EXPLOITING TOPIC THREAD STRUCTURES IN A NEWS VIDEO ARCHIVE FOR THE SEMI-AUTOMATIC GENERATION OF VIDEO SUMMARIES

Ichiro IDE †,‡ *Hiroshi MO* ‡ *Norio KATAYAMA* ‡ *Shin'ichi SATOH* ‡

† Nagoya University
Graduate School of Information Science
1 Furo-cho, Chikusa-ku,
Nagoya, 464–8601, Japan
ide@is.nagoya-u.ac.jp

‡ National Institute of Informatics
Research Divisions
2–1–2 Hitotsubashi, Chiyoda-ku,
Tokyo, 101–8430, Japan
{ide, mo, katayama, satoh}@nii.ac.jp

ABSTRACT

We propose a method that semi-automatically composes *video stories* by connecting individual stories in a news video archive along a topic-based semantic structure, namely the *topic thread*. We introduce the methods to realize the composition, namely, story segmentation, topic threading and clustering. We then evaluate the proposed approach based on preliminary tests. Since the thread structure reflects the development of topics in the real-world, we believe that the composed news *video story* should be effective for the user to gain a deeper understanding of the current topic of interest.

1. INTRODUCTION

News is a chronological representation of events, where the connection of details over days provides the context as well as the importance of the provided information. To understand a certain topic of interest, it is important to track news videos of related events up and down the timeline. It is, however, almost impossible to manually find closely related events from within the massive amount of video data in the continuously growing news archives. For example, the archive we are working with, consists of 700 hours of daily Japanese news videos.

In order to assist a user with such a task, we propose a video summarization method, that automatically composes a *video story* [1] by connecting news stories in the archive so that it should explain the development between two specified stories, based on a semantic structure within a news video archive, namely, the *topic thread* structure.

The *topic thread* structure, as exemplified in Fig. 1, is a structure represented as an acyclic directed graph that links related stories maintaining chronological orders at each edge.

[1]The definition of the term 'story' in the Topic Detection and Tracking (TDT) workshop series organized by NIST, is somewhat different from its general use. It is used in 'video story' in the latter sense, that it is a video stream that explains the development of a topic, by connecting (TDT-defined) stories from the beginning to the end. In this paper, it is denoted in italic when used in the general sense, such as *video story* and *story*.

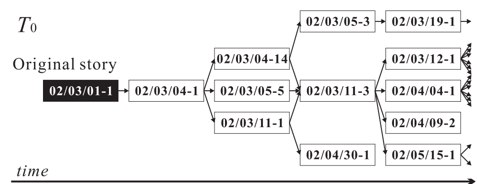

Fig. 1. Example of a topic thread structure. Topics are labeled as [Year/Month/Day-Topic Number].

It represents local relations by directly connected edges, and at the same time the development of a topic as a sequence of stories, with as few edges as possible at each node. We have previously developed a news video tracking interface that allows a user to track story after story based on the structure [1]. With this system, however, it was not clear which story the user was guided to, or which thread should be followed to understand the development of a topic of interest.

Bocconi et al. [2] propose a semi-automatic documentary video generation method by composing statements on a topic by various people. The approach is interesting as it provides the means to establish larger argument structures based on basic statements. Also, documentaries and news are very similar genres. Their method, however, requires manual annotations, which makes it difficult to be applied to a large archive.

Our method is based on the premise that the result of an event is unpredictable in the real-world. The method thus tries to compose a *story* that connects two sometimes seemingly unrelated events by connecting locally related stories.

The paper is organized as follows: Section 2 describes the establishment of the topic thread structure, followed by Section 3 which describes the composition of *video stories* from the topic threads. Section 4 concludes the paper.

2. STRUCTURING THE ARCHIVE

In this Section, we first describe how the news inherent structures are automatically analyzed in 2.1 (story segmentation).

In 2.2, we then outline how the obtained structures are further analyzed; relations between the segmented stories are analyzed according to their chronological and semantic relations (topic threading). A detailed description is provided in [1].

2.1. Story segmentation

In order to establish the topic thread structure, it is first necessary to extract the stories within a news show. The following process is applied to each sentence of a closed-caption text synchronized to the audio track:

1. Apply morphological analysis [2] to each sentence. Next, extract noun compounds according to the morphemes, followed by semantic attribute analysis by a suffix-based method proposed in [3].
2. Create keyword vectors for each sentence. Keyword vectors for four semantic attributes; general, personal, locational/organizational, and temporal, are formed by noun compounds extracted in Step 1.
3. At each sentence boundary, concatenate w ($= 1$ to 10 in the following experiments) adjacent vectors on both sides. Measure the similarity of the two concatenated vectors by the cosine measure, and choose the maximum similarity among all window sizes.
4. Sum up the similarities in each semantic attribute and detect a story boundary when it is smaller than θ_{seg}. According to a training with 384 manually given story boundaries, a weight of (general, personal, locational/ organizational, temporal) = (0.23, 0.21, 0.48, 0.08) for the summation and $\theta_{seg} = 0.17$ were obtained.

