A Modular Approach for Multimodal Summarization of TV Shows

Anonymous ACL submission

Abstract

In this paper we address the task of summarizing television shows, which touches key areas in AI research: complex reasoning, multiple modalities, and long narratives. We present a modular approach where separate components perform specialized sub-tasks which we argue affords greater flexibility compared to end-to-end methods. Our modules involve detecting scene boundaries, reordering scenes so as to minimize the number of cuts between different events, converting visual information to text, summarizing the dialogue in each scene, and fusing the scene summaries into a final summary for the entire episode. We also present a new metric, PREFS (**P**recision and **R**ecall **E**valuation of Summary **F**act**s**), to measure both precision and recall of generated summaries, which we decompose into atomic facts. Tested on the recently released SummScreen3D dataset (Papalampidi and Lapata, 2023), our method produces higher quality summaries than comparison models, as measured with ROUGE and our new fact-based metric.

1 Introduction

In this paper, we address the challenging task of summarizing television shows which has practical utility in allowing viewers to quickly recall plot points, characters, and events without the need to re-watch entire episodes or seasons. From a computational standpoint, the task serves as a testbed for complex reasoning over long narratives, involving multiple modalities, non-trivial temporal dependencies, inferences over events, and multi-party dialogue with different styles. An added difficulty concerns assessing the quality of generated summaries for long narratives, whether evaluations are conducted by humans or via automatic metrics.

Most prior work on creative summarization does not consider the above challenges all at once, focusing either on the text modality and full-length narratives with complex semantics (Gorinski and Lapata, 2015; Chen et al., 2022; Agarwal et al., 2022) or on short video clips which last only a couple of minutes (Tapaswi et al., 2016; Lei et al., 2018; Liu et al., 2020). A notable exception is Papalampidi and Lapata (2023), who incorporate multimodal information into a pre-trained textual summarizer by adding (and tuning) adapter layers (Rebuffi et al., 2017; Houlsby et al., 2019). On the evaluation front, there is no single agreed-upon metric for measuring summary quality automatically, although there is mounting evidence that ROUGE (Lin, 2004) does not discriminate between different types of errors, in particular those relating to factuality (Min et al., 2023; Clark et al., 2023).

While end-to-end models are a popular choice for summarization tasks (Chen et al., 2022; Zhang et al., 2020), more modular approaches have been gaining ground recently (Guan and Padmakumar, 2023; Gupta and Kembhavi, 2022; Sun et al., 2023) for several reasons. Modules can be developed independently, and exchanged for better versions if available, new modules can be added to create new solutions or repurposed for different tasks, and dependencies between modules can be rearranged. Aside from greater controllability, modular approaches are by design more interpretable, since errors can be inspected and attributed to specific components. In this paper we delegate the end-to-end task of summarizing from multiple modalities (i.e., TV show video and its transcript) to more specialized modules, each responsible for handling different subtasks and their interactions. Our approach is depicted graphically in Figure 1.

As scene breaks are not always given explicitly, we devise an algorithm to identify them from the order of the speaker names (row 1, Figure 1). Additionally, we select the optimal order in which to re-arrange scenes (row 2, Figure 1), as these often appear in non-linear order (e.g., there can be several subplots or flashbacks). Next, we produce summaries in a two-layer process. A vision-processing

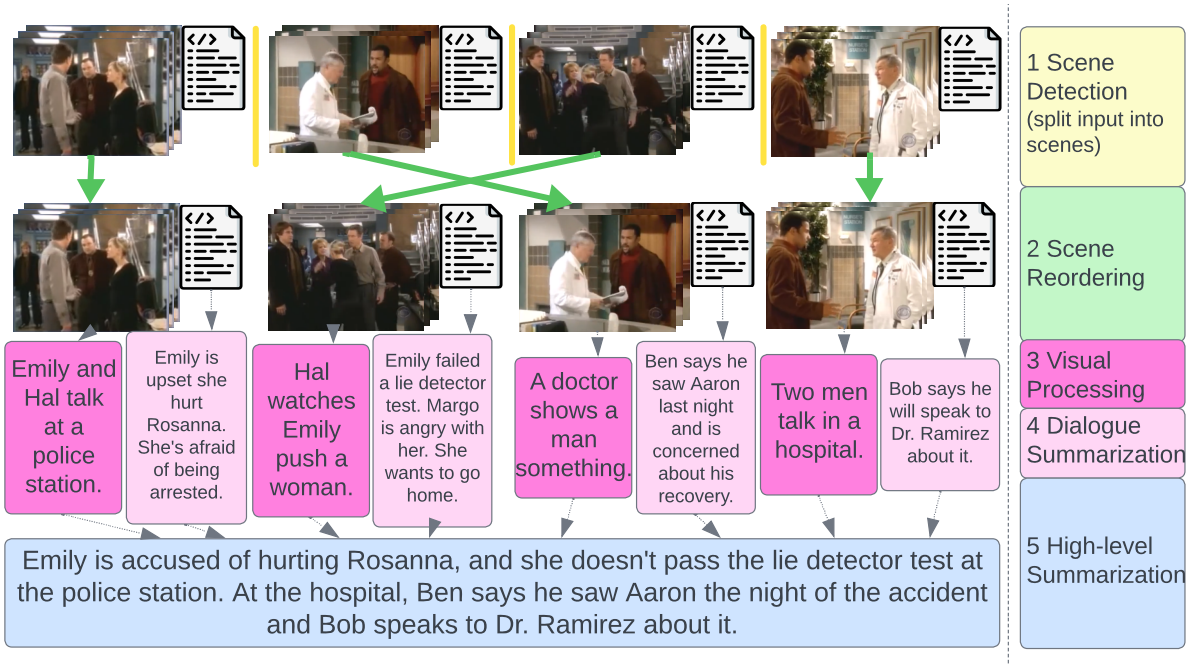

Figure 1: Graphical depiction of our approach for long-form multimodal summarization where different subtasks are performed by five, specialized modules (shown in different colors). We use simplified summaries for display and show only four scenes. This full episode (*As the World Turns* aired 01-06-05, contains 29 scenes.

module (Lei et al., 2020; Lin et al., 2022) converts the video to text using visual captioning (row 3, Figure 1), which leverages the strong specialized ability of vision-to-text models and allows us to treat the problem as one of text-to-text summarization. We also summarize each scene independently with a module specialized for dialogue summarization (row 4, Figure 1). Finally, we use a high-level summarization module specialized for narrative summarization to fuse the sequence of scene summaries into a final summary (row 5, Figure 1).

