Real-Time New Event Detection for Video Streams

Gang Luo Rong Yan Philip S. Yu
IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532, USA
{luog, yanr, psyu}@us.ibm.com

ABSTRACT

Online detection of video clips that present previously unseen events in a video stream is still an open challenge to date. For this online new event detection (ONED) task, existing studies mainly focus on optimizing the detection accuracy instead of the detection efficiency. As a result, it is difficult for existing systems to detect new events in real time, especially for large-scale video collections such as the video content available on the Web. In this paper, we propose several scalable techniques to improve the video processing speed of a baseline ONED system by orders of magnitude without sacrificing much detection accuracy. First, we use text features alone to filter out most of the non-new-event clips and to skip those expensive but unnecessary steps including image feature extraction and image similarity computation. Second, we use a combination of indexing and compression methods to speed up text processing. We implemented a prototype of our optimized ONED system on top of IBM's System S. The effectiveness of our techniques is evaluated on the standard TRECVID 2005 benchmark, which demonstrates that our techniques can achieve a 480-fold speedup with detection accuracy degraded less than 5%.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: video, H.3.3 [Information Search and Retrieval]: information filtering, H.3.4 [Systems and Software]: performance evaluation (efficiency and effectiveness)

General Terms

Algorithms, Experimentation, Performance

Keywords

Online new event detection, large-scale video streaming, real-time filtering, efficiency

1. INTRODUCTION

For streaming video, *new event detection* (NED) is the task of capturing the first video clips that present previously unseen events. This task has practical applications in a number of domains such as intelligence gathering, financial market analysis, and news analysis, where useful information is always buried in a large amount of data that grows rapidly with time. Since these applications are often time-critical and require fast turn-around, it is highly desirable to develop an *online new event detection* (ONED) system in practice. For instance, the US government is building a massive computer system that can monitor television broadcasts for anti-terrorism purposes [8, 17], and ONED is one of the most essential components for this system. Moreover, with the rapidly-increasing popularity of user generated content in multimedia sharing Web sites such as YouTube [32], ONED will provide a useful information filtering and retrieval platform for general users to automatically follow interesting stories or to discover new events from one or more large video sources.

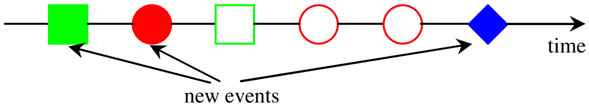


Figure 1. Events in a video stream. Different shapes correspond to different events. Filled shapes represent the clips that need to be captured.

About a decade ago, ONED on document streams started to gain more and more interest in the text processing community [2, 3, 4, 7, 14, 16, 18, 19, 22, 25, 29, 30]. As an extension of its text counterpart, ONED on video streams has also attracted a growing attention in the video processing community by leveraging both text and visual information [10, 11, 15, 28, 31]. The basic idea of video ONED systems is to compare a new clip with all the clips that arrived in the past. If their similarity values based on text and visual features are all below a certain threshold, the new clip will be predicted as presenting a new event. Previous work [11] has shown that additional image information plays an important role in identifying the relevant video clips and achieving better topic tracking results. However, to our best knowledge, all these efforts on video ONED mainly focus on optimizing the detection accuracy instead of the detection efficiency. Actually, these methods yield a quadratic time complexity with respect to the number of clips. Thus, they are not efficient enough to detect new video events in a real-time environment, especially for large-scale video collections. For example, in the intelligence gathering system being developed by the US government [8, 17], tens of thousands of television channels are required to be monitored simultaneously. For YouTube [32], hundreds of thousands of video clips are uploaded every day. In this case, it is very difficult for existing ONED systems to handle such an aggregated and extremely high-bandwidth video stream in real time.

In this paper, we propose several techniques to address the aforementioned efficiency problem and improve the video processing rate of an ONED system by orders of magnitude without sacrificing much detection accuracy. Since the computation on image features is rather time-consuming, we maximize the efficiency of our ONED system by delaying the processing of image features as much as possible. More specifically, we propose the following three optimization steps. First, we use text features alone to filter out most of the non-new-event clips, so that the expensive image feature extraction step of these clips is waived. Then, when comparing the new clip with an old clip, we first compute their text similarity and skip the costly image similarity

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

computation if their texts are sufficiently dissimilar. Finally, we use a combination of indexing and compression methods to speed up text processing. During image similarity computation, we also remove the anchor images to improve the detection accuracy of the ONED system.

We implemented a prototype of our proposed ONED system on top of IBM's System S. As described in Wu et al. [13], System S is a stream processing middleware that provides an application execution environment for processing elements (or applications) developed by users to filter and analyze data streams. Our evaluation on the standard TRECVID 2005 benchmark [26] shows that our techniques can improve the video processing rate by two orders of magnitude (reducing the processing time for the entire video collection from two hours to 15 seconds) without sacrificing much detection accuracy.

The rest of this paper is organized as follows. Section 2 introduces a baseline system. Section 3 analyzes the overall dissimilarity value computation formula of a clip pair. Section 4 presents our techniques for improving efficiency. Section 5 investigates the performance of our techniques. We conclude in Section 6.

2. A BASELINE ONED SYSTEM

Before discussing the proposed techniques in detail, we first describe our baseline ONED system in this section. This baseline system combines the two most influential information sources suggested in the state-of-the-art ONED system reported in Hsu and Chang [11], including TF-IDF text features and binary image duplicate features. Our improvements presented in the rest of this paper are built on this baseline system.

2.1 System Architecture

Figure 2 shows the architecture of the baseline ONED system, where video streams can come from one or more multi-lingual video channels. These streams are then partitioned into shots. Each shot is about several (e.g., three) seconds long and defined as a single continuous camera operation without an editor's cut, fade, or dissolve. For each shot, the feature extraction module both extracts image features from its keyframe, and obtains the English text features by using automatic speech recognition followed by machine translation, so that the original video clips in different languages become comparable. Then the ONED component uses the text and image features to identify the *new-event shots* that present previously unseen events, and sends these shots to a consumer, who can be either a person or a computer program that does deeper analysis. (Note that, although we use video shots as the basic NED unit in this work, our following analysis is not relying on this choice and thus they are universally applicable to other units such as news story and so on.)

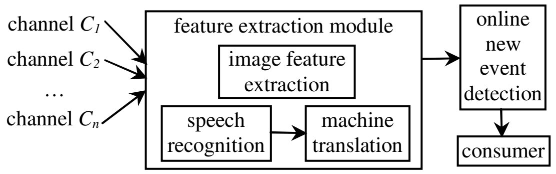


Figure 2. The baseline online new event detection system.

