Efficient and Language Independent News Story Segmentation for Telecast News Videos

Anubha Jindal*, Aditya Tiwari†, Hiranmay Ghosh‡
*TCS-Innovation Labs Delhi
249 D&E, TCS TOWERS, UDYOG VIHAR, PHASE-IV,
GURGAON, HARYANA-122016
Email: anubha.jindal@tcs.com
†Email: aditya.tiwari@tcs.com
‡Email: hiranmay.ghosh@tcs.com

Abstract—A TV news program comprises a continuous video stream containing a number of news stories, interspersed with commercials and headlines. This paper presents a method to detect the story boundaries and to separate out the stories from the other components and from each other. The method is based on movement of ticker text bands and repetition of ticker texts during different parts of a news program. The method does not use any language processing tool and is independent of language of telecast. It uses some simple features to distinguish news from the advertisements and can be used for large-scale news indexing. We produce some test results on channels telecasting in English and few other Indian languages.

Keywords—News analytics, Telecast video, Ticker Text

I. INTRODUCTION

Several news monitoring agencies need to monitor a large number of national and international news channels round the clock. The manual monitoring of a large number of channels not only requires a huge effort, but is also error-prone. In this context, we had earlier proposed a framework for automatic analysis and indexing of telecast news programs in multiple languages [10]. In this paper, we provide methods for removing advertisements and for isolating the individual stories from a continuous broadcast stream. Our method is independent of language of telecast and uses some simple features that can be detected in near real-time, so that the method can be applied in context of large-scale news indexing service.

Several methods for advertisement detection and story segmentation in news videos have been reported in the literature. Lienhart [8] proposes an algorithm for advertisement detection by combining feature based approach for pre-filtering and recognition based approach for finding the exact borders. However, this algorithm assumes some specific structure for the commercials, which is not followed worldwide. Hua [12] has presented a learning based advertisement detection approach using a set of six visual features and five audio features. Other features used for advertisement detection includes repetition of shots [9], [7] distinctive acoustic features of commercials [4], appearance of black video frames and silence in audio just before and after the advertisements [5], [8], absence of channel logos during commercials [1] and high-frequency scene changes [12], [8]. However, modern video editing techniques used in many news channel falsify these assumptions. Contemporary news presentations often use such techniques for attracting viewer attention and for creating sensation. Use of multiple features makes some of these methods more accurate but also more compute intensive. They cannot be used to process large number of telecast channels in near real-time.

Chua et al. [2] has proposed a two level multi modal framework for story segmentation, where the shots are classified using decision tree technique into some predefined categories, which is followed by story segmentation using Hidden Markov Model (HMM). This work was extended in [6], where heuristics has been used to classify a story into news and miscellaneous. This approach allowed the algorithm to scale up to process a large number of videos. Colace et al. [3] has used multi level probabilistic framework using HMM and Bayesian network for segmentation and classification. Use of multiple and complex feature set makes these algorithms slow. Speech or Close Captioned Text (CCT) has been used to distinguish the topic being discussed at different time intervals in [11]. However, CCT is not generally available with TV channels, except for a few countries, where it is mandated by law. Automatic Speech Recognition (ASR) technology is also not available for many languages and wherever available, cannot generally cope up with regional variation of accents. Thus, these algorithms cannot be universally applied on telecast news programs on any TV channel around the world.

We propose new methods for commercial detection and story segmentation in this paper, which are language independent and be performed in near real-time. Our algorithms are based on some news editing characteristics, which are found to be invariant over a large number of international, national and regional news channels. In our approach, the video corresponding to a news program is first processed to identify and remove the commercial segments, based on movement in ticker-text positions between news presentation and the advertisements. We have discovered a common
pattern for such movement over a large number of channels. Commercial removal from a news program results in several news segments. Each news segment comprises one or more stories and some news headlines, the latter being confined towards the beginning or the end of the segment. In the next step, the stories are separated from each other and from the headlines. Our story segmentation algorithm is based on repetition of ticker texts during the presentation of a story. Use of such simple features make the algorithms less compute intensive and can be applied in near real-time to a large number of news channels. Moreover, we do not use any language processing tool, such as Automatic Speech Recognition (ASR), closed caption text analysis or Optical Character Recognition (OCR), in our algorithms. In particular, the repetition of ticker texts are ascertained by image comparison methods. This makes our algorithm language independent and applicable to a large number of telecast channels in different languages from different regions of the world, where language processing tools may not be available.