We also propose a new metric for assessing the factuality of generated summaries by adapting FActScore (Min et al., 2023), a recently introduced metric for detecting hallucination in text generation. We break the generated summary into atomic facts, and check what fraction of them are supported by the reference. This we term fact-precision. We also do the same in reverse, breaking the reference into facts and measuring what fraction are supported by the generated summary, which we term fact-recall. Our metric, PREFS (**P**recision and **R**ecall **E**valuation of Summary **F**act**s**), is the harmonic average of these two scores. Our contributions are:

- We present a novel modular approach to multimodal summarization, where separate subtasks are performed by separate modules;
- Our modules involve detecting scene breaks, reordering each scene, summarizing the dialogue therein, converting the visual information to text, and fusing the scene-summaries into a final summary (see Figure 1);
- We present two novel algorithms, for determining the optimal order in which to place each scene, and for identifying where the scene breaks are located in the transcript;
- We devise a new metric for summarization, based on splitting text into atomic facts, which captures both precision and recall and correlates significantly with human judgments.

2 Related Work

In the area of long-form summarization, various methods have been proposed to deal with inputs that exceed the context size of a large language model (LLM). Memwalker (Chen et al., 2023) forms a tree of hierarchical summary nodes and traverses it during inference to find the most relevant parts of the input text. Pang et al. (2023) propose a two-layer method where the top layer captures coarse long-range information and produces top-down corrections to a local attention mechanism in the lower layer. Chang et al. (2023) describe two settings for long-form summarization with LLMs: hierarchical merging summarizes chunks of the input sequentially, whereas iterative updating con-

tinually updates a single summary for each of the chunks. We also adopt a two-layer approach but differ in that we divide the input into semantically meaningful chunks, i.e., scenes, rather than uniform chunks, and we summarize each independently before fusing them together.

The problem of generating descriptions for videos has also received much attention, largely independently from long-document summarization. It is common practice (Zhang et al., 2021; Pan et al., 2020; Ye et al., 2022) to extract features from each frame individually, then fuse them into a single feature vector and decode with a language model. Swinbert (Lin et al., 2022) instead uses an end-to-end video network, dispensing with image-based encoders, and samples frames densely. Lei et al. (2020) generate descriptions for short videos with a memory-augmented vision transformer. Popular video captioning datasets (Chen and Dolan, 2011; Xu et al., 2016) are only ∼10s in length. YouCook (Zhou et al., 2018) is a recent dataset with longer videos (5min on average), but still far shorter than the TV shows we focus on here.

Multimodal summarization is an extension of video captioning in which the input contains text as well as video. Pan et al. (2023) tackle the analogous problem for still-images with a single model whose architecture allows image and text input and can produce a text description as output. Bhattacharyya et al. (2023) convert the video to text and feed it to a visual storytelling model. Tsimpoukelli et al. (2021) train a vision encoder to produce a sequence of vectors which, when fed to an LLM, produce a textual description of the image contents. Papalampidi and Lapata (2023) apply a similar idea to multimodal summarization, extracting a feature vector from the visual input which is fed, along with the token embeddings from the transcript, to a summarization network. Our method does not try to extract a visual feature vector that functions like a token embedding, but rather extracts actual tokens, i.e., a textual description, from the video.

3 Decomposition of Multimodal Summarization into Modules

Our decomposition of the summarization task into modules is motivated by three assumptions specific to TV shows: (1) each scene is somewhat self-contained and, on a coarse level, the events it depicts can be understood independently of other scenes (2) the order in which scenes appear is not necessarily the optimal order to facilitate understanding, often shows cut back and forth between different plotlines, and sometimes they are presented non-linearly (3) an effective way to capture visual information in a multimodal text summary is to translate it into natural language. These assumptions motivate our modular approach, which is depicted graphically in Figure 1. Five separate subtasks are performed by separate components: scene-break detection (top row), scene reordering (second row), converting visual information to text (third row), dialogue summarization (fourth row), and high-level summarization (fifth row).

Assumption (1) motivates our choice to detect scene breaks and summarize each independently with a module specialized for dialogue summarization and then later fuse these with a high-level summarization module to produce a final output summary. Assumption (2) motivates our scene-reordering algorithm: we do not simply concatenate scene-level summaries in the order in which they appear, but rather we compute an optimal order designed to minimize the number of transitions between different plotlines. Assumption (3) motivates our choice of how to capture the visual information in our summaries. A visual processing module produces a textual description of the video for each scene, which is fed, alongside the dialogue summaries to the high-level summarization module. As a result, the high-level and dialogue summarization modules only need to focus on the single modality of text.

3.1 The Multimodal Summarization Task

Before discussing the details of the various modules, we provide information on our specific task and the dataset we are working with. We develop and evaluate our approach on SummScreen3D (Papalampidi and Lapata, 2023)[1], which to our knowledge is the only existing dataset for long-form video summarization. It consists of 5,421 videos of TV episodes (mostly soap operas) varying in length from 30–60min, with accompanying transcripts (on average 6K tokens long) and summaries that were written by fans and scraped from public websites (average length is 200 tokens). These are partitioned into 296 each for validation and testing, and the remaining 4,829 for training. Videos can have multiple summaries from different fan-

[1] https://github.com/ppapalampidi/video_abstractive_summarization

sites (the average number of summaries per episode is 1.53), giving a total of 8,880 training pairs.

In SummScreen3D, each data point contains a video, a transcript that includes character names and, sometimes, marked scene breaks, and a closed captions file, which is used to display subtitles and has timestamps but not speaker names. We align the lines in the transcript with those from the closed captions. These do not match perfectly, because of slight errors in the automatic transcription. For each line t in the transcript and utterance c in the caption, we estimate a similarity score as $\frac{|f(t,c)|}{\min(|t|,|c|)}$, where f computes longest common subsequence. Then we use dynamic time warping to align both sequences (Myers and Rabiner, 1981; Papalampidi et al., 2021). As a result, we obtain alignments of transcript utterances to video segments, which allows us to use the scene-breaks from the former to segment the latter.

3.2 Scene Detection

This module (Figure 1, row 1) partitions the transcript into contiguous chunks. Each line in a transcript begins with the name of the character speaking, and our algorithm seeks a partition where each chunk contains only a small number of characters.

We define a cost for a given partition, by invoking the minimum description length (MDL) principle (Grünwald, 2007), in which the optimal representation of a piece of data is that which contains the fewest of bits. Thus, the cost of a partition is the number of bits needed to specify it. We assume the total set of N character names for the entire transcript is given. Then, for each scene, we make a scene-specific codebook for the n characters that appear there, which assigns each character a code as an index from $0, \ldots, n-1$, which all have length $\leq \lceil \log n \rceil$Using exactly $\lceil \log n \rceil$ guarantees a prefix-free code (Grunwald, 1998)[2]. The number of possible codebooks with n out of N characters is $\binom{N}{n}$. Imposing e.g., lexicographic order, on these, the number of bits to specify one is

$$C(N,n) := \log \binom{N}{n} = \log \frac{N!}{n!(N-n)!}$$

and the total cost of a given scene is then

$$C(N,n) + l \log n\,, \qquad (1)$$

where N is the total number of speakers as before, and n and l are, respectively, the number of distinct speakers and the number of transcript lines in the scene. See Appendix D for details and examples.