2.2 Image and Text Features

The baseline system uses the traditional tf·idf term weights as the text features [23]. Since each shot *S* is too short to contain enough text for computing meaningful text similarity (see Section 2.3), we extend the text of *S* with both the texts of the previous *m=5* shots and the texts of the next *m* shots [11]. (All these shots come from the same channel.) Following the convention of information retrieval [23], we define a *term* as a unique word and *vocabulary* as the set of all the unique words. For each term *t* in the vocabulary and a shot *S* in a shot set *E*, the baseline system uses the following formulas to compute the term weight:

(f1) term frequency (tf) weight $w_{tf} = \ln(tf + 1)$,

(f2) inverse document frequency (idf) weight

$$w_{idf} = \ln[(N+1)/(df + 0.5)],$$

(f3) term (tf·idf) weight $w_t = w_{tf} \times w_{idf}$.

where *tf* is term *t*'s frequency (i.e., number of occurrences) in the text of *S*, *N* is the total number of shots in *E*, and *df* is the number of shots in *E* whose texts contain *t*.

In practice, there are many different ways to extract image features that are (almost equally) suitable for detecting near-duplicate images. Our current baseline system uses the color moment feature described in Campbell et al. [6], where the localized color statistics are extracted from a 3×3 grid of the keyframe image, and the first three moments for each grid in Lab color space are used to construct the *n=81* image features f_i $(1 \le i \le n)$ of *S* [6].

The IBM TALES (Translingual Automatic Language Exploitation) system [12, 20] can use computer clusters to perform both image and text feature extraction on video streams from thousands of channels simultaneously with a delay of about four minutes − almost in real time. Therefore, in the rest of this paper, we focus on the ONED components that existing systems cannot complete in real time.

2.3 Dissimilarity Value Computation

To detect new-event shots in a video ONED system, we need to compute the dissimilarity between two shots S_1 and S_2 using their text and image features. The smaller the dissimilarity is, the more likely S_1 and S_2 are to present the same event. We show the dissimilarity computation method as follows. First, the text dissimilarity value is obtained using (f4) and (f5):

(f4) normalized text dot product value

$$text\_dotprod_{S_1, S_2} = \sum_{t \in S_1, S_2} w_{t,1} \times w_{t,2} \bigg/ \sqrt{\sum_{t \in S_1} w_{t,1}^2 \times \sum_{t \in S_2} w_{t,2}^2},$$

(f5) text dissimilarity value

$$text\_dissim_{S_1, S_2} = 1 - text\_dotprod_{S_1, S_2},$$

where $w_{t,j}$ (*j*=1, 2) is the term weight for S_j. Notation $t \in S_j$ means that term *t* appears in the text of S_j. As mentioned in Braun and Kaneshiro [5], formulas (f1), (f2), (f3), (f4), and (f5) achieved the best detection accuracy in the latest TDT5 competition [25] for ONED on document streams. Next, we obtain the image dissimilarity value using (f6) and (f7) [11]:

(f6) normalized image dissimilarity value

$$image\_dissim_{S_1,S_2} = \sqrt{\sum_{i=1}^{n}(f_{i,1}-f_{i,2})^2 \Big/ n},$$

(f7) binarized image dissimilarity value

$$bin\_image\_dissim_{S_1,S_2} = I_{\{image\_dissim_{S_1,S_2} > T_{image}\}},$$

where $f_{i,j}$ (j=1, 2) is the image feature for S_j, T_{image} is a threshold for binarizing the image dissimilarity, and I is the indicator function. That is, the binarized image dissimilarity is 1 if the normalized image dissimilarity is larger than T_{image}, otherwise it is 0. Finally, the overall dissimilarity value of S_1 and S_2 is obtained as a linear combination of the text dissimilarity value and the binarized image dissimilarity value according to (f8):

(f8) $$overall\_dissim_{S_1,S_2} = text\_dissim_{S_1,S_2} + w_{image} \times bin\_image\_dissim_{S_1,S_2},$$

where w_{image} is the linear weight for the visual modality. As mentioned in Hsu and Chang [11], such a linear fusion model is one of the most effective approaches to fuse visual and text modalities in video ONED systems.

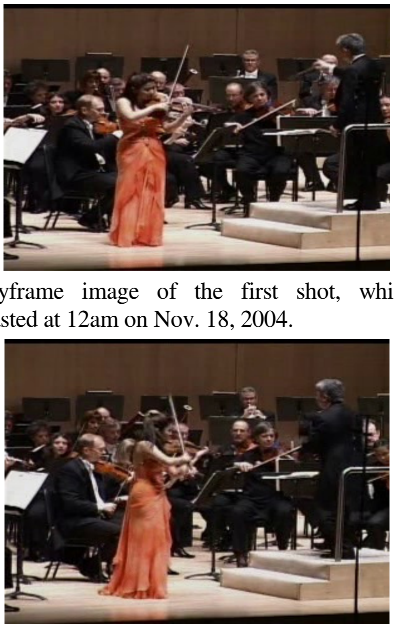
2.4 Detailed Processing Steps

In this section, we present the details of the baseline system. We follow the typical pre-processing operations in information retrieval for the text of each shot, i.e., (1) stemming is performed using the standard Porter stemmer [21], and (2) stopwords are removed by using the standard SMART stopword list [24]. Note that, the shot set E keeps changing as new shots continue to arrive in a video streaming environment. As mentioned in Braun and Kaneshiro [5], for ONED purpose, the computation of the tf and idf weights can be based on a static shot set E' that has characteristics similar to E. For a term that does not exist in the text of E', its *df* is assumed as one. Compared to the method that incrementally updates the statistics N and *df*, this static method has a much lower overhead, while the detection accuracy remains roughly the same [5].

When a shot S arrives, S is first pre-processed and its features are saved in memory. Then S is compared with all the old shots that arrived in the past except for the $L=50$ shots that just arrived from the same channel before S, as those L shots are likely to be in the same news story segment as S. If all the overall dissimilarity values between S and the old shots are above a threshold T, S is predicted to be a new-event shot. Otherwise if the overall dissimilarity value between S and an old shot S_{old} is below T, S is predicted to present the same event as S_{old}.

