Unsupervised Text Recap Extraction for TV Series

Hongliang Yu and **Shikun Zhang** and **Louis-Philippe Morency**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{yuhongliang, shikunz, morency}@cs.cmu.edu

Abstract

Sequences found at the beginning of TV shows help the audience absorb the essence of previous episodes, and grab their attention with upcoming plots. In this paper, we propose a novel task, text recap extraction. Compared with conventional summarization, text recap extraction captures the duality of summarization and plot contingency between adjacent episodes. We present a new dataset, *TVRecap*, for text recap extraction on TV shows. We propose an unsupervised model that identifies text recaps based on plot descriptions. We introduce two contingency factors, *concept coverage* and *sparse reconstruction*, that encourage recaps to prompt the upcoming story development. We also propose a multi-view extension of our model which can incorporate dialogues and synopses. We conduct extensive experiments on *TVRecap*, and conclude that our model outperforms summarization approaches.

1 Introduction

According to a study by FX Networks, in U.S., the total number of ongoing scripted TV series hit a new high of 409 on broadcast, cable, and streaming in 2015[1]. Such a large number indicates there are more shows than anyone can realistically watch. To attract prospective audiences as well as help current viewers recall the key plot when airing new episodes, some TV shows add a clip montage, which is called a recap sequence, at the beginning of new episodes or seasons. Recaps not only help the audience absorb the essence of previous episodes, but also grab people's attention with upcoming plots. However, creating those recaps for every newly aired episode is labor-intensive and time-consuming. To our advantage, there are many textual scripts freely available online which describe the events and actions happening during the TV show episodes[2]. These textual scripts contain *plot descriptions* of the events, *dialogues* of the actors, and sometimes also the *synopsis* summarizing the whole episode.

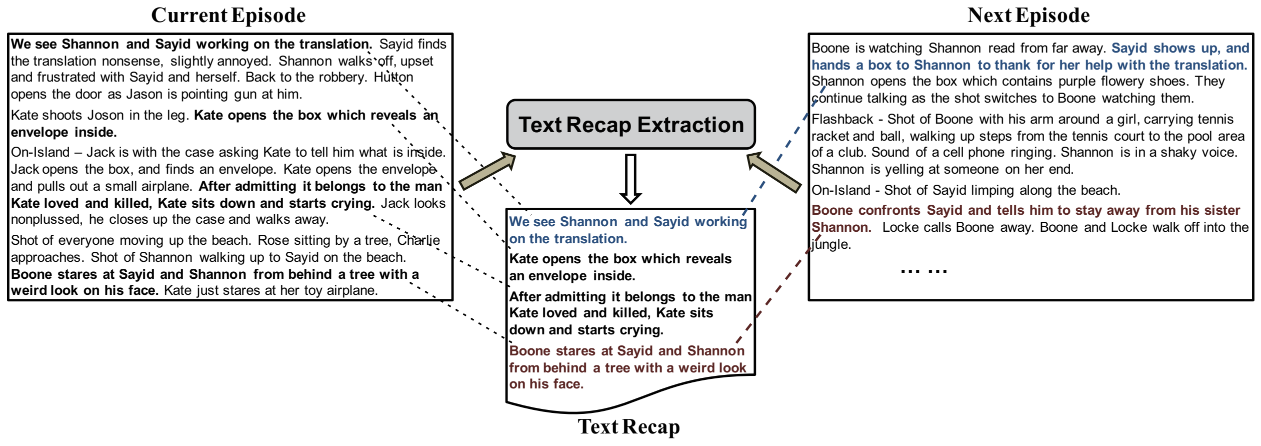
These abundant textual resources enable us to study a novel, yet challenging task: automatic text recap extraction, illustrated in Figure 1. The goal of text recap extraction is to identify segments from scripts which both summarize the current episode and prompt the story development of the next episode. This unique task brings new technical challenges as it goes beyond summarizing prior TV episodes, by introducing a concept of plot contingency to the upcoming TV episode. It differs from conventional summarization techniques which do not consider the interconnectivity between neighboring episodes. Text recaps should capture the duality of summarization and plot contingency between neighboring episodes. To our knowledge, no dataset exists to study this research topic.

In this paper, we present an unsupervised model to automatically extrapolate text recaps of TV shows from plot descriptions. Since we assume recaps should cover the main plot of the current episode and also prompt the story development of the next episode, our model jointly optimizes these two ob-

[1] http://tinyurl.com/jugyyu2

[2] http://www.simplyscripts.com/tv_all.html

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1797–1806,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics



We see Shannon and Sayid working on the translation. Sayid finds the translation nonsense, slightly annoyed. Shannon walks off, upset and frustrated with Sayid and herself. Back to the robbery. Hutton opens the door as Jason is pointing gun at him.