[2]Because we do not need to convert to concrete codes, we omit the ceiling operator in our optimization objective.

Algorithm 1 Compute the optimal partition of transcript lines into scenes.

```
m ← # lines in the transcript
P ← an m × m matrix, to store the cost of spans
Q ← an m × m matrix, to store the optimal partition of spans
for i = 2, ..., m do
    for j = 1, ..., m − 1 do
        n ← # of unique characters in lines j, ..., j + i
        P[j, j + i] ← C(N, n) + i log n
        Q[j, j + i] ← ∅
        for k = j, ..., j + i do
            if P[j, k] + P[k, j + i] < P[j, j + i] then
                P[j, j + i] ← P[j, k] + P[k, j + i]
                Q[j, j + i] ← Q[j, k] ∪ {k} ∪ Q[k, j + i]
```

Let m be the number of lines in the transcript. Then, for any $1 \leq i < j \leq m$, the cost of all scenes from line i to line j under the optimal partition, written as $S(i,j)$, can be expressed recursively as

$$S(i,j) = \min_{2 \leq k \leq i} S(i,k) + S(k,j)\,,$$

where $S(i,j)$ is defined as zero when $i = j$. This motivates a dynamic programming solution, similar to the CYK algorithm for context-free parsing (Kasami, 1966), in which we compute, in order, the optimal solution of spans of lengths $2, \ldots m$, and reuse solutions of smaller spans when computing those of larger spans. Our algorithm runs in $O(N^3)$, with only $O(N^2)$ calls to compute the scene cost as per Equation (1). The scene breaks can be transferred to the video because of the dynamic time-warping alignment between the transcript and the timestamps in the closed captions, as described in Section 3.1. Note, we do not have to specify the number of scenes, that is determined automatically by our algorithm.

3.3 Scene Reordering

We now discuss how we compute the optimal order of scenes (Figure 1, row 2), prior to summarization. We first define a cost for a given order as

$$C(s_1, \ldots, s_n) = \sum_{i=1}^{n-1} 1 - \mathrm{IOU}(s_i, s_{i+1})\,, \qquad (2)$$

where IOU is the intersection over union of character names. For example, if the characters in scene s_1 are Alice and Bob, and the characters in scene s_2 are Bob and Charlie, then $\mathrm{IOU}(s_1, s_2) = \frac{1}{3}$. Additionally, we introduce a

Algorithm 2 Minimize the number of transitions between scenes with different speakers.

function IOU(x,y)
  return $\frac{\text{\#characters in both x and y}}{\text{\#characters in x+\#characters in y}}$
function COMPUTE OPTIMAL ORDER(S)
  $s_1, \ldots, s_n \leftarrow$ scenes in order of appearance in S;
  $C \leftarrow n \times n$ matrix; ▷ cache for IOUs
  for $i = 1, \ldots, n$ **do**
    for j=i,...,n **do**
      $C[i, j] \leftarrow 1 - \text{IOU}(s_i, s_j)$;
      if $(s_i, s_j) == 0$ **then**
        $C[j, i] = 1$
      else
        $C[j, i] = \infty$
  $changed \leftarrow True$;
  while $changed$ **do**
    for $i = 2, \ldots, 2$ **do**
      $n = \min\{j | C[j, i] < 1\} + 1$ ▷ new idx for i
      $cost1 \leftarrow C[i-1, i+1] - C[i-1, i] - C[i, i+1]$;
      $cost2 \leftarrow C[n-1, i] + C[i, n] - C[n-1, n]$;
      if $cost1 + cost2 < 0$ **then**
        move s_i to position new in S;
        $changed \leftarrow True$
        break

constraint that if the same character appears in two different scenes, s_i and s_j, then we should never swap the order of s_i and s_j, as that may violate causality between the two scenes.

We approximately solve this optimization problem by passing from s_2 to s_n and moving each scene as far to the front as possible without violating the causality constraint, if this leads to an improved total order cost. We continue passing from s_2 to s_n until no changes are made. Our algorithm runs in $O(N^2)$, where N is the number of scenes, typically 30. The change in cost when moving a scene depends only on the IOU cost of that scene and its current and future neighbours, and the IOU cost of all pairs of scenes can be cached.

3.4 Vision Processing

We explore two methods for the vision-processing module which differ in architecture and scope (Figure 1, row 3). SwinBERT (Lin et al., 2022) is a dedicated video captioning model; it operates directly on a sequence of video frames for which it generates a description. SwinBERT is end-to-end trained with a masked language modeling objective, and a sparse attention mask regularizer for improving long-range video sequence modeling. We apply SwinBERT to the video for each scene, sampled at 3 frames-per-second, to obtain a video caption, which we use as a description of video contents.

Kosmos-2 (Peng et al., 2023) is pretrained on several multimodal corpora as well as grounded image-text pairs (spans from the text are associated with image regions) and instruction-tuned on various vision-language instruction datasets. Unlike SwinBERT, it operates on individual images, so we first extract three equally spaced I-frames from the h264 video encoding (Wiegand et al., 2003), and take the captions from each. We prompt Kosmos-2 with "A scene from a TV show in which".

Both SwinBERT and Kosmos-2 can generate uninformative textual descriptions, e.g., "a man is talking to another man" which we discard. We also modify them to replace unnamed entities such as 'the man' with character names where these can be easily inferred. See Appendix B for details of our filtering and renaming procedures.

3.5 Summary Generation

Because of our two-layer summarization approach, we can use a relatively small backbone model, and are still able to encode very long input. In our experiments, we use variants of BART-large (Lewis et al., 2020) for both the dialogue summarization and high-level summarization modules, but any other similar model could be used instead. For the dialogue summarization module (Figure 1, row 4), we use the public Huggingface checkpoint for BART which has been fine-tuned on the SamSum dataset (Gliwa et al., 2019), to output multi-sentence summaries of dialogue. For the high-level summarization module (Figure 1, row 5), we use BART, fine-tuned first for document summarization on the CNN/Daily Mail dataset (Hermann et al., 2015), and then on SummScreen3D. The input for the latter fine-tuning is our re-ordered scene summaries and visual text descriptions, i.e., the output of the dialogue summarization, vision-processing and reordering modules. The output is the gold summaries from the SummScreen3D (Papalampidi and Lapata, 2023) training set. Training and inference took place on a single NVIDIA A100-SXM-80GB GPU, taking seven and one hour(s), respectively.