2.5 Advantages of Using both Text and Image Features

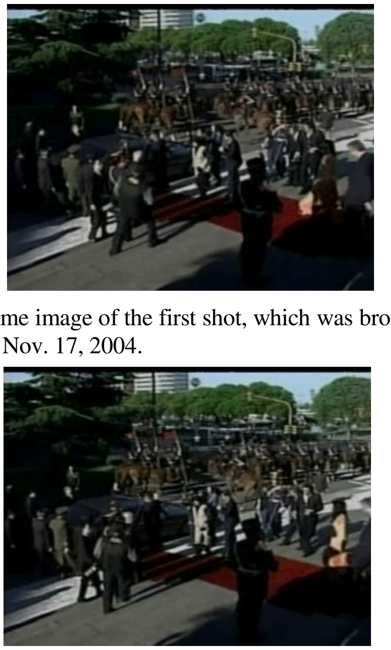
The experiments presented by Hsu and Chang [11] have shown that although text features are the most effective component in detecting new events, visual near-duplicates can still consistently enhance the detection accuracy of the text baseline. To be more specific, using both text and image features can improve the detection accuracy of the text baseline by up to 25%. This can be explained by the fact that similar images in two shots often provide evidence that they present the same event, even if their associated speech transcript may not be sufficiently similar due to paraphrasing or speech recognition/translation errors [10].



(a) Keyframe image of the first shot, which was broadcasted at 12am on Nov. 18, 2004.

(b) Keyframe image of the second shot, which was broadcasted at 7pm on Nov. 18, 2004.

Figure 3. Keyframe images of the two shots in the first example, which is about a Korean violinist's performance in Toronto.



(a) Keyframe image of the first shot, which was broadcasted at 7pm on Nov. 17, 2004.

(b) Keyframe image of the second shot, which was broadcasted at 12am on Nov. 19, 2004.

Figure 4. Keyframe images of the two shots in the second example, which is about Chinese president Jintao Hu's visit of Argentina.

To illustrate this issue, we provide two examples in Figure 3 and Figure 4. Both examples come from the TRECVID 2005 video collection [26], each of which contains two shots that present the same event but were broadcasted at different times. The first example is about a Korean violinist's performance in Toronto. The second example is about Chinese president Jintao Hu's visit to Argentina. In both examples, when we use the default parameter settings in the baseline system, text features by themselves cannot correctly detect that these two shots are presenting the same event. (The threshold for the normalized text dot product value is $1+w_{image}-T$ when only text features are used, as described in Section 3.) However, by considering the additional evidence from image features, the system can produce the correct predictions.

3. ANALYSIS OF THE OVERALL DISSIMILARITY FORMULA

To provide more insight on the overall dissimilarity value, we rewrite the original dissimilarity formula (f8) into an equivalent form that treats text and image features asymmetrically. We further analyze this alternative form to show how the NED process can be more efficient. To begin, we substitute the formulas (f5) and (f7) into (f8) and rewrite the overall dissimilarity of S_1 and S_2 to be

(f9) $$overall\_dissim_{S_1,S_2} = 1 - text\_dotprod_{S_1,S_2} + w_{image} \times I_{\{image\_dissim_{S_1,S_2} > T_{image}\}}.$$

We analyze (f9) by considering two possible cases, while either case has two sub-cases:

(1) When the keyframes of S_1 and S_2 are near-duplicate images, i.e., $image\_dissim_{S_1,S_2} \le T_{image}$, we have $overall\_dissim_{S_1,S_2} = 1 - text\_dotprod_{S_1,S_2}$. Thus, we can predict that

  (i) Sub-case 1: S_1 and S_2 present the same event if $1-T < text\_dotprod_{S_1,S_2}$, and

  (ii) Sub-case 2: S_1 and S_2 present different events if $1-T \ge text\_dotprod_{S_1,S_2}$.

(2) When the keyframes of S_1 and S_2 are not near-duplicate images, i.e., $image\_dissim_{S_1,S_2} > T_{image}$, we have $overall\_dissim_{S_1,S_2} = 1 - text\_dotprod_{S_1,S_2} + w_{image}$. Thus, we can predict that

  (i) Sub-case 3: S_1 and S_2 present the same event if $1+w_{image}-T < text\_dotprod_{S_1,S_2}$, and

  (ii) Sub-case 4: S_1 and S_2 present different events if $1+w_{image}-T \ge text\_dotprod_{S_1,S_2}$.

Figure 5 illustrates the above four sub-cases.

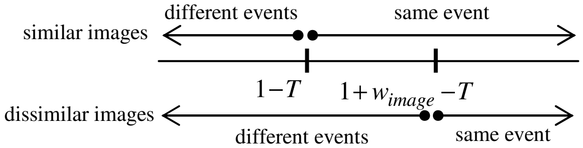


Figure 5. Graphical illustration of the above four sub-cases.

For any two shots S_1 and S_2, it seems that we must use both their text/image features and check all of the above four sub-cases to determine whether they present the same event. However, this turns to be overkill in many cases. By treating text and image asymmetrically, we can greatly simplify the NED operation by rewriting the above four sub-cases into the following equivalent three cases (see Figure 5), among which only Case 2 has two sub-cases:

(1) **Case 1**: $1-T \ge text\_dotprod_{S_1,S_2}$. In this case, we predict that S_1 and S_2 present different events, irrespective of the normalized image dissimilarity $image\_dissim_{S_1,S_2}$.

(2) **Case 2**: $1-T < text\_dotprod_{S_1,S_2} \le 1+w_{image}-T$. In this case, there are two sub-cases:

  (i) **Sub-case 1**: If $image\_dissim_{S_1,S_2} \le T_{image}$, we predict that S_1 and S_2 present the same event.

  (ii) **Sub-case 2**: If $image\_dissim_{S_1,S_2} > T_{image}$, we predict that S_1 and S_2 present different events.

(3) **Case 3**: $1+w_{image}-T < text\_dotprod_{S_1,S_2}$. In this case, we predict that S_1 and S_2 present the same event, irrespective of the normalized image dissimilarity $image\_dissim_{S_1,S_2}$.

In the above cases, both Case 1 and Case 3 only require the text features of shots S_1 and S_2. Hence, for ONED purpose, text features and image features can be treated asymmetrically, i.e., we can use text features as a pre-filter to filter out most of the unnecessary operations on image features. This can bring a huge benefit to the detection efficiency, because the text similarities of most shot pairs are low [18], and hence Case 1 is the most frequently occurring case. On the other hand, it is undesirable to process image features before text features because using image features alone cannot determine whether S_1 and S_2 present the same event [11].