The rest of the paper is organized as follows. Section II provides an overview of the TV news analytics system, which provides the context for the work presented in this paper. Section III describes the structure of a news program. Algorithms devised for advertisement detection and for story segmentation are explained in section IV and section V respectively. VI provides the details of experiments, results and performance. Section VII will contain the conclusion of the paper. Finally the last section is for the references, which are used in the paper.

II. SYSTEM OVERVIEW

Figure 1 provides an overview of our TV news analytics system. The scope of the system extends to several international, national and regional channels in different languages. While most of the channels monitored are dedicated news channels, some of the channels carry news with other programs. Since the focus of our system is to create a news archive, the recording in the system is restricted to news hours using electronic program guides.

A telecast stream is recorded in manageable chunks of video, typically of 10 minutes duration. Several consecutive video chunks recorded from the same channel constitute a news program. Once a video chunk has been recorded, the video frames are extracted for further processing, following which the shots are detected using a standard HSV Histogram method. While the frame-rate for the recorded videos are 24 fps, we use a lower frame-rate (2.4 fps) for efficient processing. At the next stage, we analyze the video-frames constituting a news program to detect and remove the advertisements. The advertisements are generally localized and their removal leads to a few news segments within a news program. In general, each news segment constitutes a few stories and some headlines, which are confined to the beginning and the end of the segments. Next, we analyze the news segments to remove the headlines and identify the story-boundaries. Finally, we extract keywords from speech and the ticker-text segments (wherever robust language tools are available) from these news stories and index them for later retrieval [10]. When the language tools are not available, the keywords are supplied by an operator. In the rest of this paper, we focus on the two stages of processing that involves advertisement detection and story segmentation.

III. STRUCTURE OF A NEWS PROGRAM

News programs follow a general structure as depicted in figure 2. A number of news segments are interspersed with commercial segments. The commercial segments contain several advertisement and usually spans over several minutes. A news segment consists of several news stories and headlines. News headlines present the summary of the news and is presented in either textual or visual format. They are generally confined to the beginning or at the end of a news segment. A news segment is also several minutes long. This structure has been observed over a large number of news channels. We assume a news program to follow this structure in the algorithms for commercial detection and story segmentation.
A. Ticker Text Bands

Ticker text is the text superimposed on the video frames during a news program and are used to convey news summary or headline and to catch the viewer’s attention. Our algorithms are based on movement of these ticker text bands between news programs and advertisements and repetition of the ticker texts during a news story. In general, three types of ticker texts are used during a news program

1) **Local Ticket Text** contains the highlights of the news currently being presented,

2) **Global ticker text** contains the highlights of all important stories in the news program, and

3) **Scrolling text** provides the gist of relatively unimportant news.

The different types of ticker texts are presented in distinct bands. The global ticker text and the scrolling text, if present, are generally at the bottom of the video frame. There can be one or two local ticker text bands, the ‘upper’ one generally near the top edge of the frame and the ‘lower’ one generally above the global ticker text. Ticker texts generally repeat during a news program. The news headlines are generally characterized by unique and non-repeating ticker texts. The normal positions of these bands, during a news presentation, are determined by the video editing style used by the news editor of a channel and are fixed. However, they change during the commercials. It has also been observed that the local ticker text band may not be present during the entire news story. However, at least one of the global ticker text or the scrolling text is always present during the complete news program. During advertisements, only one of the global or scrolling ticker text is present, which is either at the same position or shifts downward and rest of the bands are absent. We use these characteristics of ticker texts for advertisement detection and story segmentation. The exact positions of these bands may vary from channel to channel and can be ascertained by text localization methods and some heuristics. In our algorithms, we assume their normal positions (during news transmission) to be known. In some frames, e.g. news headlines and breaking news, more text may appear. We ignore these texts in our analysis. Figure 3 depicts the ticker text bands as typically seen in a news frame. The bands labelled R1 and R2 are the “Local” ticker text bands. The band R3 and R4 are the “Global” and the “scrolling” ticker ticker text bands.