According to evaluations applied to 130 manually annotated story boundaries as ground-truth, a precision of 90.5% and a recall of 95.4% were achieved if mis-judgments at a maximum of ± 1 sentences were allowed.

2.2. Topic threading

Having segmented the various stories within a news show, we now describe how we establish the relations between them. The difference between the topic thread structure with a hierarchical tree that simply expands related stories at each node is that the it lets a story appear only once in the tree where it is a child of the chronologically closest story (Compare Figs. 1 and 2). The topic thread structure is extracted as follows:

1. Expand a story relation tree recursively from the original story satisfying the following conditions:

    (a) Child nodes are stories related to their parent node, while at the same time, their time stamps succeed their parent's.

[2]JUMAN 3.61 distributed from Kyoto University was used.

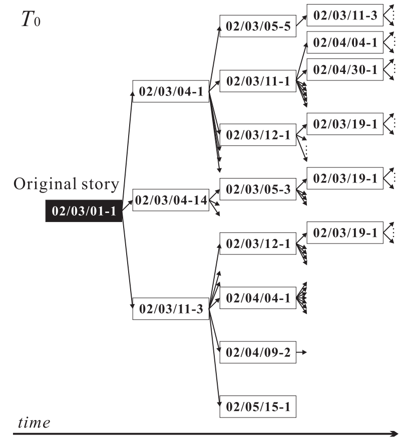

Fig. 2. Example of a simple hierarchical story relation tree without threading.

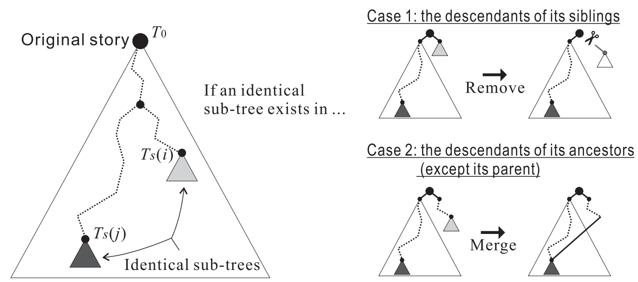

Fig. 3. The topic threading scheme.

    (b) Siblings are sorted so that their time stamps succeed their left-siblings'.

   The relation between two stories is defined as the cosine measure between the keyword vectors of the stories. When its value exceeds a threshold θ_{trk}, the stories are considered related. This procedure forms a hierarchical story relation tree T_0 as shown in Fig. 2.

2. For each sub-tree $T_s(i)$ in the story relation tree T_0, if an identical sub-tree $T_s(j)$ exists on the left-side (in the future), apply either of the following operations:

    (a) Remove $T_s(i)$ if $T_s(j)$ is a descendant of $T_s(i)$'s sibling.

    (b) Else, merge $T_s(i)$ with $T_s(j)$ if $T_s(j)$ is a descendant of $T_s(i)$'s ancestor excluding its parent.

   The sub-tree is removed in (a) instead of merging, to avoid creating a shortcut without a story en route. The scheme is illustrated in Fig. 3. As a result of this operation, the thread structure will form a chronologically-ordered directed graph as shown in Fig. 1.

Topic tracking has been a strong interest in the text retrieval field, as seen in the TDT workshop series. They define the term 'topic' as "*a seminal event or activity, along with all directly related events and activities*". Compared to this definition, the *topic thread* is slightly different in the sense that it connects gradually developing stories even across topics, where topic tracking generally terminates when it encounters a certain degree of gradual transition from the original story. The thread structuring may also seem similar to the Hierarchical Topic Detection task in TDT-2004. This again is different, since their aim is to analyze the hierarchical structure of subtopics within a topic, while our approach connects gradually shifting topics, and as a result connects topics that are locally related but not necessarily consistent at both ends.

2.3. Implementation

The implementation is done so that it processes incoming news video automatically everyday after the broadcast. Since March 16, 2001, a daily Japanese news show with closed-caption texts is archived for more than 1,650 days. For each show, the stories are segmented within 1 min.[3], which has so far derived approximately 20,000 stories.