4. TECHNIQUES FOR IMPROVING EFFICIENCY

In this section, we describe our techniques for improving the efficiency of the ONED system based on the analysis of Section 3. We first give a high-level overview of our optimized ONED system, and then elaborate on the individual techniques.

4.1 Architecture of the Optimized System

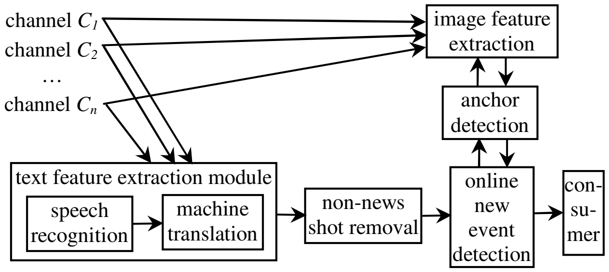


Figure 6. Our optimized online new event detection system.

Figure 6 shows the architecture of our optimized ONED system. Video streams from one or more channels are divided into shots. For each shot S, the text features are extracted by using speech recognition as well as machine translation techniques. The text features are used to identify and remove the non-news shots. The remaining news shots are fed to the ONED component, where new-event shots are identified and sent to the consumer. During the ONED process, we extract the image features of S only when it is necessary to determine whether the keyframe of *S* is an anchor image and to compute the image similarities between *S* and the old shots. Figure 7 shows the high-level description of the algorithm used in the ONED component. The details of this algorithm are explained in Sections 4.3-4.6.

```
Image_feature_extraction_flag=FALSE; /* whether image
        features have been extracted for the new shot S */
First_story_shot_flag=TRUE; // whether S is a new-event
        shot
Use the pre-filtering method described in Section 4.6 to
identify the old shots that need to be compared with S. For
each such old shot $S_{old}$ {
  Compute $text\_dotprod_{S,\,S_{old}}$;
  If ($1-T \geq text\_dotprod_{S,\,S_{old}}$) /* Case 1*/
    Predict that S and $S_{old}$ present different events;
  Else if ($1+w_{image}-T < text\_dotprod_{S,\,S_{old}}$) { // Case 3
    Predict that S and $S_{old}$ present the same event;
    First_story_shot_flag=FALSE;
    Exit the for loop;
  }
  Else if ($1-T < text\_dotprod_{S,\,S_{old}} \leq 1+w_{image}-T$) { // Case 2
    If (!Image_feature_extraction_flag) {
      Extract image features for S;
      Determine whether S is an anchor shot;
      Image_feature_extraction_flag=TRUE;
    }
    If ((S is an anchor shot) || ($S_{old}$ is an anchor shot))
      $image\_dissim_{S,\,S_{old}} = T_{image}+1$; /* treat the keyframes
            of S and $S_{old}$ to be dissimilar */
    Else compute $image\_dissim_{S,\,S_{old}}$;
    If ($image\_dissim_{S,\,S_{old}} \leq T_{image}$) { /* Sub-case 1 */
      Predict that S and $S_{old}$ present the same event;
      First_story_shot_flag=FALSE;
      Exit the for loop;
    }
    Else if ($image\_dissim_{S,\,S_{old}} > T_{image}$) /* Sub-case 2 */
      Predict that S and $S_{old}$ present different events;
  } /* end of Case 2 */
} /* end of the for loop */
If (First_story_shot_flag) {
  If (!Image_feature_extraction_flag) {
    Extract image features for S;
    Determine whether S is an anchor shot;
  }
  Save S's information in memory;
  Send S to the consumer of the ONED system;
}
```

Figure 7. High-level description of the algorithm used in the online new event detection component.

4.2 Detecting and Removing Non-News Shots

In broadcast videos, non-news video segments (e.g., commercials, TV shows) are usually mixed with news stories. For ONED purpose, non-news shots should not be treated as new-event shots, even if no similar shots have appeared before. Removing these shots can not only reduce the number of shots that need to be processed by the ONED component, but also improve the efficiency and the detection accuracy of the ONED system.

To this end, a simple method is to manually specify the regular time periods when news videos are broadcasted. However, such a method is not scalable to tens of thousands of channels, as is the typical case that an ONED system needs to handle [8, 17]. Moreover, our purpose here is to remove all the non-news shots rather than commercials only [1, 9, 10]. As an alternative, we apply a simple text-based method to remove the non-news shots. Its basic idea is that non-news shots (e.g., commercials) often have larger background noise than news shots, which makes it difficult for the speech recognizer to recognize the text in the non-news video. Also, in news shots the anchor person tends to talk at a faster pace than non-news shots (e.g., TV shows). Based on these two properties, we predict that a shot *S* is not news if the recognized text of *S* contains fewer than *J* distinct terms where *J* is a predetermined constant. Our experiments in Section 5.1 show that a reasonable choice for *J* is usually between 50 and 80. Although this method is rather simple, it is highly accurate and has a low overhead that helps to improve the efficiency of the ONED system. Also, the expensive image feature extraction step is no longer needed for the dropped non-news shots.

4.3 Delaying the Processing of Image Features

As mentioned in Section 3, it is desirable to delay the processing of image features as much as possible. As shown in Figure 6 and Figure 7, when processing a new shot *S*, we first extract its text features but not its image features. When comparing *S* with an old shot S_{old}, we first compute their normalized text dot product instead of their image dissimilarity. If $1-T \geq text\_dotprod_{S,\,S_{old}}$ (Case 1 of Section 3), we predict that *S* and S_{old} present different events. If $1+w_{image}-T < text\_dotprod_{S,\,S_{old}}$ (Case 3), we predict that *S* and S_{old} present the same event. In both Case 1 and Case 3, we skip the costly but unnecessary image dissimilarity computation step. Only in Case 2 (when $1-T < text\_dotprod_{S,\,S_{old}} \leq 1+w_{image}-T$), we need to compute the image dissimilarity. Since the text dot products of most pairs of shots are low [18], Case 2 usually occurs much less frequently than Case 1 and Case 3. Consequently, most image dissimilarity computations can be saved.

Moreover, when we make the prediction that a new shot is not a new event, if all the compared old shots belong to either Case 1 or Case 3, we can skip the expensive image feature extraction step. In other words, we only need to extract image features for a new shot *S* when either we predict that *S* is a new-event shot or we have $1-T < text\_dotprod_{S,\,S_{old}} \leq 1+w_{image}-T$ for some S_{old}. In practice, in the presence of a large number of channels, most shots will be presenting existing events due to the repeated mention of the same event both across different channels and within the same channel [25, 18]. Also, Case 1 and Case 3 occur much more frequently than Case 2. Thus, we can skip the expensive image feature extraction step for a large fraction of the shots.