IV. ADVERTISEMENT DETECTION

Our advertisement detection algorithm is based on change in position of the ticker text bands during the commercials. Since there is no current news item, the local ticker text bands disappear. One of the global or the scrolling ticker text band disappears and the other one either shifts further to the bottom (making more room for visuals) or may remain at the same location if it is already at the lowest part of the screen. In order to detect the movement of ticker text bands, we define a region of interest present at the lower part of the screen, for each channel, in which the text is present during the news but absent at the time of advertisements (see figure 4). Text is discriminated from non-text (empty or image) region by the presence of large number of edges. If text is not available in that region in consecutive frames for a significant duration, we conclude that the entire duration constitutes advertisement. Observation over a significant duration is necessary to eliminate any noise. Figure 5 provides the algorithm for advertisement detection and removal. We have used a very simple text detection algorithm for the sake of achieving speed. The ticker-text bands generally have a uniform background and the algorithm works with a fair degree of accuracy. Further, commercials always appear in large blocks. Use of $\tau$ and $T$ in “rounding off” the advertisement blocks adds to the reliability of results. Finally, the knowledge that the start and end of advertisement blocks must be aligned with shot boundaries help to identify their locations more accurately. At the end of this stage, the advertisement segments in a news program have been marked as $Ad = (adStart, adEnd)$. All contiguous segments separated by the advertisement segments are marked as news segments.
PROCEDURE: ADVERTISEMENT DETECTION
Input: A video corresponding to a news program
Output: An array of markers \(\{adBegin, adEnd\}[j]\) to mark the begin and the end of the advertisement blocks

1) Choose Region of Interest (RoI) to be the area on the lower part of the screen, on which ticker text is present during news but absent during advertisements.
2) For every frame \(F_i\) in the news program, create an Edge-image \(E_i\) of the RoI.
3) Classify the edge images \(E_i\) into two categories, those containing text and those not containing text. This is done by comparing the number of horizontal edges in \(E_i\) against a threshold, which is channel specific and determined based on observations over several hours of news. Label each frame \(F_i\) with the property ‘text’ or ‘non-text’.
4) Choose a sliding window of size \(M = T_{\text{min}} \times \text{fps}\), where \(T_{\text{min}}\) represents a possible minimum time for a commercial block in seconds and \(\text{fps}\) represents the frame-rate of the video. Also, choose a threshold \(\tau = k \times M\), where \(k\) is a fraction close to unity.
5) Create a flag \(A\) to indicate if the currently examined section is advertisement or not. Create array of tuples \(\{adStart, adEnd\}[j]\) to hold advertisement block information. Set \(A = false\), \(j = 0\)
6) Move the sliding window from the beginning of the news video program the video to the end. For each position of sliding window starting at frame \(F_i\)
   a) Count the number of \(E_i\)’s classified as ‘non-text’.
   b) If the count is greater than \(\tau\)
      i) If \(A = false\), then set \(A = true\) and \(adStart[j] = i\)
      ii) Else, do nothing (advertisement block continues)
   c) Else
      i) If \(A = true\), then set \(A = false\), \(adEnd[j] = i + M\) and \(j = j + 1\)
      ii) Else, do nothing (news block continues)
7) If \(j\), such that \(adEnd[j] < T\), when \(T\) is some threshold, then they are merged and the array is updated.
8) Each entry in the array \(\{adStart, adEnd\}\) is then aligned with the shot boundaries. Wherever an ad block overlaps more than half of a shot, the complete shot is assumed to be Advertisement, else it is considered as news.

\[\text{Figure 5: Algorithm for advertisement detection}\]

V. STORY SEGMENTATION

The basic principle behind story segmentation is that the local ticker texts that provide the highlight of the current news story, generally keeps on repeating during the news story and that they do not repeat across news stories. Figure 6 illustrates the principle. In the figure, T1 \ldots T5 represent some repeating ticker texts in any local ticker text bands. Note that the set \{T1,T2,T3\} are interlaced and so is \{T4,T5\}. The place where the set \{T1,T2,T3\} stops repeating and the set \{T4,T5\} appears marks a story boundary. Ticker text regions are marked manually. But any appropriate ticker text detection method can be used. The repetitions of the ticker texts are ascertained by image comparison of local ticker text bands without using any optical character recognition. There can be short regions between repeating local ticker text sets, which are devoid of any local ticker text. This region can be a part of either the preceding or the following story. Since their classification cannot be further determined, they are merged with both the adjacent stories. If there is long enough region between two stories, with non-repeating local ticker text, it is considered as an independent story. Figure 7 provides the algorithm for news story boundary detection. We make a few amendments to the algorithm described in figure 7 to take care of a few special cases.

