is recently activated.

2. Pop out the last event in the *queue*, and judge whether it has been frozen for N days. If yes, it means no story was assigned to the event through the past N days and it is time to remove the event from the *Event Candidate Queue*, and relate it to a historical topic.

3. Once an event needs to be assigned to a topic, we calculate the similarities between the event and every topic in the *Topic Pool*. Note that any event or topic is represented as a keyword vector that has only K dominant keywords, thus it is a trivial task to do the real-time similarity calculation. If and only if the similarity between the event and the most similar topic is larger than the given threshold, it will be assigned to the topic and the topic will be updated, otherwise, a new topic will be generated and inserted into the *Topic Pool*.

5.2 Video Retrieval

Information retrieval aims to find out some objects such as Web pages, images, videos, which are hit by the keywords within a submitted query, and to rank them in order to put the most satisfactory one on the top of the result list. Traditionally, video retrieval task is performed on the initial keywords that come up with the titles, tags. But in this paper, the manifold-based keyword propagation algorithm is used here to enrich the keyword information and remove noisy keywords for Web news videos, and we expect it to work better than traditional baseline methods.

In this experiment, we propose a retrieval problem for Web news videos by selecting several query keywords randomly and evaluating the performance improvements. We implement our system as follows.

1. According to Section 4.2, we spread keywords iteratively until the keyword distribution converges to a stable state. Although it is possible for a video to receive dozens of keywords, only the top K dominant keywords are reserved while the last ones in the sorted keywords list with low weights are considered as noisy ones and removed.

2. Calculate the similarity value between the query vector (keyword vector) and each video vector. And a result ranking list is generated in descending order in terms of both creation time and similarity relevance. In other words, the higher similarity value is, the more relevant with the video is.

Note that we employ cosine similarity to calculate the similarity between a query and a news video as shown in Equation 6. Specifically, given a submitted query q, it is parsed into some text tokens which are considered to form a keyword vector. According to the Eqation 6 we calculate the similarity between the keyword vector of query q and that of a video u.

$$similarity(q,u) = \sum_{k \in q \cap u} w(k,q) \times w(k,u) \qquad (6)$$

Notice that notation k denotes the common words shared by query keywords and video ones.

5.3 Classification

In this section we investigate the task of classification for Web news videos. We select a dataset which contains three categories, i.e., Shenzhou Series (Space Flight), World Cup (Football), Moon Craft (Chang E). After keywords propagation and noise removal, we obtain 2113 video vectors, which are normalized into *TF-IDF* representation as shown in Equation 1.

We explore two popular classification algorithms, KNN and SVM, to handle the classification problem. For KNN and SVM, we firstly do the same as we did in the retrieval task, i.e., keywords propagation. However the following steps in these two methods are not identical. For KNN, we calculate the similarity between the test video and each training video, and then a vote is taken by the most similar K training videos, which assigns a label to the test video. On the contrary, SVM predicts the label of the test videos in a straightforward way.

The similarity between videos v_i and v_j is defined as Equation 5 indicates, and the K in KNN is set to 1. The width of the RBF kernel for SVM is set to 5. The discriminating process repeated several times (with different random samples of training data) and more details are described in Section 6.

6. RESULTS AND DISCUSSIONS

The baseline methods are taken on the original keywords of videos, and are compared with the enriched approaches based on the keywords after propagation. Three applications, i.e., TDT, Retrieval and Classification, are explored on the same algorithms but on different datasets.

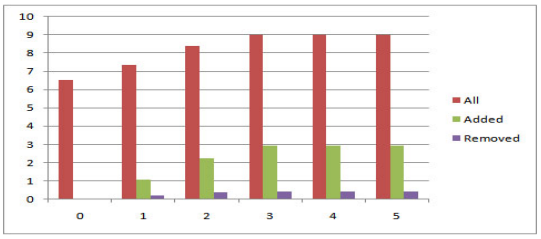


Figure 5: The average number of keywords versus Iteration number of the propagation algorithm.

Firstly, we compare the average number of keywords after propagation with that of the initial ones. As Figure 5 shows, the average keyword number of the initial videos is 6.5 (when iteration number is 0) while the average length of keyword list of videos increases to 9 after propagation. The average number of keywords added and removed are 2.9 and 0.4 respectively. This means that our method enriches information of news videos (by keyword propagation) indeed. In fact, when we check the keyword list of videos after propagation carefully, we find that the algorithm adds some new topic-relevant keywords of videos and also removes some noisy keywords. As the iteration process continues, the average number of keywords increases gradually. Note that the number of keywords would not increase after the iteration number reaches 3, because the relative order of dominant keywords gets stable and arrives at the termination condition at this moment.