4.4 Detecting Anchor Images

In news videos, news stories are typically broadcasted by anchor persons. Figure 8 shows an image example of an anchor person from the CNN news. Two news shots from the same channel often have keyframes with the same anchor person, but present different events. However, in this case, the similar keyframes should not be treated as a hint that these two shots present the same event. To take this factor into account, we use the method described in Campbell et al. [6] to detect which keyframes are anchor images based on Support Vector Machines and low-level color correlogram features. When comparing two shots, we set the binarized image dissimilarity to be 1 if the keyframe of either shot is an anchor image. That is to say, we treat their keyframes to be dissimilar if either of them is an anchor shot. This can reduce the effect of the false evidence of anchor shots on the detection accuracy of the ONED system.

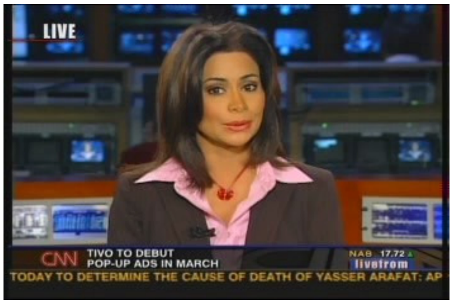


Figure 8. A keyframe image example of an anchor person.

4.5 Reducing the Amount of Saved Information

Typically, the discussion of an event only lasts for a finite amount of time in news videos, and a new shot is unlikely to present the same event as a shot that is fairly old. Hence, we only keep in memory the information of those old shots that are within a sliding window of the last *W* days. Here *W* is a predetermined constant. The information kept for a shot *S* includes both its text features and its image features (see Section 4.6 for details) but not its video images, as only these features are needed for comparing *S* with future shots. Once an old shot expires from the sliding window, its information is thrown away immediately.

Typically, an event is presented by a large number of shots. Only one of these shots is the new-event shot. All the shots that present the same event tend to be similar to each other. Therefore, it is overkill to compare a new shot with all the old shots that present the same event. Instead, we only keep the information of the new-event shots. When a new shot *S* arrives, *S* is compared with the old new-event shots. If *S* is predicted to be a new-event shot that presents a new event, *S*'s information is saved in memory. Otherwise *S* is discarded.

All the terms in the text of a shot can be sorted in descending order of their term weights. In general, those terms with larger weights are more important for NED. Hence, for each saved shot, we keep only the top-*K* terms with the largest weights rather than all the terms. Here *K* is a predetermined constant. Only the top-*K* terms are used to compute the text dot product.

4.6 Pre-filtering

To reduce the overhead of computing dissimilarity values, a pre-filtering technique is developed by using a low-overhead method to quickly filter out most of the shots that present different events from the new shot. In this way, we can substantially reduce the number of dissimilarity values that need to be computed. Consider two shots S_1 and S_2. If S_1 and S_2 present the same event, the top terms of their texts tend to have some overlap. That is, some term(s) is likely to appear in the top terms of both S_1's text and S_2's text. Thus, these top terms can be used to quickly filter out unnecessary computations. More specifically, we have a predetermined constant M ($M \leq K$). Before computing the text dot product of S_1 and S_2, we first check whether the top-*M* terms of S_1 and S_2 intersect. If so, we continue to compute the text dot product of S_1 and S_2. Otherwise, we predict that S_1 and S_2 present different events and do not compute their text dot product.

We build indices to avoid unnecessary processing of the shots that have been pre-filtered out. Each term in the vocabulary has a term id. Each shot has a shot id corresponding to its arrival time. Two indices are kept for all the saved shots: a forward index and an inverted index. The forward index has an entry for each saved shot. These entries are sorted in descending order of shots' arrival time. This allows us to quickly identify and drop the information of those shots that have expired from the sliding window of the last *W* days (see Section 4.5). For each saved shot, the corresponding entry keeps both the image features and the top-*K* terms associated with their term weights. These terms are sorted in ascending order of their term ids. Consequently, the text dot product of two shots can be computed through an efficient "merge" of their term lists.

For each saved shot, only its top-*M* terms are tracked by the inverted index. The inverted index has an entry for each term in the vocabulary. The entry for term *t* is a posting (linked) list of the shot ids of all the shots whose top-*M* terms contain *t*. These shot ids are sorted in descending order so that merging posting lists can be done efficiently. When a new shot *S* arrives, we only scan the *M* posting lists that correspond to *S*'s top-*M* terms. These *M* posting lists are merged together to find the shot ids of the candidate shots that may present the same event as *S*. This is the pre-filtering technique described above. Then for each such candidate shot S_c, the forward index is used to compute the text dot product and the image dissimilarity (if needed) of *S* and S_c. This computation is performed at the same time that candidate shot ids are generated. In this way, if the overall dissimilarity value of *S* and an old shot is smaller than the threshold *T*, *S* is predicted to be a non-new-event shot and the processing for *S* stops immediately. Otherwise if *S* is predicted to be a new-event shot, *S*'s information can be easily added into the inverted index, as *S*'s shot id is larger than the shot ids of the saved shots.

5. PERFORMANCE EVALUATION

We implemented a prototype of our optimized ONED system on top of IBM's System S [13]. Our implementation uses two processing elements (PEs) that consume and produce streams of data through input and output ports, respectively. One PE produces the video stream and sends it to another PE that implements ONED.

To evaluate the performance of the proposed system, we use the largest available video retrieval benchmark, TRECVID 2005 [26]. This benchmark includes 171 hours of videos from six channels in three languages (Arabic, English, and Chinese). The time span is from October 30 to December 1, 2004. Our measurements were performed on two computers, each with one 1.6GHz processor, 1GB main memory, one 75GB disk, and running Linux.

The default parameters in our ONED system are as follows: $T=0.9$ (the threshold for the overall dissimilarity value), $w_{image}=0.1$ (the weight for the visual modality), $T_{image}=0.2$ (the threshold value for image dissimilarity), $J=70$ (the threshold of the number of distinct terms for determining whether a shot is news), $W=29$ (the sliding window size in days), $K=250$ (the number of top terms kept in each saved shot), and $M=10$ (the number of top terms used for pre-filtering purpose).