Kate shoots Joson in the leg. **Kate opens the box which reveals an envelope inside.**

On-Island – Jack is with the case asking Kate to tell him what is inside. Jack opens the box, and finds an envelope. Kate opens the envelope and pulls out a small airplane. **After admitting it belongs to the man Kate loved and killed, Kate sits down and starts crying.** Jack looks nonplussed, he closes up the case and walks away.

Shot of everyone moving up the beach. Rose sitting by a tree, Charlie approaches. Shot of Shannon walking up to Sayid on the beach. **Boone stares at Sayid and Shannon from behind a tree with a weird look on his face.** Kate just stares at her toy airplane.

We see Shannon and Sayid working on the translation.

Kate opens the box which reveals an envelope inside.

After admitting it belongs to the man Kate loved and killed, Kate sits down and starts crying.

Boone stares at Sayid and Shannon from behind a tree with a weird look on his face.

Boone is watching Shannon read from far away. **Sayid shows up, and hands a box to Shannon to thank for her help with the translation.** Shannon opens the box which contains purple flowery shoes. They continue talking as the shot switches to Boone watching them.

Flashback - Shot of Boone with his arm around a girl, carrying tennis racket and ball, walking up steps from the tennis court to the pool area of a club. Sound of a cell phone ringing. Shannon is in a shaky voice. Shannon is yelling at someone on her end.

On-Island - Shot of Sayid limping along the beach.

Boone confronts Sayid and tells him to stay away from his sister Shannon. Locke calls Boone away. Boone and Locke walk off into the jungle.

… …

Figure 1: Illustration of text recap extraction. The system extracts sentences from the current episode. The text recap sentences in black summarize the current episode, while colored sentences motivate the next episode.

jectives. To summarize the current episode, our model exploits coverage-based summarization techniques. To connect to the next episode, we devise two types of plot contingency factors between adjacent episodes. These factors implement the coverage and reconstruction assumptions to the next episode. We also show how our model can be extended to integrate dialogues and synopses when available.

We introduce a new dataset[3], named *TVRecap* for text recap extraction which consists of TV series with textual scripts, including descriptions, dialogues and synopses. The dataset enables us to study whether contingency-based methods which exploit relationships between adjacent episodes can improve summarization-based methods.

The rest of this paper is organized as follows. In Section 2, we discuss related work and the motivation for our work. In Section 3, we introduce our new dataset for text recap extraction. Section 4 explains our proposed model for text recap extraction, and Section 5 expands the model by incorporating synopses and dialogues. In Section 6 and 7, we present our experimental results and analyses, and finally conclude our work in Section 8.

2 Related Work

In this section, we discuss three related research topics. Text summarization is an relevant task that aims to create a summary that retains the most important points of the original document. Then we discuss the evaluation metrics of text summarization. Finally, we discuss the video description which is complementary to our work.

Generic Text Summarization Alogrithms Text summarization is widely explored in the news domain (Hong and Nenkova, 2014; McKeown, 2005). Generally, there are two approaches: *extractive* and *abstractive* summarization.

Extractive summarization forms a summary by choosing the most representative sentences from the original corpus. The early system LEAD (Wasson, 1998) was pioneering work. It selected leading text of the document as the summary, and was applied in news searching to help online customers focus their queries on the beginning of news documents. He et al. (2012) assumed that summarization should consist of sentences that could best reconstruct the original document. They modeled relationship among sentences by forming an optimization problem. Moreover, Sipos et al. (2012) and Lin and Bilmes (2010) studied multi-document summarization using coverage-based methods. Among them, Lin and Bilmes (2010) proposed to approximate the optimal solution of a class of functions by exploiting submodularity.

Abstractive summarization automatically create new sentences. For example, compared with the sentence-level analysis in extractive summarization, Bing et al. (2015) explored fine-grained syntactic units, i.e. noun/verb phrases, to represent concepts in input documents. The informative phrases were

[3]http://multicomp.cs.cmu.edu

then used to generate sentences.

In this paper, we generalize the idea of text summarization to text recap extraction. Instead of summarizing a given document or collection, our model emphasizes plot contingency with the next episode.

Summarization Applications Summarization techniques are not restricted to informative resources (e.g. news), applications in broader areas are gaining attention (Aparício et al., 2016). As the prevailance of online forums, Misra et al. (2015) developed tools to recognize arguments from opinionated conversations, and group them across discussions. In entertainment industry, Sang and Xu (2010) proposed a character-based movie summarization approach by incorporating scripts into movie analysis. Moreover, recent applications include multimedia artifact generation (Figueiredo et al., 2015), music summarization (Raposo et al., 2015) and customer satisfaction analysis (Roy et al., 2016).