4 Fact-Precision, Fact-Recall and PREFS

Hallucinations are a widely known issue with abstractive summarization (Song et al., 2018; Maynez et al., 2020; Kryscinski et al., 2020; Gabriel et al., 2021), especially when the output is long (Min et al., 2023) and information is being consolidated from multiple sources (Lebanoff et al., 2019). Automated metrics are crucial for our task and related creative summarization applications where human

evaluation is extremely labor-intensive (e.g., watching long videos or reading book-length transcripts), costly, and difficult to design (Krishna et al., 2023).

Our proposal is based on a recent metric, FActScore (Min et al., 2023), which aims to detect hallucination in text generation. FActScore first parses the generated text into atomic facts (i.e., short sentences conveying one piece of information), and then determines whether these are *supported* by an external knowledge source, such as Wikipedia. The FActScore for some model output is the fraction of the extracted facts that are judged to be supported. Min et al. (2023) recommend using InstructGPT for the first stage of converting the text into facts, and Llama-7B (Touvron et al., 2023) for checking whether these facts are supported (i.e., by zero-shot prompting Llama to estimate whether a generated fact is `True` or `False`).[3]

We adapt this for summary evaluation by replacing the external knowledge source with the gold summaries. Note that this incorporates relevance as well as accuracy. There are many true facts about the TV show being summarized, and only some of them appear in the gold summaries. We assume that facts that do not appear in the gold summary are irrelevant and so should not be marked as correct if they appear in our model summaries. This is our fact-precision metric. For fact-recall, we do the same *in reverse*: we convert the gold summary into atomic facts, and then check whether each of these is supported by the generated summary. Again, the score is the fraction of such facts that are supported. A summary will get a high fact-precision score if every atomic piece of information is both true and relevant enough to appear in the gold summary. It will get a high fact-recall score if it contains every atomic piece of information that was contained in the gold summary. The final score for our metric, which we term PREFS (**P**recision and **R**ecall **E**valuation of Summary **F**act**s**), is the harmonic mean of fact-precision FP and fact-recall FR:

$$\text{PREFS} = \frac{2}{\frac{1}{FP} + \frac{1}{FR}}.$$

In our implementation, we used GPT4-Turbo for the extraction of atomic facts *and* the estimation of whether they are supported, as we found that Llama-7B overestimates the number of supported facts. We use the same prompts as Min et al. (2023) and make a separate query, with in-context learning examples, for each fact. In order to penalize repetitive/redundant information, repeated facts are regarded as unsupported. In the case that GPT indicates that the sentence is not properly formed, we convert it to a single fact which is scored as unsupported (see Appendix C for an example).

We examined whether PREFS correlates with human judgments of factuality. Two annotators, both English native speakers, were asked to watch four randomly selected shows from the SummScreen3D development set. They were then shown facts generated by GPT4 corresponding to summaries produced by several automatic systems (those described in Section 5.1). For each fact they were asked to decide whether it was supported by the episode (1/`True` or 0/`False`). Human ratings significantly correlate with GPT4's estimate on whether a fact is supported (Pearson's $r = 0.5$, $N = 520$, $p < 0.01$), with an inter-annotator agreement of $r = 0.57$ ($p < 0.01$) as upper bound.

5 Experimental Evaluation

5.1 Comparison Models

Our full model was composed of the modules described in Section 3. The scene detection module was employed on SummScreen3D transcripts without explicitly marked breaks (approximately 20% of the time). The high-level summarization module was trained for a max of 10 epochs, with early stopping using ROUGE on the validation set, with a patience of 2. The optimizer was AdamW with learning rate 1e-6, and a linear scheduler with 0 warmup steps. The other models in our framework are not fine-tuned.

We compare our approach to the end-to-end model of Papalampidi and Lapata (2023), which, to our knowledge, represents the state of the art on our task, and various text-only models which operate on the transcript:

Unilimiformer (Bertsch et al., 2023) is a retrieval-based method that is particularly suited to processing long inputs. It wraps around any transformer, with context size k, and can extend this size arbitrarily by storing an index of all input tokens and replacing the transformer query mechanism with the k-nearest neighbours from this index.

LLama-7B, Mistral-7B As our backbone BART models are relatively small, ∼400M parameters, we may be able to obtain better summaries, simply using a larger model; we test this hypothesis

[3] https://github.com/shmsw25/FActScore

	r1	r2	rlsum	fact-prec	fact-rec	PREFS
mistral-7b	28.38 (0.16)	4.82 (0.06)	26.14 (0.17)	31.75	38.00	34.59
llama-7b	16.31 (3.31)	1.99 (0.75)	14.40 (2.96)	16.10	26.60	18.84
central	40.42 (0.27)	9.04 (0.29)	38.13 (0.27)	34.57	31.41	32.91
startend	32.53 (0.38)	6.37 (0.20)	31.80 (0.35)	34.78	33.62	34.19
unlimiformer	42.24 (0.42)	10.32 (0.34)	40.40 (0.42)	37.31	*47.69*	41.87
adapter-e2e	34.13 (0.07)	4.28 (0.08)	31.81 (0.07)	31.15	39.72	34.93
modular-swinbert	**44.89 (0.64)**	*11.39 (0.22)*	*42.92 (0.59)*	**42.71**	46.47	*44.51*
modular-kosmos	*44.86 (0.60)*	**11.83 (0.15)**	**42.97 (0.56)**	*42.29*	**48.54**	**45.20**
upper-bound	42.62	9.87	40.03	71.07	91.25	79.91

Table 1: Automatic evaluation results on SummScreen3D. The first block presents text-only models and the second one multimodal ones. Best results are in **bold**, second-best in *italics*. For ROUGE, we report the mean of five independent random seeds, with standard deviation in parentheses.

with LLama-7B and Mistral-7B, both fine-tuned for three epochs on our training set.

startend We implemented a simple baseline which uses BART fine-tuned for dialogue summarization and takes scenes from the start and end of the transcript up to the maximum that can fit in the context size (1,024). This is inspired by recent work showing that even long-context transformers take information mostly from the beginning and end of the input (Liu et al., 2023).

central is inspired by the method of Papalampidi et al. (2021). Again it uses BART and selects a subset of the input to fit in the context size. It computes a weighted graph whose nodes are scenes and edge-weights are tf-idf similarity scores, then uses the page rank algorithm to rank scenes in order of importance, and selects the topmost important scenes that can fit in the context window.