For ONED on text document streams, Luo et al. [18] has shown that those techniques proposed in Sections 4.5 and 4.6 can significantly improve the processing rate of the ONED system without sacrificing much detection accuracy. In our experiments on the TRECVID 2005 collection, we also found that the output results of the ONED system differ by only 5% when those techniques are used and when they are not used. This also confirms that those techniques do not have much impact on the detection accuracy of the ONED system. Moreover, Luo et al. [18] showed that the default values of the parameters *W*, *K*, and *M* are reasonable in detecting new events for news streams. Therefore, in this section, we focus on evaluating the techniques described in Sections 4.2~4.4 by varying the values of the parameters *T*, W_{image}, T_{image}, and *J*. Among these four parameters, *T*, W_{image}, and T_{image} are also needed in the baseline system.

At present, there is no publicly available benchmark for video ONED with official annotation. It is difficult to label the ground truth for ONED on the TRECVID 2005 video collection by ourselves, as the labeling procedure used by the organizations responsible for creating benchmarks typically involves two steps [27]: (1) running multiple *different* systems on the video set to generate the candidate results; (2) hiring professional analysts to make the judgment. Nevertheless, this issue should not become a problem in the evaluation because we mainly focus on improving the detection efficiency. Moreover, our technique of delaying the processing of image features (Section 4.3) does not affect the detection accuracy of the ONED system. It is also easy to see that our techniques of removing non-news shots (Section 4.2) and handling anchor images (Section 4.4) can improve the detection accuracy of the ONED system, as it is widely known that similar techniques can improve the accuracy of video retrieval [6]. Therefore, in our experiments, we focus on the processing speed of the ONED system rather than on the detection accuracy.

5.1 Justification for Some of the Default Parameters

In this section, we provide some justification for the default parameters $T=0.9$, $w_{image}=0.1$, and $T_{image}=0.2$. Our experiments in Section 5.2 show that the effectiveness of our techniques is insensitive to these exact values within a fairly large range.

We first consider the default parameter of $T_{image}=0.2$. We manually verify that by choosing this parameter and using the $n=81$ color moments described in Section 2.2 as image features, the following two conditions are satisfied. First, for any two shots S_1 and S_2, if their keyframe images are judged to be similar according to formula (f7) in Section 2.3, these images are indeed similar and provide clue that S_1 and S_2 present the same event, as shown in Figure 4 and Figure 5. This ensures that T_{image} is not too large. Second, we have a sufficient number of reasonably similar keyframe image pairs among all the shots. Note that if T_{image} is too small, we are basically only allowing identical images to be judged as similar images and hence the image features have almost no effect in the overall dissimilarity value computation formula (f8), which is undesirable [11]. In our experiments, we find that within the range [0.15, 0.25], the exact value of T_{image} does not have much impact on the performance of the ONED system.

Next, we discuss the other two default parameters $T=0.9$ and $w_{image}=0.1$. Consider a special case where the keyframe images of the shots are all dissimilar to each other. In this situation, we are going back to the traditional case of ONED on document streams [18], where image features are unavailable and we can only use text features. Then according to formula (f9) in Section 3, for any two shots, $1-T+w_{image}$ is the threshold value for using their normalized text dot product value to determine whether they present the same event. As has been shown in Braun and Kaneshiro [5] for ONED on document streams, the optimal threshold value for the text dot product value is around 0.2. Thus, we should have $1-T+w_{image}=0.2$ in our video ONED system. If we let the image contribution w_{image} be half of that threshold value, we have $1-T=w_{image}=0.1$. That is, $T=0.9$ and $w_{image}=0.1$.

5.2 Results and Sensitivity Analysis

In this section, we report the efficiency of the proposed algorithm and carry out a series of sensitivity analysis to evaluate the impact of parameters on the performance of the ONED system. The TRECVID 2005 collection was originally used as the benchmark for video retrieval rather than ONED. We found that its size is too small to evaluate ONED systems appropriately and results in certain undesirable effects. So we make the following adjustments to compensate for these undesirable effects.

First, when we measure the system throughput and the total video set processing time, we do not include the time spent on extracting text and image features. Instead, we measure the feature extraction time separately. This is because the time spent on extracting text and image features is proportional to the number of shots processed, while the time spent on the ONED component is quadratic with respect to the number of shots processed, which is much more expensive in a typical ONED situation where tens of thousands of channels are handled simultaneously [8, 17].

Second, due to the small size of the TRECVID 2005 video set, few shots there have the chance of getting repeated. Consequently, most news shots there (a rough estimate is about 65%) are predicted as new-event shots. However, in a large scale ONED environment, we would expect most news shots to be non-new-event ones. For example, TDT5 [25] is the standard benchmark for ONED on text document streams. It has one hundred times more (voice) text than the TRECVID 2005 video set, and a rough estimate is that about 85% of all the documents in that benchmark are non-new-event documents [18]. To compensate this effect, we only consider non-new-event shots when measuring the percentage of news shots whose image feature extraction steps are saved. If this percentage is large, we would expect the corresponding percentage for all the news shots (i.e., both new-event shots and non-new-event shots) to be large in a typical video ONED scenario.

Our main results are as follows. Using the proposed techniques for improving efficiency, it takes 15 seconds to process all the shots in the TRECVID 2005 video set. In contrast, the baseline system described in Section 2 uses 7,203 seconds (two hours) to process the same shots. Compared to the baseline, our techniques improve the efficiency by two orders of magnitude (480 times). In the following experiments, we varied the value of each parameter while keeping the other parameters fixed in order to analyze the sensitivity of their value settings.

Table 1. Speed up ratio gained by our techniques.

processing time of baseline system	7203 seconds
processing time of optimized system	15 seconds
speed up ratio	480 times

***J* (the threshold of the number of distinct terms for determining whether a shot is news)**

The first experiment concerns *J*, the threshold of the number of distinct terms for determining whether a shot is news. We varied *J* from 40 to 100. To test the identification accuracy of our method described in Section 4.2, we randomly selected one hour's video sequence in the entire TRECVID 2005 video set and manually labeled the non-news shots and the news shots. (We also tested a few other video sequences in the TRECVID 2005 video set and the results are similar.) Figures 9 and 10 show the impact of *J* on both the precision and recall of identifying non-news shots and news shots. When *J* is too large or too small, either the precision or the recall drops significantly. A good value for *J* is between 50 and 80, where we can obtain both good precision and good recall.