Video Description Generating video descriptions is a task that studies automatic generation of natural language that describes events happening in video clips. Most work uses sequential learning for encoding temporal information and language generation (Guadarrama et al., 2013; Rohrbach et al., 2013, 2015; Donahue et al., 2015). Our work is complementary to video description: the large number of unlabeled videos can be utilized to train end-to-end recap extraction system when video description models can properly output textual descriptions.

Contributions of This Paper In contrast with prior work, the main contributions of this paper are:
(1) We propose a novel problem, text recap extraction for TV series. Our task aims to identify segments from scripts which both summarize the current episode and prompt the story development of the upcoming episode;
(2) We propose an unsupervised model for text recap extraction from descriptions. It models the episode contingency through two factors, next episode summarization and sparse reconstruction;
(3) We introduce a new dataset for TV show recap extraction, where descriptions, dialogues and synopses are provided.

3 The TVRecap Dataset

We collected a new dataset, called *TVRecap*, for text recap extraction on TV series. We gathered and processed scripts, subtitles and synopses from websites[4] as components to build our model upon. We also established ground truth to help future research on this challenging topic. *TVRecap* includes all seasons from the widely-known show "*Lost*" with a total of 106 episodes. Statistics of our dataset are shown in Table 1.

	# sent.	avg. # sent.	# words	avg. # w./s.
description	14,686	138.5	140,684	9.57
dialogue	37,714	355.8	284,514	7.54
synopsis	453	4.27	7,868	17.36
recap	619	17.19	5,892	9.52

Table 1: Statistics of *TVRecap*.

This section describes how textual scripts and synopses are processed, and how we automatically define the ground truth of text recap annotations.

Descriptions, Dialogues and Synopses A script for one TV series episode is a sequence of dialogues interleaved with descriptions (marked by square brackets). We automatically split the script into descriptions and dialogues. For each episode, We also downloaded the synopsis, a human-written paragraph summarizing the main plot of the episode. Figure 2 shows examples of a script and a synopsis from our *TVRecap* dataset.

LOCKE: Two players. Two sides. One is light... one is dark. Walt, do you want to know a secret?

[*Claire writing in her diary. Jin approaches and offers her some urchin. She shakes her head, but then gives in and takes some.*]

CLAIRE: No. Thank you. No, it's okay. [*Jin keeps insisting*] No, really. Okay. Thanks.

(a) Script: containing descriptions and dialogues.

Boone steals the decreasing water supply in a misguided attempt to help everyone, but the survivors turn on him. A sleep-deprived Jack chases after what appears to be his deceased father in the forests and eventually discovers caves with fresh water. Jack comes to terms with his role as leader. In flashbacks, Jack goes to Australia to retrieve his deceased father.

(b) Synopsis.

Figure 2: Example of a script (including descriptions and dialogues) and a synopsis.

[4]`http://lostpedia.wikia.com/` and `https://www.wikipedia.org/`

All plot descriptions and dialogues are time-aligned automatically using the subtitle files[5]. We first aligned the dialogue sentences from the script with the subtitle files which contain time-stamps (in milliseconds) of the spoken dialogues. Then we estimated time-stamps of description sentences using surrounding dialogues.

Since descriptions sometimes contain words not relevant to the event, we manually post-processed all descriptions and recap sentences as follows: (1) remove trivial sentences such as "*music on*", (2) remove introductory terms like "*Shot of*", (3) complete missing grammatical components (like omitted subjects) of sentences when possible.

Text Recap Annotations The goal of our ground truth annotation is to identify the text descriptions associated with the TV show recap. We performed this annotation task in three steps.

First, we automatically extracted the recap sequence, which is a montage of important scenes from previous episodes to inform viewers of what has happened in the show, from the TV show video. These recap sequences, if available, are always shown at the beginning of TV episodes. We automatically separated video recap sequences from full-length video files by detecting a lengthy appearance of black frames in the first several minutes of the episode. Second, we located the frames of the recap sequences in the videos of previous episodes, and recorded their time-stamps. Finally, the recap annotations are automatically identified by comparing the video time-stamps with the text description time-stamps. A description is annotated as part of the recap if at least 4 frames from the video recap are present during this description.

4 Our Text Recap Extraction Model

In our Text Recap Extraction Model (TREM), we assume a good text recap should have two characteristics: (a) it covers the main plot of the current episode, and (b) it holds plot contingency with the next episode. Under the first assumption, the text recap can be seen as a summarization that retains the most important plots. Under assumption (b), the text recap should capture the connections between two consecutive episodes.