We also compare to an upper-bound which tests the gold summaries against each other. Recall that SummScreen3D often contains multiple summaries for each episode. We select one from the three websites with the most uniformly-sized summaries and treat it as if it was the predicted summary, then test it against the remaining summaries.

5.2 Results

Table 1 summarizes our results, as measured by ROUGE-1 (r1), ROUGE-2 (r2), ROUGE-Lsum (rlsum) (computed using the python-rouge package) and our new PREFS metric. We present results with BERTScore (Zhang et al., 2019) in Appendix E, for the sake of brevity. Results for Papalampidi and Lapata (2023), which we denote as 'adapter-e2e', were reproduced with their code.[4] For PREFS, as described in Section 4, we use a separate query to GPT for each fact to check whether it is supported. There are roughly 70 facts per generated summary, which can lead to significant financial cost if used excessively. Therefore, while we report five random seeds for ROUGE, we select a single seed (randomly) for fact-precision and fact-recall. Example output is shown in Appendix A.

Our modular approach outperforms comparison models across all metrics. As shown in Table 1, our method, which we denote as 'modular' significantly outperforms the previous end-to-end multimodal system of Papalampidi and Lapata (2023), and all comparison text-only models (upper block). A two-sampled t-test shows that improvement over the comparison models is significant at $\alpha = 0.01$ (calculation in Appendix F). We observe that billion-parameter models like Mistral and Llama struggle with this task (although Mistral is superior), despite being fined-tuned on SummScreen3D. Amongst text-only models, Unlimiformer performs best which suggests that the ability to selectively process long context has a greater impact on the summarization task. As far as our model is concerned, we find that the type of visual processing module has an effect, allbeit small, on the quality of the output sumaries. Overall, Kosmos-2 (Peng et al., 2023) has a slight advantage over SwinBERT (Lin et al., 2022) which we attribute to it having been trained on various image understanding tasks, besides captioning.

[4] https://github.com/ppapalampidi/video_abstractive_summarization

	r1	r2	rlsum	fact-prec	fact-rec	PREFS
w/o transcript	30.71 (0.38)	5.48 (0.15)	28.64 (0.25)	16.53	21.91	18.84
w/o video	43.80 (0.76)	10.91 (0.15)	41.67 (0.72)	40.45	41.47	40.95
w/o reordering	**44.98 (0.52)**	11.61 (0.13)	**42.97 (0.50)**	39.82	45.17	42.33
w/o scene-detection	44.57 (0.82)	11.22 (0.34)	42.56 (0.83)	37.91	39.52	36.70
full	44.86 (0.60)	**11.83 (0.15)**	**42.97 (0.56)**	**42.29**	**48.54**	**45.20**

Table 2: Effect of removing various modules from our method.

PREFS better reflects summary quality than ROUGE. Interestingly, upper-bound ROUGE scores are lower than those for several models. We regard this as a shortcoming of ROUGE as a metric. Qualitatively, reading the different gold summaries shows that they are more similar to and accurate with respect to each other than any of the predicted summaries, including ours. Although they differ in phrasing and length, they all describe the same key events. Our proposed metric, PREFS, better reflects this similarity and gives a much higher score to 'upper-bound' than to any of the models. The large gap to get to the level of 'gold upper-bound' reflects the difficulty of the task. The comparison between ROUGE and PREFS suggests that the former is useful for detecting which summaries are of very low quality, e.g., if ROUGE-2 is < 5, the summary can confidently be regarded as poor. However, for distinguishing between multiple relatively decent summaries, PREFS is more useful.

Ablations show all modules are important, but the transcript is the most important of all. Table 2 shows the effect of removing four of the five modules. In 'w/o scene-detection', we remove the scene-detection module and just split the input into equally-sized chunks equal to the context size (1,024). In 'w/o transcript' we do not include summaries of the dialogue from the transcript, and the only input to the high-level summarization module is the output of the visual processing module. In 'w/o video' we do the reverse: use only the dialogue summaries without the output of the vision-processing module. In 'w/o reordering', we remove the scene-reordering module and present the scene summaries to the high-level summarization module in the order in which they appear. All these experiments are performed using the Kosmos-2 captions (except 'w/o video' which has no captions). As in Table 1, ROUGE is scored over five random seeds, PREFS for a single randomly selected seed.

Aside from ROUGE-1 and ROUGE-Lsum, which are roughly the same for 'w/o reordering' and 'w/o scene-detection', all ablation settings lead to a drop in all metrics. We provide detailed analysis of the accuracy of the scene detection module in Appendix E.1. Unsurprisingly, the largest drop is in 'w/o transcript' as most information is in the dialogue — a TV show without sound or subtitles is difficult to follow. Interestingly, however, this setting still gets some n-grams (ROUGE) and facts (PREFS) correct, showing that our model is extracting useful information from the video. This is also clear from the 'w/o video' setting, which shows a drop in all metrics. Qualitatively, we observe that the vision-processing module is most useful for identifying locations, e.g., 'at the hospital' which is generally not mentioned in the dialogue. For 'w/o scene-detection' and 'w/o reordering', the difference is moderate for ROUGE. For PREFS, it is more substantial and highly significant when taken over all facts in the dataset.

6 Conclusion

We addressed the task of summarizing TV shows from videos and dialogue transcripts. We proposed a modular approach where different specialized components perform separate sub-tasks. A scene-detection module splits the TV show into scenes, and a scene-reordering module places these scenes in an optimal order for summarization. A dialogue summarization module condenses the dialogue for each scene and a visual processing module produces a textual description of the video contents. Finally, a high-level summarization module fuses these into an output summary for the entire episode. We also introduced PREFS, a new metric for long-form summarization, based on splitting predicted and reference summaries into atomic facts. It captures both precision and recall, and correlates significantly with human judgments. In the future, we plan to test our method on even longer inputs, and explore settings where transcripts are not available.

Limitations

While the modular approach we propose has advantages, such as allowing specialization of individual modules, and the ability to replace one module without affecting the others, it also has the disadvantage that it is difficult to fine-tune all modules. We fine-tune only the high-level summarization module, whereas for a monolithic end-to-end model, all parameters can be trained.

Our proposed PREFS metric requires multiple calls to GPT which incurs a financial cost. We estimate that all the results presented in this paper cost about $1300. This is still many times cheaper than hiring human evaluators to extract and score facts manually, which we estimate would take 400 person-hours and cost about $15,000.

There is still a significant gap, in terms of PREFS, between our summaries and the upper bound of comparing the gold summaries to each other. This shows that the task is challenging and requires further advances to reach human-level.