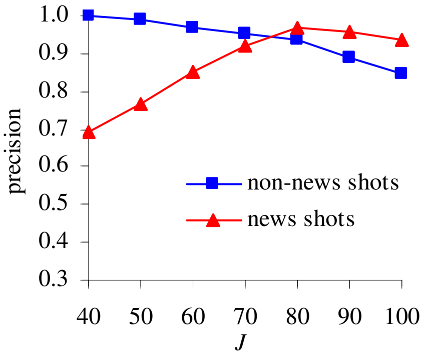


Figure 9. Precision vs. *J*.

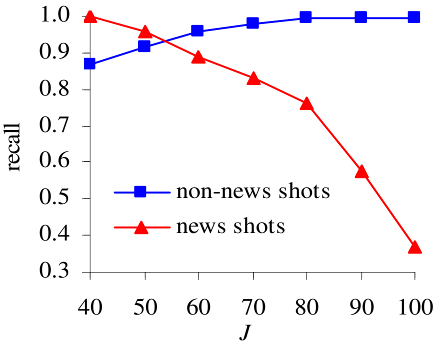


Figure 10. Recall vs. *J*.

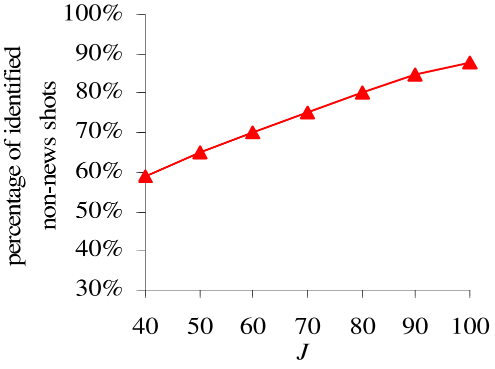


Figure 11. Percentage of identified non-news shots vs. *J*.

Figure 11 shows the impact of *J* on the percentage of identified non-news shots in the entire TRECVID 2005 video set. Figure 12 shows the impact of *J* on the throughput of our ONED system. The larger the *J*, the more shots are identified and dropped as non-news shots and hence the higher the processing rate of the ONED system. In the default case that *J=70*, 75% of all the shots are identified and dropped as non-news shots.

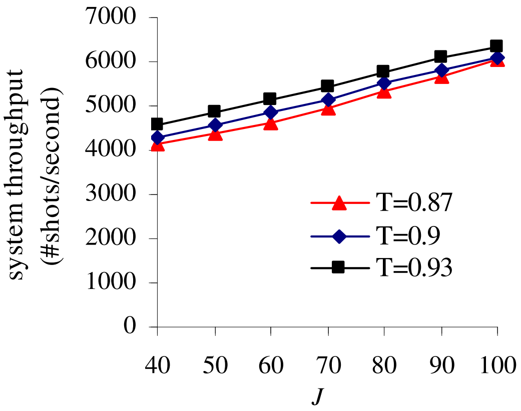


Figure 12. System throughput vs. *J*.

***T* (threshold for the overall dissimilarity value)**

The second experiment concerns *T*, the threshold for the overall dissimilarity value. We varied *T* from 0.85 to 0.95. As mentioned in Section 4.3, delaying the processing of image features can save many unnecessary image dissimilarity computations. Figure 13 shows the impact of *T* on the percentage of saved image dissimilarity computations. Most shot pairs present different events and have small normalized text dot product values. That is, most shot comparisons fall into Case 1 of Section 3, where $1-T \geq text\_dotprod_{S,S_{old}}$. The larger the *T*, the fewer shot pairs fall into Case 1. Thus, the percentage of saved image dissimilarity computations decreases as *T* increases. In the default case that *T=0.9* and *J=70*, 77% of all the image dissimilarity computations are saved.

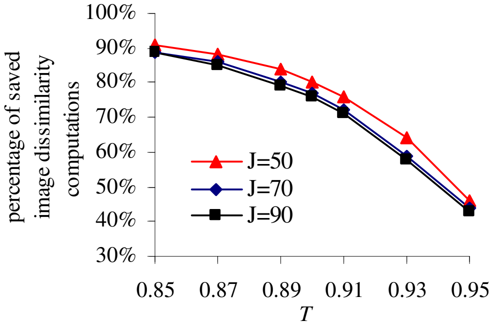


Figure 13. Percentage of saved image dissimilarity computations vs. *T*.

w_{image} (weight for the visual modality)

The third experiment concerns w_{image}, the weight for the visual modality. We varied w_{image} from 0.05 to 0.15. As mentioned in Section 4.3, delaying the processing of image features can waive the expensive image feature extraction step for a large number of shots.

Figure 14 shows the impact of w_{image} on the percentage of non-new-event news shots whose image feature extraction steps are saved. The larger the w_{image}, the fewer shot pairs fall into Case 3 of Section 3 ($1 + w_{image} - T < text\_dotprod_{S,\,S_{old}}$), and the fewer image feature extraction steps can be saved. Consequently, the percentage of non-new-event news shots whose image feature extraction steps are saved decreases as w_{image} increases. In the default case that w_{image}=*0.1* and *J=70*, 45% of all the image feature extraction steps are saved for the non-new-event news shots.

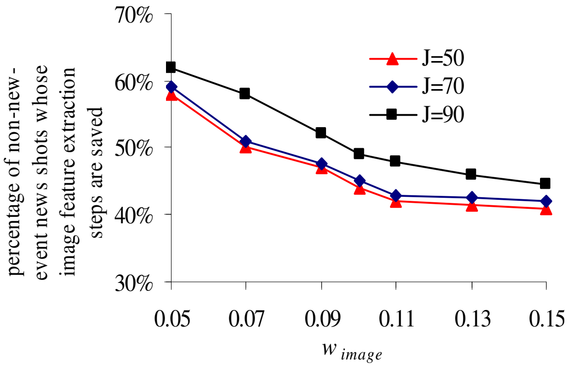


Figure 14. Percentage of non-new-event news shots whose image feature extraction steps are saved vs. w_{image}.

On our computer, for each 3-second shot, it takes about 2 seconds to extract the text features by performing speech recognition followed by machine translation, and it takes another 2 seconds to extract the image features. Hence, in the case that 75% of all the shots are dropped as non-news shots, 85% of all the news shots are identified as non-new-event shots, and 45% of the image feature extraction steps are skipped for these non-new-event shots, we can save about 85% (75%+25%×85%×45%) of all the image feature extraction steps, or equivalently 40% of the total overhead on text and image feature extraction for all the shots.