Evaluations of relations between new stories and all existing stories follows the segmentation, which currently takes approximately 1 hour. Tables with relations between all combinations of story pairs are generated during this process, since it consumes too much time to do so each time when a real-time interface needs to generate a thread structure.

3. COMPOSING A VIDEO STORY

This Section discusses the composition of a *video story* from a selected topic thread. Since we have just started working on it, the following part mostly discusses ideas and statistical numbers so far obtained from preliminary experiments.

3.1. Finding stories in the thread structure

We consider the following two approaches that involve interactions with users to find *video story* candidates:

Approach 1: Connect two manually specified stories
Users are requested to specify two stories; the origin S_O and the destination S_D. Hereafter, we denote a *video story* that connects stories S_O and S_D as $V_S(S_O, S_D)$.

In order to find threads that represent a *story* $V_S(S_O, S_D)$, we need to dynamically adjust the threshold θ_{trk} to find a thread structure that originates from S_O and contains S_D, if any. Our implementation tries to find such a structure by starting from a very high value and gradually lowering it. If S_O and S_D are not connected by a thread with closely related stories, the process will start searching through a very large thread structure connecting less related stories as θ_{trk} is lowered, which will not return a result in realistic time.

Approach 2: Propose candidates originating from a specified story
Users are requested to specify only the original story S_O. First, the system retrieves a thread structure that originates from S_O with two pre-defined parameters; the relation threshold θ_{trk} and the search range d. Next the system considers that the leaf nodes $S_D(i)(i = 1, 2, ..., L)$ are candidates for the destination, and presents them to the users for selection.

Fig. 4(a) shows an example of an actual thread structure. In this example, six *video story* candidates $V_S(S_O, S_D(i))(i = 1, 2, ..., 6)$ are extracted from the structure.

As a preliminary experiment, we analyzed thread structures for 1,431 stories obtained from 120 shows in March to June, 2002, with the parameters; $\theta_{trk} = 0.40, d = 100$ days. As a result, 310 stories returned thread structures deeper than 3 layers, where 1,577 *video story* candidates were found. The process took an average of 5 secs. per structure, with a maximum of 1,233 secs. excluding one complex case that took 10 hours. In total, 96.5% out of the 310 structures took less than 1 min. to analyze, which is quick enough for our purpose.

3.2. Topic clustering in the thread structure

After a *video story* is specified by either approach described in 3.1, a certain path in the specified thread structure should be selected to compose the *story*, since in most cases, there are several paths that may connect the original and the destination stories. Moreover, simply connecting all the stories along a thread should make the summarized *video story* too lengthy.

To handle these problems, topic clusters composed of similar stories along a thread are detected. This allows us to select a path which crosses as few topic clusters as possible so that it should explain a *story* as simple as possible, while at the same time, generally composing a short *video story*.

Topic clusters along a thread is detected as follows:

1. Set S_O as both the cluster center ($N_0 = S_O$) and the current node ($N = S_O$).
2. Let child nodes of N be $N_c(j)(j = 1, ..., C)$. If none of the relations between N_0 and $N_c(j)$ exceed a threshold θ_{cls}, set N as the new cluster center ($N_0 = N$).
3. Apply Step 2. recursively until reaching all the leaf nodes $S_D(i)(i = 1, ..., L)$.

This is different from the general clustering-based topic detection, since it is applied after tracking across multiple topics to find local topic boundaries along a thread without altering the structure. The shortest (in terms of the number of topics involved) thread for each *story* is chosen as the thread that explains it. For example, $V_S(S_O, S_D(6))$ involves topic clusters $C_T(1), C_T(2), C_T(3), C_T(4), C_T(10)$, and $C_T(11)$, as can be seen from Fig. 4(b).

[3]The processing times noted hereafter are user times measured on a Sun Blade 1000 workstation with dual UltraSparc III 750MHz processors and 2GB of main memory.