References

Divyansh Agarwal, Alexander R. Fabbri, Simeng Han, Wojciech Kryscinski, Faisal Ladhak, Bryan Li, Kathleen McKeown, Dragomir Radev, Tianyi Zhang, and Sam Wiseman. 2022. CREATIVESUMM: Shared task on automatic summarization for creative writing. In *Proceedings of The Workshop on Automatic Summarization for Creative Writing*, pages 67–73, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R Gormley. 2023. Unlimiformer: Long-range transformers with unlimited length input. *arXiv preprint arXiv:2305.01625*.

Aanisha Bhattacharyya, Yaman K Singla, Balaji Krishnamurthy, Rajiv Ratn Shah, and Changyou Chen. 2023. A video is worth 4096 tokens: Verbalize videos to understand them in zero shot. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9822–9839, Singapore. Association for Computational Linguistics.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. *arXiv preprint arXiv:2310.00785*.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh. 2023. SEAHORSE: A multilingual, multifaceted dataset for summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9397–9413, Singapore. Association for Computational Linguistics.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. GO FIGURE: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 478–487, Online. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Peter Grunwald. 1998. *The minimum description length principle and reasoning under uncertainty*. University of Amsterdam.

Peter D Grünwald. 2007. *The minimum description length principle*. MIT press.

Shuo Guan and Vishakh Padmakumar. 2023. Extract, select and rewrite: A modular sentence summarization method. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 41–48, Singapore. Association for Computational Linguistics.

Tanmay Gupta and Aniruddha Kembhavi. 2022. Visual programming: Compositional visual reasoning without training. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Lear ning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Tadao Kasami. 1966. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020. MART: Memory-augmented recurrent transformer for coherent video paragraph captioning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2603–2614, Online. Association for Computational Linguistics.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958.

J. Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. 2020. Violin: A large-scale dataset for video-and-language inference. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10897–10907.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Cory S Myers and Lawrence R Rabiner. 1981. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *Bell System Technical Journal*, 60(7):1389–1409.

Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. 2020. Spatio-temporal graph for video captioning with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10870–10879.

Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. 2023. Kosmos-g: Generating images in context with multimodal large language models. *arXiv preprint arXiv:2310.02992*.

Bo Pang, Erik Nijkamp, Wojciech Kryscinski, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2023. Long document summarization with top-down and bottom-up inference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1267–1284, Dubrovnik, Croatia. Association for Computational Linguistics.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2021. Movie summarization via sparse graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13631–13639.

Pinelopi Papalampidi and Mirella Lapata. 2023. Hierarchical3d adapters for long video-to-text summarization. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1267–1290.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. volume abs/1705.08045.

T. Sheng and M. Huber. 2020. Unsupervised embedding learning for human activity recognition using wearable sensor data. In *Proc. FLAIRS*.

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1717–1729, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Simeng Sun, Yang Liu, Shuohang Wang, Chenguang Zhu, and Mohit Iyyer. 2023. Pearl: Prompting large language models to plan and execute actions over long documents.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralbã, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Hanhua Ye, Guorong Li, Yuankai Qi, Shuhui Wang, Qingming Huang, and Ming-Hsuan Yang. 2022. Hierarchical modular network for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17939–17948.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Ying Shan, Bing Li, Ying Deng, and Weiming Hu. 2021. Open-book video captioning with retrieve-copy-generate network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9837–9846.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*.

A Example Summaries

Table 3 shows the gold summary for an episode from the SummScreen3D testset. Tables 4 and 5–7 show the summary of our model, and those of comparison models, respectively, for the same episode.

The gold Summary of *The Bold and the Beautiful* episode (aired 05-05-06)
Ridge continues to beg Brooke to reconsider her decision to leave Forrester as Stephanie continues to voice her opinion. At Marone, Taylor pays Nick a visit. Nick is still angry about what Taylor implied when she disclosed that Brooke and Ridge slept together. Taylor tries to apologize and asks if things are all right between Nick and Brooke. Nick tells her that everything is fine and Brooke is quitting her job at Forrester. Taylor is unconvinced that Brooke will be able to let go of either Forrester or Ridge ! Brooke tells Ridge that she cannot fight with Stephanie any longer and that her future is with Nick. After kissing Ridge and saying that a part of her will always love him, she takes her things and leaves. Bridget and Dante are at home discussing Stephanie's interference in the custody of Dino. Bridget suggests that perhaps Eric can help them. Dante worries about what losing his job would do to his work visa. Bridget convinces him that because they all love Dino, they should be able to work something out. After some wine, Bridget reveals that she is ready to make love with Dante. As the two were in bed, Dante stops and reaches in the bedside drawer and presents Bridget with an engagement ring and pops the question. Brooke goes to see Nick at his office and tells him that she has left Forrester. Nick is pleased, although Brooke confesses that she hurt Ridge badly by walking out. Nick whisks her off to the Marone jet for a surprise getaway! At Forrester, Ridge angrily accuses Stephanie of causing all his problems with Brooke. Stephanie is stunned as Ridge bashes her with a vengeance and then clutching his chest, collapses to the floor!

Table 3: Gold summary from the SummScreen3D dataset for *The Bold and the Beautiful* episode (aired 05-05-06). Several summaries exist for this episode from different fansites. This summary is from TVMegaSite.

Modular-Kosmos Summary (ours) of *The Bold and the Beautiful* episode (aired 05-05-06)
Bridget and Dante plan to get married in Italy, but Bridget wants to spend more time with Dante. Stephanie tells Bridget that she wants to fire Dante and send him to Italy. Ridge tells Stephanie that he wants Brooke out of Forrester Creations. Brooke tells Nick that she is through fighting, she is moving to Paris. Nick tells her that she has to move out of the office. Nick says that he will not force her to work in Paris, but she says that she needs to move in. Ridge says that Nick is her future, and she has no choice but to leave. Stephanie says that Ridge needs her, but he does not want to accept her. He says that they are better off without her. She says that is not what she wants. She tells him that she does not know how he feels. She does not understand how he could do this to Brooke. He tells her to move on with her life. Taylor apologizes to Nick for suggesting that Ridge and Brooke slept together. Nick tells Taylor that he doesn't let it affect his relationship with Brooke. Brooke tells Nick that she needs something to cheer her up, and Nick tells her that they are on their way to a tropical island. Ridge tries to convince Brooke to stay at Forrester Creations, but she tells him that she is leaving anyway. Taylor tells Ridge that she does not want to believe that he and Brooke have feelings for each other, but Ridge tells Taylor not to give up on her dreams. Nick and Brooke plan to take off on their trip, but Nick tells Brooke that he has a surprise for her, and that they will be going to a beautiful beach and a fruity drink. Brooke is very happy about Nick's trip plans, but when she asks where they are going, Nick reveals that he wants to take her to a romantic beach.