[5]http://www.tvsubtitles.net/

Formally, the system is given E episodes from a specific TV show, where each episode contains textual descriptions. We define these descriptions as $D = \{D^1, \cdots, D^E\}$, where D^i is the set of descriptions of episode i. D^i is composed of descriptive sentences as $D^i = \{d^i_1, \cdots, d^i_{|D^i|}\}$, where d^i_j is the j-th sentence. The goal of our task is to find text recaps $R = \{R^1, \cdots, R^{E-1}\}$ where the components of R^i are selected from D^i with a length budget (constraint on the number of sentences) $|R^i| \leq K$.

In our TREM model, the text recap R^i of the i-th episode is optimized by:

$$\begin{aligned} \max_{R^i \subset D^i} \quad & \mathcal{F}(R^i) = \mathcal{S}(R^i, D^i) + \mathcal{M}(R^i, D^{i+1}) \\ \text{s.t} \quad & |R^i| \leq K, \end{aligned} \tag{1}$$

where $\mathcal{S}(R^i, D^i)$ measures how well R^i summarizes D^i, and $\mathcal{M}(R^i, D^{i+1})$ quantifies the level of connectivity between the text recap of the current episode and the plot description of the next episode. By using $\mathcal{M}(\cdot,\cdot)$, we expect to produce text recaps with better plot contingency with the upcoming story.

In the following sections, we demonstrate in details: (1) the definition of the summarization function $\mathcal{S}(\cdot,\cdot)$; (2) two factors that derive the contingency function $\mathcal{M}(\cdot,\cdot)$ based on different hypotheses.

4.1 Plot Summarization

In this section, we discuss the summarization component of our model's objective function. Our model is inspired by the coverage-based summarization (Lin and Bilmes, 2010), whose key idea is to find a proxy that approximates the information overlap between the summary and the original document. In this work, any text is assumed to be represented by a set of "*concepts*" using weights to distinguish their importance. To be more specific, a *concept* is defined as a noun/verb/adjective or noun/verb phrase. In terms of *concepts*, we define the summarization term $\mathcal{S}(R^i, D^i)$ as follows:

$$\mathcal{S}(R^i, D^i) = \sum_{c \in \mathcal{C}(D^i)} z(c, D^i) \max_{r \in R^i} w(c, r), \tag{2}$$

where $\mathcal{C}(D^i) = \{c' | c' \in d^i_j, \forall d^i_j \in D^i\}$ is the concept set of D^i, and $z(c, D^i)$ measures the importance of c in D^i. We use Term Frequency Inverse

Document Frequency (TF-IDF) (Salton and Buckley, 1988) to calculate $z(c, D^i)$. Finally, $w(c, r)$ denotes the relatedness of a concept c to a sentence r.

We use Word2Vec (Mikolov et al., 2013) vectors as the semantic representation of concepts, and define $w(c, r)$ as:

$$w(c, r) = |c| \cdot \max_{c' \in r} \cos(\mathbf{c}, \mathbf{c}'), \tag{3}$$

where bold notations are the Word2Vec representations of c and c'. Note that if c is a phrase, $\mathbf{c}$ is mean pooled by the embeddings of its component words. $|c|$ is the number of words in c.

4.2 Plot Contingency

We model plot contingency on the concept level as well as on the sentence level. Therefore, the component $\mathcal{M}(\cdot, \cdot)$ is decomposed into two factors:

$$\begin{aligned} \mathcal{M}(R^i, D^{i+1}) =& \lambda_s \mathcal{M}_s(R^i, D^{i+1}) + \\ & \lambda_r \mathcal{M}_r(R^i, D^{i+1}). \end{aligned} \tag{4}$$

where $\mathcal{M}_s(R^i, D^{i+1})$ measures how well R^i can summarize the next episode D^{i+1} and $\mathcal{M}_r(R^i, D^{i+1})$ is the factor that quantify the ability of R^i to reconstruct D^{i+1}. $\lambda_s, \lambda_r \geq 0$ are coefficients for $\mathcal{M}_s(\cdot, \cdot)$ and $\mathcal{M}_r(\cdot, \cdot)$ respectively. In the following sections, we define and explain these two factors in details.

4.2.1 Concept Coverage

Following the coverage assumption of Section 4.1, we argue that the text recap should also cover important concepts from the next episode. Therefore, the first contingency factor can be defined in the same form as the summarization component where D^i's in Equation 2 are replaced by D^{i+1}'s:

$$\mathcal{M}_s(R^i, D^{i+1}) = \sum_{c \in \mathcal{C}(D^{i+1})} z(c, D^{i+1}) \max_{r \in R^i} w(c, r). \tag{5}$$

4.2.2 Sparse Reconstruction

As events happening in the current episode can have an impact on the next episode, there exist hidden connections between the descriptive sentences in D^i and D^{i+1}. To be more specific, assuming descriptive sentences from D^{i+1} are dependent on a few sentences in D^i, we aim to infer such hidden contingency. Here we assume that sentence d_j^{i+1} is related to a small number of sentences in D^i.