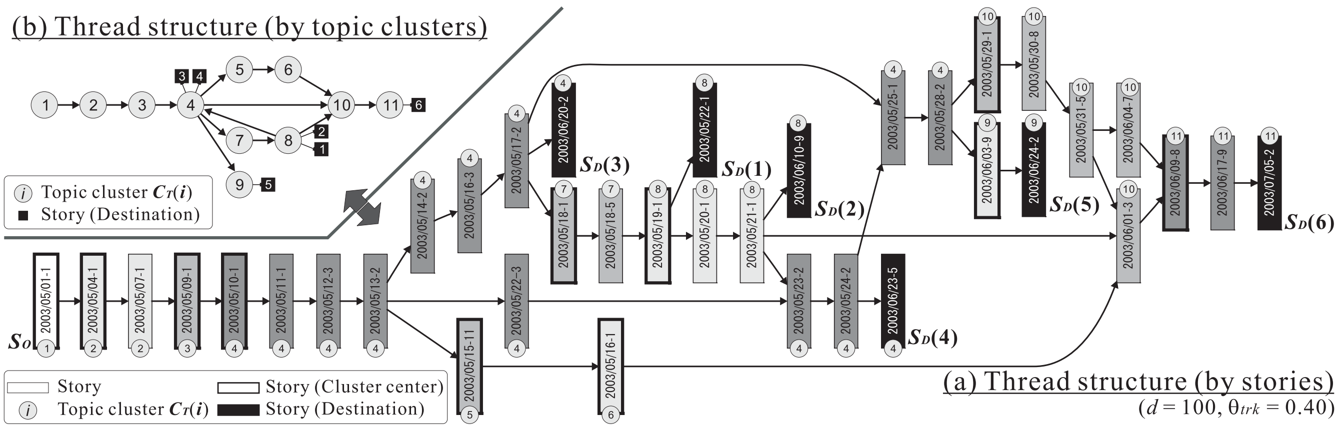

Fig. 4. Topic thread structure originating from Story #1 on May 1, 2003. Video story $V_S(S_O, S_D(6))$ is composed as follows: Story S_O (= Topic cluster $C_T(1)$): SARS outbreak in Beijing; $C_T(2)$: Epidemic spreads in mainland China; $C_T(3)$: WHO sends a mission to Beijing; $C_T(4)$: Epidemic slows down in mainland China, spreads in Taiwan; $C_T(10)$: Epidemic calms down in mainland China, some infection reported in Toronto; $C_T(11)$: Epidemic calms down in Taiwan; Story $S_D(6)$: WHO announces the cease of the epidemic. It took 23 secs. to obtain this structure, including the threading and the clustering.

Topic clustering applied to the threads for each *story* candidate found in the experiment for Approach 2 in 3.1 showed the following averages when $\theta_{cls} = \theta_{trk} = 0.40$.

- Period: 59.9 days (1 $\sim$ 100)
- Stories involved: 8.44 (3 $\sim$ 29)
- Topics involved: 3.36 (1 $\sim$ 9)[4]

This shows that most of the *stories* within the thread structure explained their development across multiple topics by an average of one story per week during a period of two months. These are the *stories* that would not have been extracted by general topic detection and summarization methods.

3.3. Composing a summarized video story

At this moment, a summarized *video story* is composed by connecting the stories that were cluster centers in the process described in 3.2. Note that as a side effect, if we only consider connecting the topic clusters, we could ignore the minor branching and merger of stories within a topic cluster, which should make the *video story* simpler. The summarized *video stories* showed the following averages:

- Original video length: 3,106 secs. (9 $\sim$ 19,038)
- Summarized video length: 1,415 secs. (7 $\sim$ 6,741)
- Summarization rate: 58% (4 $\sim$ 100)

The original video tends to be too lengthy for a user to watch casually, which indicates the needs for summarization. In some cases, the summarized video is still too lengthy, but the summarization rate could be adjusted by tuning θ_{cls} and elaborating the summarization scheme in the future.

[4]Approximately 87% of them involved more than 2 topics.

4. CONCLUSIONS

We proposed a semi-automatic method to compose a *video story* by connecting news stories along a chronological semantic structure called a *topic thread* in a news video archive. Although quality evaluation and details on the composition of the *video story* involving users is yet to be done, the topic clusters in the thread structure were consistent to the development of actual *stories* in the real-world according to our knowledge. We will also develop a faster topic threading algorithm to apply the process to a longer time range.

5. ACKNOWLEDGEMENTS

We would like to thank Dr. Frank Nack from CWI Amsterdam for his valuable comments on our work. The presented works were supported by the Grants-in-Aid for Scientific Research from MEXT, and the Research Grant from Kayamori Foundation of Information Science Advancement.

6. REFERENCES

[1] I. Ide, H. Mo, N. Katayama, and S. Satoh, "Topic threading for structuring a large-scale news video archive," in *Image and Video Retrieval –Third Intl Conf, CIVR2004, Procs–*, July 2004, vol. LNCS 3115, pp. 123–131.

[2] S. Bocconi, F. Nack, and L. Hardman, "Using theoretical annotations for generating video documentaries," in *Proc IEEE 2005 Intl Conf on Multimedia and Expo*, July 2005.

[3] I. Ide, R. Hamada, S. Sakai, and H. Tanaka, "Semantic analysis of television news captions referring to suffixes," in *Proc 4th Intl Workshop on Information Retrieval with Asian Languages*, Nov. 1999, pp. 37–42.