Table 4: The summary generated by our model for *The Bold and the Beautiful* episode (aired 05-05-06).

Unlimiformer Summary of *The Bold and the Beautiful* episode (aired 05-05-06)
Brooke tells Ridge that Stephanie will never let her and Nick run Forrester Creations. She tells him that she is sick of her mother's obsession with getting rid of her and she has had enough. Ridge tells his mother that she has been trying to get rid of Brooke for so long that it has become a compulsion for her. Bridget tells Dante that Stephanie cannot get away with this. She will call her dad and see what he will do. Dante tells Bridget that all she has to do is fire him and he will lose his work visa and go back to Italy. He needs a lawyer and Bridget says she will make sure he gets the best one. Dante thanks her for her support. She says he has no idea how much her support means to him.. Later, Dante opens a red velvet'.. Dante proposes to Bridget.. Source: tv.com tv.co.uk/soulmatestories: http://www.starz.com/dailymail.co/news/features/2013/01/25/24/ro-lorry-logan-says-that-if-he-walks-out-of-this-company-now they-may-never-get-back-what-they-had.html#storylinkc̄py. Brooke tells him she still cares about him. He says he doesn't care what his mother. ..Source: TV.co Back to Mail Online. Back to the page you came from: http: www. tv.cnn.com.com/. Back to the episode airs at 8 p.m. ET on Thursday, September 14, 2013. The episode airs on ABC. Click here to watch the full episode of The Young and the Restless.

Table 5: The summary generated by Unlimiformer (Bertsch et al., 2023) for *The Bold and the Beautiful* episode (aired 05-05-06).

Mistral-7B Summary of *The Bold and the Beautiful* episode (aired 05-05-06)
Ridge and Brooke are having a conversation about their relationship. Ridge tells Brooke that he doesn't care what his mother thinks or wants anymore. He only cares about her. Brooke tells Ridge that Nick doesn't want her working at Forrester Creations. Ridge tells Brooke that Nick is telling her what to do now. Brooke tells Ridge that she is through fighting and that Stephanie has won. Dante and Bridget are having a conversation about Dante's work visa. Dante tells Bridget that he needs a lawyer. Bridget tells Dante that she will make sure he gets the best possible one. Dante tells Bridget that he wants to thank her for her support. Bridget tells Dante that he deserves to play a huge part in Dino's life. Bridget tells Dante that she will do whatever she can to make sure that happens. Nick and Taylor are having a conversation about Ridge and Brooke. Nick tells Taylor that he wants the containers off the ship. Nick tells Taylor that his people have schedules to keep. Nick tells Taylor that time and money are important. Taylor tells Nick that she wants to apologize. Taylor tells Nick that she still thinks she did the right thing telling him that Ridge and Brooke slept together. Taylor tells Nick that she never should have implied that Brooke wasn't being completely honest. Nick tells Taylor that he never believed her anyway.

Table 6: The summary generated by Mistral-7B for *The Bold and the Beautiful* episode (aired 05-05-06).

Adapter e2e Summary of *The Bold and the Beautiful* episode (aired 05-05-06)
Ridge: "You deserve to play a huge part in his life. And I'm going to do whatever I can to make sure that happens" "This is where we began. This is where our love affair began, Ridge. And it only seems fitting that this is where it should end" "I will never forget our accomplishments here. And part of me will always wonder what more we could have done together" "We'll leave L.A. and move to Paris. We'll work out of the office there – the international office" "You wouldn't even be with him if my mother hadn't broken us up. Maybe we would still be married. We'd be raising our children." "I'm sorry, I know that you want me to be relieved. But on the other, I just don't think that – I want you to be happy I've got a surprise for you" "It's really healed me, I guess. Because I wasn't ready to be with you." "You're the only one that's heartbroken if you got deported. After all, Dino wouldn't be the onlyOne that's Heartbroken if you got deported" "There's nothing anyone can say or do to change that now. You understand me? Good, I was hoping you'd say that. I never believed you anyway" "What do you want from me?" "I don't know what to say. I just want to be friends with you. I want to talk to you about everything." "What's your plan?" "We're going to Paris."

Table 7: The summary generated by the method of (Papalampidi and Lapata, 2023) for *The Bold and the Beautiful* episode (aired 05-05-06).

B Caption Filtering and Inference of Names

The output from the visual processing module for a given scene is a textual description of the visual information for that scene. Sometimes, this description is vague and merely adds noise to the high-level summarization module input. This is also clear from manually viewing the TV show episodes: many scenes are just headshots of characters talking, and do not convey plot information. For both SwinBERT and Kosmos-2, we filter out any captions that contain the following phrases 'a commercial', 'talking', 'is shown', 'sitting on a chair', 'sitting on a couch', 'sitting in a chair', 'walking around'. Addtionally, we replace occurrences of the phrase "is/are seen" with "is/are". This is because the captioning datasets these models are trained on often use the phrase "is/are seen", e.g., "a person is seen riding a bicycle" instead of "a person is riding a bicycle", but we do not want such passive voice constructions in our summaries.

The captions do not contain character names, as these names cannot be inferred from the video alone. Therefore, for each sentence in the output of the visual processing module, we employ the following method to insert names, where they can be easily inferred. We categorize each name appearing in the transcript for the scene as male, female or neutral, using the vocabulary list for English names from Python's NLTK. Similarly, we assume noun-phrases "he", "a man", or "a boy" are male and noun-phrases "she", "a woman", "a girl" are female. If there is only one male name, then we replace all male noun phrases with that name, similarly for female names. For example, from the caption for Scene 2 in *One Life to Live*, (aired 10-18-10), as shown in Figure 2, the output of the visual processing module contains the sentence "a man is kissing a woman", and the only names in the transcript for that scene are "Brody", which is listed as male, and "Jessica", which is listed as female. Our method to insert names then transforms this caption to "Brody is kissing Jessica".

Figure 2: A selected keyframe from Scene 2 in *One Life to Live*, (aired 10-18-10), which Kosmos-2 captions as "a man is kissing a woman". Our post-processing method to insert character names transforms this caption to "Brody is kissing Jessica".

C PREFS Metric

Often, facts extracted by GPT4 are too vague and uninformative to be useful for assessing summary quality. Therefore, we perform a simple filtering to remove such facts. Specifically, we eliminate out any facts containing the following words for phrases: "someone", "somebody", "something", "is a person", "are people", "is a character", "are characters". Additionally, we eliminate all facts containing only two words, as we observe they are almost always uninformative and often are an ungrammatical use of a transitive verb in an intransitive context. Table 8 shows the full list of facts extracted from our summary for *The Bold and the Beautiful*, (aired 05-05-06), showing which are eliminated, which are judged supported and which are judged unsupported.