Let $\boldsymbol{\alpha}_j^{i+1} \in \mathbb{R}^{|D^i|}$ be the indicator that determines which sentences in D^i prompt d_j^{i+1}, and $\mathbf{W}$ be the matrix that transforms these contingent sentences to the embedding space of d_j^{i+1}. Intuitively, our model learns $\mathbf{W}$ by assuming each sentence in D^{i+1} can be reconstructed by contingent sentences from D^i:

$$\mathbf{d}_j^{i+1} \approx \mathbf{W}\mathbf{D}^i \boldsymbol{\alpha}_j^{i+1}, \tag{6}$$

In the equation, we first convert every description sentence to its distributed representation using the pre-trained skip-thought model proposed by Kiros et al. (2015). The sentence embedding is denoted in bold (e.g. $\mathbf{d}_j^i$ for sentence d_j^i). $\mathbf{D}^i = [\mathbf{d}_1^i; \cdots; \mathbf{d}_{|D^i|}^i]$ stacks the vector representations of all sentences in D^i, and $\boldsymbol{\alpha}_j^{i+1}$ linearly combines the contingent sentences.

We propose to jointly optimize $\boldsymbol{\alpha}_j^{i+1}$ and $\mathbf{W}$ by:

$$\begin{aligned} \min_{\{\boldsymbol{\alpha}^{i+1}\}_{i=1}^{E-1}, \mathbf{W}} \sum_{i,j} \big(& \|\mathbf{W}\mathbf{D}^i \boldsymbol{\alpha}_j^{i+1} - \mathbf{d}_j^{i+1}\|_2^2 \\ & + \gamma \|\boldsymbol{\alpha}_j^{i+1}\|_1 \big) + \theta \|\mathbf{W}\|_F^2, \end{aligned} \tag{7}$$

where we denote $\boldsymbol{\alpha}^{i+1} = [\boldsymbol{\alpha}_1^{i+1}; \cdots; \boldsymbol{\alpha}_{|D^{i+1}|}^{i+1}]$. We impose sparsity constraint on $\boldsymbol{\alpha}_j^{i+1}$ with L_1 norm such that only a small fraction of sentences in D^i will be linked to d_j^{i+1}. γ and θ are coefficients of the regularization terms.

Given the optimal $\mathbf{W}^*$ from Equation 7, our main objective is to identify the subset of descriptions in D^i that best capture the contingency between D^i and D^{i+1}. The reconstruction contingency factor can be defined as:

$$\mathcal{M}_r(R^i, D^{i+1}) = \sum_{d \in D^{i+1}} \max_{r \in R^i} \mathbf{r}^\intercal \mathbf{W}^* \mathbf{d}. \tag{8}$$

4.3 Optimization

In this section, we describe our approach to optimize the main objective function expressed in Equations 1 and 7.

Finding an efficient algorithm to optimize a set function like Equation 1 is often challenging. However, it can be easily shown that the objective function of Equation 1 is submodular, since all its components $\mathcal{S}(R^i, D^i), \mathcal{M}_s(R^i, D^{i+1})$ and

$\mathcal{M}_r(R^i, D^{i+1})$ are submodular with respect to R^i. According to Lin and Bilmes (2011), there exists a simple greedy algorithm for monotonic submodular function maximization where the solution is guaranteed to be close to the real optimum. Specifically, if we denote R^i_{greedy} as the approximation optimized by greedy algorithm and R^{i*} as the best possible solution, then $\mathcal{F}(R^i_{greedy}) \geq (1 - \frac{1}{e}) \cdot \mathcal{F}(R^{i*})$, where $\mathcal{F}(\cdot)$ is the objective function of Equation 1 and $e \approx 2.718$ denotes the natural constant. The greedy approach is shown in Algorithm 1.