1) A headline, together with its ticker-text can repeat at the two ends of a news segment making the extent for the ticker text spanning over all news stories in the segment. This would cause the algorithm to fail. We observe that
   a) The headlines are restricted within first or the last quarter of a news segment, and
   b) The ticker-text on the headlines are distinct from those on the news stories.

Thus, to avoid the repetition of headline, we apply the algorithm independently to first 75% and to the last 75% of a news-segment. The two sets of results are merged to identify story boundaries.

2) Very short stories, which are less than a certain time-period in duration, appearing at the beginning with no long story preceding them or at the end with no long story following them are discarded as headlines.

3) It is not necessary the ticker texts starts and ends exactly at the boundaries of a news story. Following methods are used to extend the story boundaries to cover parts of a news segment, which do not form part of any story following the above algorithm:
   a) Story boundaries are always marked with a visual change in the scene. Stories are therefore aligned to shot-boundaries as in the case of advertisement detection.
   b) There can be shots between the stories without any ticker-text. These unclaimed shots can be
PROCEDURE: STORY BOUNDARY DETECTION
Input: A contiguous news segment
Output: A set of story extents (start time, end time)
1) For every extracted frame $F_i$ of the news segment and every local ticker text band $j$, create a cropped image $R_{ij}$ to retain the regions corresponding to the local ticker text bands.
2) For every $R_{ij}$, create an Edge image $E_{ij}$. (see Figure 8)
3) Classify the edge images $E_{ij}$ into two categories, those containing text and those not containing text. Label each image $E_{ij}$ with the property ‘text’ or ‘non-text’.
4) For every local ticker text band $j$
   a) Create a set of Ticker Text Blocks $TTB_j[k]$, each comprising a set of images $E_{ij}$ with contiguous values of $i$. Each $TTB$ contains one or more distinct ticker text instances.
   b) For each $TTB$, compare successive images $E_{ij}$ and $E_{i+1,j}$ and identify contiguous blocks of Stable Ticker Texts (STTB). An STTB comprises of a set of contiguous Frames $\{F_i, i = x \ldots y\}$, such that $\forall i \in x \ldots y - 1$, $diff(E_{ij}, E_{i+1,j}) < \delta$, where $diff$ represents an absolute difference measure between two images based on template matching algorithm and $\delta$ denotes a threshold. Each STTB contains exactly one stable ticker text. Take the middle frame in a STTB as the Representative Ticker Text Image $RTTI$.
   c) The RTTIs are then pair-wise compared to get a set of Unique Ticker Text Images. Each STTB’s are now renumbered as $STTBJ[pq]$, where $j$ indicates ticker text band, $p$ indicates a unique ticker text number and $q$ indicates the instance of occurrence of the unique ticker text.
   d) Define extent of a unique ticker text $p$ in band $j$ as $extent_{jp} = (sjp, ejp)$, where $sjp$ represents the start time for its first occurrence and $ejp$ represents the end time of its last occurrence.
5) Collate the set of all extents, irrespective of which local ticker text band they may belong to. Renumber the extents as $extent_1 \ldots extent_n$, where $n = \sum_j n_j$, where $n_j$ represents the number of unique ticker texts found in local ticker text band $j$.
6) Compare the extents pair-wise. Merge two extents $extent_p$ and $extent_q$, if $sjp < ejp_q$ AND $ejp_p > sjp_q$. Continue till closure, i.e. till no two extents can be merged.
7) The extents after merge are treated as the extents of the stories. If a story is created by merging a set of extents $extent_i (i=1..k)$, the story start and end times are computed as $min_i(sjp_i)$ and $max_i(ejp_i)$ respectively.

Figure 7: Algorithm for story boundary detection

part of either the preceding or the succeeding story and further resolution is not possible. They are included in both the stories, so that any important information is not missed out.

Figure 9 shows extension of story boundary to shot boundaries and to cover unclaimed shots.