6.1 Topic Detection and Tracking

Since whether a video belongs to an event or an event is relevant to a topic is subjective, we invite nine participants to decide which event a story belongs to, and which topic an event belongs to. Every final decision goes to the choice proposed by the majority. Based on the procedure in Figure 4, we performed TDT for the original video stream (Baseline for abbr.), and for the enriched video stream (Enriched for abbr.), respectively. *False Alarm* (*fa* for abbr.), *Miss*, and *Precision* measures [28, 20] are adopted for evaluation.

Firstly we do *New Event Detection* on 93,003 videos that all emerged in year 2007. Since it is not necessary to judge whether any video in year 2007 belongs to a specified event, we collected videos which emerged between the first story and the last story of the current event, and evaluated the effectiveness based on them. In Table 1 we list four bursty events that have been detected. It demonstrates that the proposed method outperforms, or at least be comparable to, the baseline over all four events in terms of *false alarm* and *miss*. Interestingly in Event 3, the baseline does well as the enriched one since both *fa* and *miss* are zero in these two methods, because there are so few videos belonging to this event that it is in vain practically to propagate keywords.

Nevertheless, a topic spans long compared with a bursty event, resulting in the inability to find every topic-relevant video from the massive video stream. Since it is not difficult for us to judge whether a video is relevant to a specified topic, we only adopt *precision* to evaluate the task *Topic Tracking*. Seven prominent topics are listed in Table 2 ("#" denotes the number of videos within the topic). As the figure indicates, the most significant improvement is for *Topic* 7, and the least for *Topic* 6 is about 2.4%. As a result, the

Event	Dominant Keywords (enriched)	Baseline		Enriched	
		miss	*fa*	*miss*	*fa*
1	Womens' World Cup, FIFA, China, Germany, Marta, Brazil	16.7%	8.8%	5.3%	2.6%
2	North Korea, Nuclear Power, Threat, US, Six-Party Talk	25.0%	4.9%	4.9%	3.8%
3	Virus, Nimaya, Safety, Spread, Internet	0.0%	0.0%	0.0%	0.0%
4	Shenzhou-7, Rocket Spacecraft, Test, Astronauts	6.5%	0.0%	2.3%	0.0%

Table 1: Comparison of the Detected Events result between the Baseline and the Enriched

overall *precision* measure increases sharply from 43.0% to 82.3%. In a word, the keyword propagation method helps improve the effectiveness by a large margin in terms of both the number of retrieved videos and the *precision* measure.

Topic	Dominant Keywords	Baseline		Enriched	
		# v	*p* (%)	# v	*p* (%)
1	FIFA World Cup 2006, Goals, Germany, Italy,France, Zidane	301	64.2	356	81.8
2	Yao Ming, Rockets, score, NBA, dunk, McGrady, playoffs	544	87.5	731	90.0
3	Shenzhou Series, Spacecraft, China, Astronaut Yang,rocket	117	72.3	142	87.6
4	Palestine, Israel, conflicts, suicide, bomb, revenge, battle	143	79.6	167	92.2
5	Beijing, Olympics, preparation, ticket, stadium, torch bearer	282	94.0	319	96.9
6	Premier League, ManUtd, Liverpool, Chelsea, Arsenal	317	89.6	346	91.9
7	Iran, US, Bush, Nuclear Plan, sanction, Force, debate	270	43.0	331	72.3

Table 2: Comparison of the Detected Topics result between the Baseline and the Enriched

More interestingly, we find out many videos get associated with some new keywords after propagation, which were not contained initially. For example, before spreading, a video consisted of six keywords, i.e., *NBA, Yao, Rocket, Star, Game, Advertisement*, but they become *NBA, Yao, Rocket, Star, Game, and Basketball*. Obviously the keyword *Basketball* joins the keyword list, and such noisy keywords as *Advertisement* have been filtered out, that does a great favor to detect the relevant event.

6.2 Video Retrieval

In this experiment, we use *precision versus scope* and *recall versus scope* curve to evaluate the performance of the baseline method and the enriched one. Considering the randomness of the selected query keywords, we ran 10 times of ten different queries and recorded each result. Then we computed the average of the results over all the 10 queries.

Firstly the initial retrieval is evaluated. The *precision (recall) versus scope curve* is shown in Figure 6, where the scope stands for the range of the top K retrieval results. In order to perform a systematic evaluation, we varied the size of dataset and compared the *average precision* (P20) and

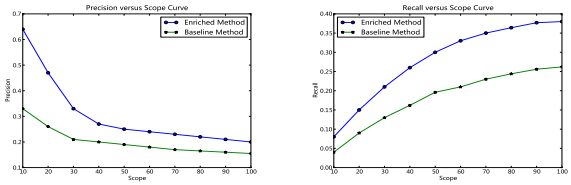
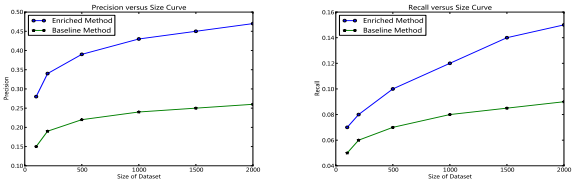


Figure 6: Comparison of the retrieval result between the Enriched method and the baseline method.