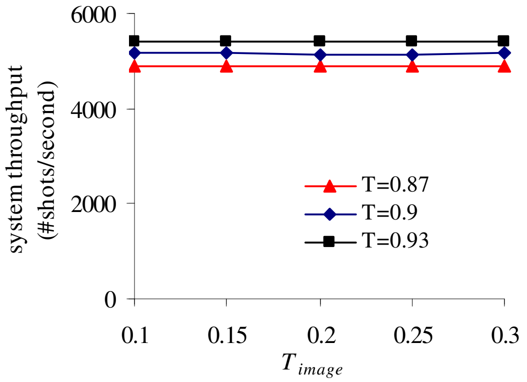


Figure 15. System throughput vs. T_{image}.

T_{image} (threshold value for image dissimilarity)

The fourth experiment concerns T_{image}, the threshold value for image dissimilarity. We varied T_{image} from 0.1 to 0.3. Figure 15 shows the impact of T_{image} on the throughput of our ONED system. For each keyframe image of a shot, very few (if any) keyframe images of the other shots are similar to it. As long as T_{image} is within a reasonable range, these images can pass the image similarity filtering condition $image\_dissim_{S_1,S_2} \le T_{image}$ almost irrespective of the concrete value of T_{image}. Consequently, T_{image} has almost no effect on the throughput of our ONED system.

6. CONCLUSION

This paper proposes several techniques for improving the efficiency of online new event detection on video streams so that video ONED becomes real-time. We implemented a prototype of our framework on top of a stream processing middleware. Our experiments with the standard TRECVID 2005 benchmark show that the proposed techniques can improve the video processing rate by two orders of magnitude without sacrificing much detection accuracy (less than 5%). Also, the effectiveness of our techniques is insensitive to the choice of the exact parameter values.

7. ACKNOWLEDGEMENTS

We thank Jerome L. Quinn for answering our questions on the IBM TALES system.

8. REFERENCES

[1] L. Agnihotri, N. Dimitrova, and T. McGee et al. Envolvable Visual Commercial Detector. CVPR (2) 2003: 79-84.

[2] J. Allan, V. Lavrenko, and H. Jin. First Story Detection in TDT is Hard. CIKM 2000: 374-381.

[3] J. Allan, R. Papka, and V. Lavrenko. On-Line New Event Detection and Tracking. SIGIR 1998: 37-45.

[4] T. Brants, F. Chen. A System for New Event Detection. SIGIR 2003: 330-337.

[5] R. Braun, R. Kaneshiro. Exploiting Topic Pragmatics for New Event Detection in TDT-2004. TDT-2004 Workshop.

[6] M. Campbell, S. Ebadollahi, and D. Joshi et al. IBM Research TRECVID-2006 Video Retrieval System. NIST TRECVID workshop, 2006.

[7] F. Chen, A. Farahat, and T. Brants. Story Link Detection and New Event Detection are Asymmetric. HLT-NAACL 2003.

[8] M. Clayton. US Plans Massive Data Sweep. The Christian Science Monitor, February 09, 2006. http://www.csmonitor.com/2006/0209/p01s02-uspo.html, 2006.

[9] P. Duygulu, M. Chen, and A.G. Hauptmann. Comparison and Combination of Two Novel Commercial Detection Methods. ICME 2004: 1267-1270.

[10] P. Duygulu, J. Pan, and D.A. Forsyth. Towards Auto-documentary: Tracking the Evolution of News Stories. ACM Multimedia 2004: 820-827.

[11] W. Hsu, S. Chang. Topic Tracking across Broadcast News Videos with Visual Duplicates and Semantic Concepts. ICIP 2006: 141-144.

[12] IBM Technology Translates Arabic Media Broadcasts to English. http://www.sda-asia.com/sda/news/psecom,id,11163,srn,4,channel,developer,nodeid,4,_language,Singapore.html#, 2006.

[13] K. Wu, P.S. Yu, and B. Gedik et al. Challenges and Experience in Prototyping a Multi-Modal Stream Analytic and Monitoring Application on System S. VLDB 2007: 1185-1196.

[14] G. Kumaran, J. Allan. Text Classification and Named Entities for New Event Detection. SIGIR 2004: 297-304.

[15] J.R. Kender, M.R. Naphade. Visual Concepts for News Story Tracking: Analyzing and Exploiting the NIST TRECVID Video Annotation Experiment. CVPR 2005: 1174-1181.

[16] X. Li, B.W. Croft. Novelty Detection Based on Sentence Level Patterns. CIKM 2005: 744-751.

[17] E. Lipton. Software to Monitor Overseas Opinions of U.S. The New York Times, October 4, 2006. http://news.zdnet.com/2100-9588_22-6122641.html, 2006.

[18] G. Luo, C. Tang, and P.S. Yu. Resource-Adaptive Real-Time New Event Detection. SIGMOD 2007: 497-508.

[19] Z. Li, B. Wang, and M. Li et al. A Probabilistic Model for Retrospective News Event Detection. SIGIR 2005: 106-113.

[20] R. Peterson. IBM Strives for Super Human Speech. http://www.accessible-devices.com/superspeech.html, 2006.

[21] M.F. Porter. An Algorithm for Suffix Stripping. Program 14(3): 130-137, 1980.

[22] N. Stokes, J. Carthy. Combining Semantic and Syntactic Document Classifiers to Improve First Story Detection. SIGIR 2001: 424-425.

[23] A. Singhal. Modern Information Retrieval: A Brief Overview. IEEE Data Eng. Bull. 24(4): 35-43, 2001.

[24] SMART Stopword List. http://www.lextek.com/manuals/onix/stopwords2.html, 2005.

[25] TDT Homepage. http://www.nist.gov/speech/tests/tdt.

[26] TREC Video Retrieval Evaluation. http://www-nlpir.nist.gov/projects/trecvid.

[27] E.M. Voorhees. Overview of TREC 2005. TREC 2005: 1-15.

[28] X. Wu, C. Ngo, and Q. Li. Threading and Autodocumenting News Videos: a Promising Solution to Rapidly Browse News Topics. IEEE Signal Processing Magazine 23(2): 59-68, 2006.

[29] Y. Yang, T. Pierce, and J.G. Carbonell. A Study of Retrospective and On-Line Event Detection. SIGIR 1998: 28-36.

[30] Y. Yang, J. Zhang, and J.G. Carbonell et al. Topic-conditioned Novelty Detection. KDD 2002: 688-693.

[31] Y. Zhai, M. Shah. Tracking News Stories across Different Sources. ACM Multimedia 2005: 2-10.

[32] YouTube Homepage. http://www.youtube.com.