Algorithm 1 Text Recap Extraction

```
Input: Vectorized sentence representations $\{\mathbf{D}^i\}_{i=1}^E$, parameters $\lambda_s, \lambda_r, \theta, \gamma$, budget $K$,
    optimal $\mathbf{W}^*$ for Equation 7.
Output: Text recaps $\{R^i\}_{i=1}^E$.
1: for $i = 1, \cdots, E$
2:     Initialize $R^i \leftarrow \emptyset$;
3:     REPEAT
4:         $r^* \leftarrow \arg\max \mathcal{F}(R^i \cup \{r\})$;
5:         $R^i \leftarrow R^i \cup \{r^*\}$;
6:     UNTIL $|R^i| \geq K$
7: end
```

Algorithm 1 requires the optimal $\mathbf{W}^*$ learned from the adjacent episode pairs in Equation 7. We utilize the algorithm that iteratively updates $\mathbf{W}$ and $\boldsymbol{\alpha}$ given the current solution. At each iteration, each variable ($\mathbf{W}$ or $\{\boldsymbol{\alpha}^{i+1}\}$) is updated by fixing the other. At t-th iteration, $\mathbf{W}^{(t)}$ is computed as the solution of ridge regression (Hoerl and Kennard, 1970):

$$\mathbf{W}^{(t)} = \mathbf{D}\mathbf{X}^\intercal(\mathbf{X}\mathbf{X}^\intercal + \theta\mathbf{I})^{-1}, \qquad (9)$$

where $\mathbf{D}$ and $\mathbf{X}$ stack all $\mathbf{d}_j^{i+1}$ and $\mathbf{x}_j^{i+1} \triangleq \mathbf{D}^i\boldsymbol{\alpha}_j^{i+1}, \forall i = 1, \cdots, E-1, j = 1, \cdots, |D^i|$. Fixing $\mathbf{W}$, each $\boldsymbol{\alpha}_j^{i+1}$ can be solved separately by general sparse coding algorithms as stated in Mairal et al. (2009). Algorithm 2 shows the optimization process of Equation 7.

5 Multi-View Recap Extraction

In addition to plot descriptions, there are also dialogues and plot synopses available for TV shows. Descriptions, dialogues and synopses can be seen as three different views of the same TV show episode.

Algorithm 2 Reconstruction Matrix Optimization

```
Input: Vectorized sentence representations $\{\mathbf{D}^i\}_{i=1}^E$, $\theta$ and $\gamma$.
Output: Contingency matrix $\mathbf{W}$.
1: Initialize $\mathbf{W}^{(0)} \leftarrow \mathbf{I}$ and $\boldsymbol{\alpha}_j^{i+1(0)} \leftarrow \mathbf{0}, \forall i, j$;
2: Initialize iteration step $t \leftarrow 0$;
3: REPEAT
4:     $t \leftarrow t + 1$;
5:     $\mathbf{W}^{(t)}$ is updated according to Equation 9;
6:     $\forall i, j, \boldsymbol{\alpha}_j^{i+1,(t)} \leftarrow \text{sparse\_coding}(\mathbf{W}^{(t)})$;
7: UNTIL $\|\mathbf{W}^{(t)} - \mathbf{W}^{(t-1)}\|_F^2 \leq \epsilon$
```

Previously, we build TREM using plot descriptions. In this section, we expand our TREM model to incorporate plot synopses and dialogues. We define text synopses and dialogues as $S = \{S^1, \cdots, S^E\}$ and $T = \{T^1, \cdots, T^E\}$, where S^i and T^i are the set of sentences from synopses and dialogues of the i-th episode.

Dialogues In TV shows, a lot of useful information is presented via actors' dialogues which motivates us to extend our TREM model to include dialogues. Both views can be used to identify recap segments which are assumed to be summative and contingent. Denote the neighboring dialogues of R^i as $N(R^i) = \{t \in T^i \big| \exists r \in R^i, \text{ s.t. } |\text{time}(t) - \text{time}(r)| < \delta\}$, we extend the optimization objective (Equation 1) into:

$$\begin{aligned}\mathcal{F}(R^i) = &\big(\mathcal{S}(R^i, D^i) + \mathcal{S}(N(R^i), T^i)\big) \\ &+ \big(\mathcal{M}(R^i, D^{i+1}) + \mathcal{M}(N(R^i), T^{i+1})\big).\end{aligned} \qquad (10)$$

Synopses Since a synopsis is a concise summary of each episode, we can treat plot summarization as text alignment where R^i is assumed to match the content of S^i. Therefore, the summarization term can be redefined by substituting D^i with S^i:

$$\mathcal{S}(R^i, S^i) = \sum_{c \in \mathcal{C}(S^i)} z(c, S^i) \max_{r \in R^i} w(c, r). \qquad (11)$$

Similarly, the contingency component can be modified to include connections from synopses to detailed descriptions. For Equation 8, we substitute

	ROUGE-1	ROUGE-2	ROUGE-SU4
ILP-Ext (Banerjee et al., 2015)	0.308	0.112	0.091
ILP-Abs (Banerjee et al., 2015)	0.361	0.158	0.120
Our approach TREM	**0.405**	**0.207**	**0.148**
w/o SR	0.393	0.189	0.144
w/o CC	0.383	0.171	0.132
w/o SR&CC (summarization only)	0.374	0.168	0.129

Table 2: Experimental results on different methods using descriptions. Contingency-based methods generally outperforms summarization-based methods.