VI. EXPERIMENTS AND RESULTS
A. Experimental setup
We have created an infrastructure to monitor a large number of Indian telecast channels as well as international transmissions available in India. We have experimented with 6 news channels, out of which 3 channels are in English and the others are in different Indian Languages. We have recorded 50 hours of news programs from these channels, out of which 20 hours are in Indian languages, for which language tools do not exist.

B. Experimental Results
We measure the accuracy of the advertisement detection in terms of total duration of the shots correctly classified as advertisements in a news program. The accuracy of the algorithm is defined as: \[Accuracy = \frac{TP + TN}{TP + FP + TN + FN}\]

- $TP$ = total duration of shots correctly classified as advertisement
- $FP$ = total duration of shots incorrectly classified as advertisement (they are actually news)
- $TN$ = total duration of shots correctly classified as news
- $FN$ = total duration of shots incorrectly classified as news (they are actually advertisements)

We consider story boundary to be a success, if both the start and the end times of the stories are identified with a maximum deviation of two shots. If $N$ denotes the number of stories in a news program and if $n$ is the number of stories identified with an error of $\pm 2$ shots, the accuracy is defined as: \[Accuracy = \frac{n}{N}\]. Results are shown in the Table I.
Table I
EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>News Channel</th>
<th>Language</th>
<th>Duration (in hours)</th>
<th>Accuracy (%) Ad</th>
<th>Accuracy (%) Story</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNIBN</td>
<td>English</td>
<td>10</td>
<td>98.2</td>
<td>90</td>
</tr>
<tr>
<td>NDTV</td>
<td>English</td>
<td>10</td>
<td>99.2</td>
<td>91.7</td>
</tr>
<tr>
<td>TimesNow</td>
<td>English</td>
<td>10</td>
<td>97</td>
<td>91.9</td>
</tr>
<tr>
<td>Star Majha</td>
<td>Marathi</td>
<td>5</td>
<td>94</td>
<td>87.5</td>
</tr>
<tr>
<td>Zee24Taas</td>
<td>Marathi</td>
<td>5</td>
<td>98.6</td>
<td>88.5</td>
</tr>
<tr>
<td>AajTak</td>
<td>Hindi</td>
<td>10</td>
<td>94</td>
<td>81.6</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>50</td>
<td>96.8</td>
<td>88.5</td>
</tr>
</tbody>
</table>

C. Result Analysis

In advertisement detection, we have achieved an overall 97% accuracy. Most of the false positives (news mistakenly classified as text) occur because of that text does not persist in the region of interest throughout news segment. False negatives (advertisements mistakenly classified as news) are primarily caused by incorrect classification of video texture as text. Though most of these errors are taken care of by rounding off the advertisement blocks to large contiguous segments, a few residual errors are found at the advertisement block boundaries, i.e. at the time of transitions from news to commercials and vice versa. Since the goal of the system is to assist human users to deal with the news stories, loss of story information is a more serious problem than a few extra advertisement shots being presented. To alleviate the issue of missing news story, we extend the story boundaries into the advertisement blocks by a few shots during presentation. We have achieved an overall 89% accuracy in story boundary detection. Most of the errors occurred when one news story has been detected as multiple stories, because of disjoint sets of ticker texts appearing at different parts of the story. In such cases, we have observed that the different parts of the story dwells on different aspects of the story, e.g. rescue operations and sentiments expressed after an accident. Thus, recording such story segments as independent stories can be viewed more as a policy decision than an error.

D. Performance

We have conducted our experiments on a desktop PC having Intel Core2Duo 3Ghz processor, and 2GB ram and Windows-XP operating system. It takes 27 minutes for advertisement detection and 38 minutes for story detection to process one hour of news video program. The system described in section II dedicates a processor for every channel to be processed. This is sufficient to process news channels in near real time, since we process news programs only and one channel at a time. The advantage of pipelining the advertisement detection and story-boundary detection algorithms for successive hours of news video on a dual-core processor needs to be explored.

VII. CONCLUSION

The novel algorithm presented for advertisement detection and story segmentation using ticker text information present is working decently. Assuming the given editing style observed over a number of International, National and Regional channels transmitted in India, the presented algorithm works so long the assumptions are met. Assumptions anyway are necessary. We observed ‘a few’ local channels where these assumptions are violated. Our method need to be adopted for these channels. They can be classified into a few generic classes and a variant of the algorithm can be applied for each. If there will be any change in ticker text regions that can be refined.

REFERENCES