(a) (b)

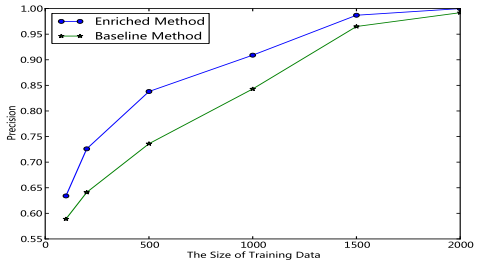
Figure 7: Comparison of the retrieval result between the Enriched method and the baseline method under different size of data. 7(a) P20 versus the percentage of data. 7(b) R20 versus the percentage of training data.

recall (R20) of the top 20 retrieved videos of the enriched method with those of the baseline one. The *precision* (*recall*) versus the size of dataset is demonstrated in Figure 7. As the figures indicate, our propagation method outperforms the baseline method by a large margin. What is more important, the improvement was still very significant even when only a small number of videos were employed.

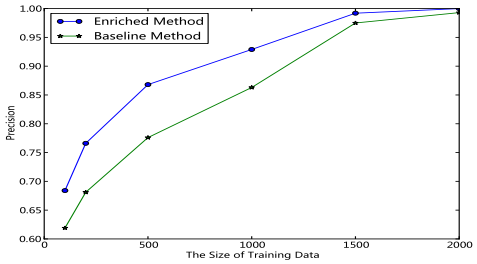
As mentioned in Section 6.1, we expect that the example video without the keyword *basketball* would validate the effectiveness of our method. Actually, it is exciting to find that if we take *basketball* as the query, the example video will be found out while we know the baseline method will not.

6.3 Classification

We explored two classification algorithms, KNN and SVM, to conduct our experiments. For KNN, we firstly calculated the similarity between the test video and the training one, selected the most similar video, and assigned its label to the test one. In other words, the K was set to 1. For SVM, we used keyword vectors with weight as input data, and selected the RBF kernel as the kernel of SVM, which was set to 5. The classifier was expected to predict the label of the test data with three categories. The *precision* is utilized to evaluate the performance and experimental result is shown in Figure 8. We also change the size of training data and compare the *precision* of these two methods. From the experimental results demonstrated in these two figures, it is obvious that our method outperforms the baseline approach significantly.



(a) KNN



(b) SVM

Figure 8: Comparison of the classification result (precision versus the size of training data) between the Enriched method and the baseline method.

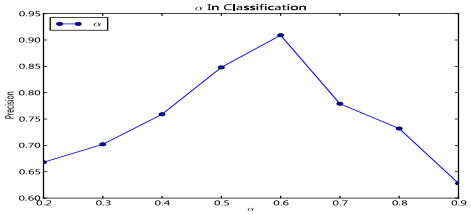


Figure 9: Classification precision versus parameter α in our Enriched method.

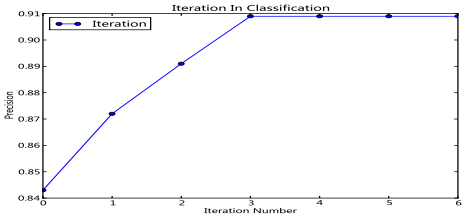


Figure 10: Classification precision versus Iteration number in our Enriched method.

6.4 Time complexity of propagation

A single iteration of propagation (cf. Section 4) consists of a single matrix multiplication $SF(t)$ which is an $O(n^2c)$ operation, where n is the number of videos, c denotes the size of keyword vector, and typically they are about several thousand in the experiments. In our implementation, the keyword distribution converges within a few iterations ranging from 3 to 6, and it takes less than one minute in a regular desktop computer. While for large scale datasets, there are several advanced implementation techniques for matrix multiplication, such as $PowerMethod$ [19].

6.5 Parameter sensibility

6.5.1 α Parameters

The use of α in Equation 3 plays an important role in balancing the initial distribution weight and propagation weight. The value of α also has an impact on the performance and speed of convergence. With α close to 1, the keyword distribution relies almost entirely on the spreading process. Based on our experimental comparisons in Figure 9, the best value for parameter α is 0.6, which is also close to the empirical value in large-scale Web applications [16, 18].