D^{i+1} to S^{i+1} where our model only focuses on high-level storyline:

$$\mathcal{M}_r(R^i, S^{i+1}) = \sum_{s \in S^{i+1}} \max_{r \in R^i} \mathbf{r}^\intercal \mathbf{W}^* \mathbf{s}. \tag{12}$$

6 Experimental Setup

We designed our experiments to evaluate whether our TREM model, by considering contingency between adjacent episodes, can achieve better results than summarization techniques. Furthermore, we want to examine how each contingency factor as proposed in Section 4.2 contributes to the system performance. As our model can integrate multiple views, we want to dissect the effects of using different combinations of three views.

6.1 Comparison Models

To answer the research questions presented above, we compare the following methods in our experiments.

– **ILP-Ext** and **ILP-Abs** (Banerjee et al., 2015): This summarizer generates sentences by optimizing the integer linear programming problem in which the information content and linguistic quality are defined. Both extractive and abstractive implementations are used in our experiments.

– **TREM**: Our TREM model proposed in Section 4 extracts sentences that can both summarize the current episode and prompt the next episode with two contingency factors.

– **TREM w/o SR**: The TREM model without the sparse reconstruction factor proposed in Section 4.2.2.

– **TREM w/o CC**: The TREM model without the concept coverage factor proposed in Section 4.2.1.

– **TREM w/o SR&CC**: The summarization-only TREM model without contingency factors. In the rest of the paper, we also call it as **TREM-Summ**.

– **Multi-view TREM**: The augmented TREM model with descriptions, dialogues and synopses as proposed in Section 5. Different views and combinations will be tested in our experiments.

6.2 Methodology

Using *TVRecap*, we measure the quality of generated sentences following the standard metrics in the summarization community, ROUGE (Lin and Hovy, 2003).

For the purpose of evaluation, we defined a development and a test set, by randomly selecting 18 adjacent pairs of episodes from all seasons. These episodes were selected to have at least two recap description sentences. The remaining 70 episodes were only used during the learning process of $\mathbf{W}$. After tuning hyper-parameters on development set, we report the comparison results on the test set.

7 Results and Discussion

7.1 Overall Results

Table 2 shows our experimental results comparing TREM and baseline models using descriptions.

In general, contingency-based methods (TREM, TREM w/o SR and TREM w/o CC) outperform summarization-based methods. Our contingency assumptions are verified as adding CC and SC both improve TREM with summarization component only. Moreover, the best result is achieved by the complete TREM model with both contingency factors. It suggests that these two factors, modeling word-level summarization and sentence-level reconstruction, are complementary.

From the summarization-based methods, we can see that our TREM-Summ gets higher ROUGE scores than two ILP approaches. Additionally, we

Target sentence from next episode	Sentences with highest reconstruction value from current episode
Kate is putting water bottles in a pack.	We see three bottles of water. They go into a room with a body bag on a gurney. Kate is going through clothes, as Claire approaches.
Locke is with his knife case, holding a pencil, sitting by a fire.	Boone is coming up to camp and sees Locke sitting by a fire. Locke throws a knife into the ground, just out of Boone's reach. Boone quickly cuts through the ropes and starts running.
In another part of the temple grounds, Miles and Hurley are playing Tic-Tac-Toe by placing leaves in a grid of sticks on the ground.	John contemplates the fabric swatches he is holding. On the beach, Frank covers Locke's body with a tarp. Helen closes the door and brings the case inside to the kitchen.

Table 3: A case study on sparse reconstruction as proposed in Section 4.2.2. Sentences in the first column are reconstructed by sentences in the second column. The first two examples successfully captures related sentences, while the third example fails.

note that the performance of ILP-Ext is poor. This is because ILP-Ext tends to output short sentences, while ROUGE is a recall-oriented measurement.

Model	Current	Next	R-1	R-2	R-SU4
TREM-Summ	des	-	0.374	0.168	0.129
	syn	-	0.369	0.163	0.121
	dial	-	0.354	0.138	0.115
	des+syn	-	0.384	0.172	0.132
	des+dial	-	0.386	0.168	0.135
TREM	des	des	0.405	0.207	0.148
	des	syn	**0.411**	**0.219**	**0.154**
	des	dial	0.375	0.158	0.127
	des	des+syn	0.409	0.210	**0.154**
	des	des+dial	0.395	0.177	0.142

Table 4: Comparison of views in summarization-only TREM and full TREM with contingency factors. "*des*", "*syn*", and "*dial*" are abbreviated for description, synopses and dialogues.

7.2 Multi-view Comparison

As shown in Table 4, The second study examines the effect of different views in both types of methods using the TREM model. In single-view summarization, TREM-Summ with descriptions outperforms methods based on the other two views. In terms of hybrid of views, only ROUGE-1 is significantly improved, while ROUGE-2 and ROUGE-SU4, which focus more on semantic consistency, have little improvement.

In contingency-based methods, we keep the current episode represented as descriptions which obtain the best performance in single-view summarization, and change the views of the next episode. Comparing the model using descriptions with the one fusing descriptions and synopses, we can see that simply adding views does not guarantee higher ROUGE scores. In both TREM-Summ and full TREM, dialogue is inferior to others. It might be because dialogues contain too many trivial sentences. Synopses, however, are relatively short, but provide key plots to summarize the story, and hence achieve the best ROUGE scores.

7.3 Qualitative Study on Sparse Reconstruction

In this section, we give some examples to illustrate the process of sparse reconstruction. Equation 7 assumes that each descriptive sentence can be reconstructed by a few sentences from the previous episode. Table 3 shows three examples of sentences with their top-3 reconstructive sentences, which are defined by values in the indicator vector $\boldsymbol{\alpha}_j^{i+1}$.

7.4 Limitations and Future Work

TREM restrains the contingency within adjacent episodes. However, storylines sometimes proceed through multiple episodes. In our model, with more connectivity terms $\mathcal{M}(R^i, D^j)$ where $i < j$, we can develop more general system with longer dependencies.

While our model and dataset are appropriate for text recap extraction and algorithm comparison, this task can be further applied to multimedia settings, where visual or acoustic information can be included. Therefore, in future work, we plan to expand our work to broader applications where interconnectivity between consecutive instances is crucial, such as educational lectures, news series and book chapters. Specifically, TREM can be integrated with video description results to get an end-to-end system that produces video recaps.

8 Conclusion

In this paper, we explore a new problem of text recap extraction for TV shows. We propose an unsupervised model that identifies recap segments from multiple views of textual scripts. To facilitate the study of this new research topic, we create a dataset called *TVRecap*, which we test our approach on. From the experimental results, we conclude that contingency-based methods improve summarization-based methods at ROUGE measurements by exploiting plot connection between adjacent episodes.

Acknowledgement

This material is based in part upon work partially supported by the National Science Foundation (IIS-1523162). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Marta Aparício, Paulo Figueiredo, Francisco Raposo, David Martins de Matos, Ricardo Ribeiro, and Luís Marujo. 2016. Summarization of films and documentaries based on subtitles and scripts. *Pattern Recognition Letters* 73:7–12.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ilp based multi-sentence compression. In *24th International Joint Conference on Artificial Intelligence (IJCAI). Buenos Aires, Argentina: AAAI press*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. *arXiv preprint arXiv:1506.01597* .

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 2625–2634.

Paulo Figueiredo, Marta Aparício, David Martins de Matos, and Ricardo Ribeiro. 2015. Generation of multimedia artifacts: An extractive summarization-based approach. *arXiv preprint arXiv:1508.03170* .

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2712–2719.

Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document summarization based on data reconstruction. In *AAAI*.

Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *EACL*. pages 712–721.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *arXiv preprint arXiv:1506.06726* .

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pages 71–78.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 912–920.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the*

Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, pages 510–520.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*. ACM, pages 689–696.

Kathleen McKeown. 2005. Text summarization: News and beyond. In *Proceedings of the Australasian Language Technology Workshop*. pages 4–4.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Amita Misra, Pranav Anand, JEF Tree, and MA Walker. 2015. Using summarization to discover argument facets in online idealogical dialog. In *NAACL HLT*. pages 430–440.

Francisco Raposo, Ricardo Ribeiro, and David Martins de Matos. 2015. On the application of generic summarization algorithms to music. *IEEE Signal Processing Letters* 22(1):26–30.

Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. 2015. The long-short story of movie description. In *Pattern Recognition*, Springer, pages 209–221.

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 433–440.

Shourya Roy, Ragunathan Mariappan, Sandipan Dandapat, Saurabh Srivastava, Sainyam Galhotra, and Balaji Peddamuthu. 2016. Qa rt: A system for real-time holistic quality assurance for contact center dialogues. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24(5):513–523.

Jitao Sang and Changsheng Xu. 2010. Character-based movie summarization. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, pages 855–858.

Ruben Sipos, Adith Swaminathan, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Temporal corpus summarization using submodular word coverage. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, pages 754–763.

Mark Wasson. 1998. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pages 1364–1368.