6.5.2 Number of Iterations

The number of iterations is also important for the propagation method, since it has an impact on the performance and the speed of convergence. As the iteration number increases to 3, the keyword distribution gets to a satisfactory and stable state. According to the experimental comparisons in Figure 10, the best value for iteration number is 3, which also means that our method helps the distribution converge to a stable one in a short period of time (less than 1 minute in these experiments).

7. CONCLUSION AND FUTURE WORK

In this paper we have introduced a manifold-based keyword propagation algorithm for Web news videos, in order to enrich the keyword information by adding new topic-relevant keywords and filter out noisy keywords. Both text similarity and temporal similarity are employed to explore the relationship between any pair of videos and to construct the propagation model. Three typical applications, i.e., *Topic Detection and Tracking*, *Retrieval* and *Classification* have been explored in details on a Web news video dataset. Extensive experimental results demonstrate the higher effectiveness of keyword propagation model.

Nevertheless, there are still some issues that are not discussed in this paper. First, we suppose that there is a short time gap between the time when the video's story really took place and the video's creation time, however, it is actually not the case. Thus our future work will focus on how to automatically detect whether they are synchronized, or at least lay in a relatively short period. Second, as the sizes of datasets increase, the dimensions of matrix in propagation model will get much huge and the speed of convergence gets lower. Therefore, to run our model in large-scale video datasets will also be a challenge.

8. ACKNOWLEDGMENTS

This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2011CB302206, and National Natural Science Foundation of China under Grant No. 60833003.

9. REFERENCES

[1] http://www.alexa.com/siteinfo/youku.com.

[2] http://www.itl.nist.gov/iad/mig/tests/tdt/.

[3] http://www.tudou.com.

[4] http://www.youku.com.

[5] http://www.youtube.com.

[6] http://youtubereport2009.com/category/youtube-statistics.

[7] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR'98*, pages 37–45, 1998.

[8] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. In *Machine Learning*, volume 56, pages 209–239, 2004.

[9] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *SIGIR '03*, pages 330–337, 2003.

[10] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. Umass at tdt 2004. In *Working Notes of the TDT-2004 Evaluation*, 2004.

[11] A. Feng and J. Allan. Hierarchical topic detection in tdt-2004. IR 389, University of Massachusetts, 2005.

[12] G. P. C. Fung, J. X. Yu, H. Liu, and P. S. Yu. Time-dependent event hierarchy construction. In *KDD '07*, pages 300–309, 2007.

[13] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *ACM Multimedia*, pages 9–16, 2004.

[14] W. H. Hsu and S.-F. Chang. Topic tracking across broadcast news videos with visual duplicates and semantic concepts. In *ICIP*, pages 141–144, 2006.

[15] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *ACM Multimedia*, pages 971–980, 2007.

[16] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating pagerank computations. In *WWW*, pages 261–270, 2003.

[17] L. S. Kennedy and S.-F. Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In *CIVR*, pages 333–340, 2007.

[18] A. N. Langville and C. D. Meyer. A survey of eigenvector methods for web information retrieval. In *SIAM Review*, volume 47, pages 135–161, 2005.

[19] J. J. Leader. *Numerical Analysis and Scientific Computation*. Addison Wesley, 2004.

[20] L. Liu, L. Sun, Y. Rui, Y. Shi, and S. Yang. Web video topic discovery and tracking via bipartite graph reinforcement model. In *WWW '08*, pages 1009–1018, 2008.

[21] G. Luo, R. Yan, and P. S. Yu. Real-time new event detection for video streams. In *CIKM '08*, pages 379–388, 2008.

[22] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. In *Journal of Machine Learning*, volume 8,(May), pages 935–983, 2007.

[23] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, February 2001.

[24] H. Tong, J. He, M. Li, W. Ma, H. Zhang, and C. Zhang. Manifold-ranking based keyword propagation for image retrieval. volume 21, pages 1–10 Special Issue on Information Mining from Multimedia Database, 2006.

[25] D. Trieschnigg and W. Kraaij. TNO hierarchical topic detection at TDT 2004. Technical report, 2004. TDT 2004 workshop notes.

[26] C. Wang, M. Zhang, S. Ma, and L. Ru. Automatic online news issue construction in web environment. In *WWW*, pages 457–466, 2008.

[27] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *Proceedings of the 23rd international conference on Machine learning*, pages 985–992, 2006.

[28] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. T. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14(4):32–43, 1999.

[29] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *SIGIR'98*, pages 28–36, 1998.

[30] Y. Zhai and M. Shah. Tracking news stories across different sources. In *ACM MM '05*, pages 2–10, 2005.

[31] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, and B. S. Olkopf. Learning with local and global consistency. In *NIPS '03*, pages 321–328. MIT Press, 2003.

[32] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS '04*. MIT Press, 2004.