D Scene Detection Details and Examples

Here, we provide further details on the scene detection algorithm from Section 3.2 and report a measure of its accuracy.

We calculate the cost for a given partition by assuming that the scene breaks under this partition and the placeholders where the speakers would go are given, and then asking how many bits are needed to fill in the speaker names. Then the receiver can infer the length of the scene-specific codebook as 2^m, where m is the length of each code in the scene. Note, we still need prefix-freeness, as just knowing the number of speaker lines in a scene doesn't allow us to distinguish between 10 followed by 11 and 101 followed by 1.

For example, consider the following sequence of character names from *As the World Turns* (aired 01-09-07): 'Elwood', 'Casey', 'Elwood', 'Casey', 'Elwood', 'Casey', 'Elwood', 'Casey', 'Elwood', 'Meg', 'Paul', 'Meg', 'Paul', 'Meg', 'Paul', 'Luke', 'Meg', 'Paul', 'Luke', 'Meg', 'Luke', 'Meg', 'Luke', 'Meg', 'Luke', 'Meg', 'Paul', 'Meg',

Bridget and Dante are planning to get married. Bridget wants to spend more time with Dante. They plan to get married in Italy. Stephanie wants to fire Dante. Stephanie wants to send Dante to Italy. Stephanie tells Bridget. Bridget receives the information from Stephanie. Ridge wants Brooke out of Forrester Creations. Ridge told Stephanie this. Brooke tells Nick something. Brooke tells Nick that she is through fighting. Brooke tells Nick that she is moving to Paris. Nick told her something. Nick told her she has to move. She has to move. She has to move out of the office. Nick told her she has to move out of the office. Nick will not force her. She will not work in Paris. She needs to move in. She needs to move to Paris. She will not work in Paris but she needs to move in. Ridge believes that Nick is her future. Ridge has no choice. Ridge must leave. Stephanie believes Ridge needs her. Ridge does not want to accept Stephanie. Stephanie and Ridge have a relationship. Stephanie and Ridge have a troubled relationship. He says something. He says that. He says that they are better off without her. She says. That is not what she wants. She tells him something. She tells him that she does not know. She tells him that she does not know how he feels. She does not know how he feels. She does not understand. He could do this to Brooke. He tells her something. He tells her to move on. He tells her to move on with her life. He does not have children. He is a proud uncle. He has four nieces and nephews. Taylor apologized. Taylor apologized to Nick. Taylor suggested that Ridge and Brooke slept together. Ridge and Brooke are two people. Ridge and Brooke slept together. Nick tells Taylor something. Nick tells Taylor he doesn't let something affect his relationship with Brooke. Nick has a relationship with Brooke. Nick's relationship with Brooke is affected by something. Nick doesn't let something affect his relationship with Brooke. Brooke tells Nick that she needs something to cheer her up. Nick tells her something. Nick tells her they are on their way. They are on their way to a tropical island. Ridge tries to convince Brooke. Brooke is leaving. Brooke is leaving Forrester Creations. Brooke is leaving Forrester Creations anyway. Taylor tells Ridge something. Taylor does not want to believe something. Ridge and Brooke have feelings for each other. Ridge tells Taylor something. Taylor should not give up on her dreams. Nick and Brooke plan to take off on a trip. Nick has a surprise for Brooke. They will be going to a beautiful beach. They will be going to a beach. They will be going to a fruity drink. They will be going to a beautiful beach and a fruity drink. Nick tells Brooke about the surprise. Nick tells Brooke that they will be going to a beautiful beach. Nick tells Brooke that they will be going to a fruity drink. Brooke is happy about Nick's trip plans. Nick has trip plans. Nick wants to take Brooke somewhere. Nick wants to take Brooke to a beach. The beach is romantic.

Table 8: Facts extracted with GPT4 from our summary of *The Bold and the Beautiful*, (aired 05-05-06), from SummScreen3D. Facts that we filter out are written in black, those that are judged as supported by the gold summary are in blue, while those unsupported are in red. After applying our filtering procedure, the number of facts reduces from **83** to **67**, of which 33 are judged supported and 34 unsupported, giving a factscore-precision of 49.25.

'Luke', 'Meg', 'Luke', 'Meg', 'Luke', 'Meg', 'Luke', 'Meg', 'Luke', 'Adam', 'Gwen', 'Adam', 'Gwen', 'Adam', 'Gwen', 'Adam', 'Gwen', 'Adam'.

Abbreviating the character names as their first letters, the correct partition is

1. ECECECECE,
2. MPMPMPLMPLMLMLMLMPMPMLMLMLML,
3. AGAGAGAGA

Now consider the cost, as defined in Section 3.2 of this partition. The total vocabulary size is

$$N = |\{E, C, M, P, L, A, G\}| = 7\,.$$

For the first scene, $n_1 = 2, n_2 = 3, n_3 = 2, l_1 = 9, l_2 = 28, l_3 = 7$. Letting c_i denote the cost for the ith scene, we then have similarly,

$$\begin{aligned} c_1 &= \log \binom{7}{2} + 9 \log 2 \approx 13.392 \\ c_2 &= \log \binom{7}{3} + 28 \log 3 \approx 49.508 \\ c_3 &= \log \binom{7}{2} + 9 \log 2 \approx 13.392\,, \end{aligned}$$

giving a total cost of approximately $13.392 + 49.508 + 13.392 = 76.292$.

Now compare this to the cost of a different partition of this sequence, say into scenes of uniform length:

1. ECECECECEMPMPMP,
2. LMPLMLMLMLMPMPM,
3. LMLMLMLAGAGAGAGA

In that case

$$c_1 = \log \binom{7}{4} + 15 \log 4 \approx 34.392$$
$$c_2 = \log \binom{7}{3} + 15 \log 3 \approx 28.167$$
$$c_3 = \log \binom{7}{4} + 16 \log 4 \approx 36.392\,,$$

giving a total cost of approximately $34.392 + 28.167 + 36.392 = 98.951$. Therefore, because our algorithm computes the global minimum to this cost function, it is guaranteed to choose the correct partition over this uniform one. In fact, it selects the correct partition.

E Additional Results

E.1 Accuracy of Scene-detection Algorithm

Here, in Table 9, we report an empirical test of the accuracy of our scene-detection algorithm. These figures are produced using an episode which has explicitly marked scene breaks. We assign each transcript line a label based on the scene it is in with respect to these explicit scene breaks, e.g., all lines in the first scene get the label '0'. Then we do the same for the scene breaks produced by our algorithm, and compare the two sets of labels using unsupervised label comparison metrics, as commonly used in clustering problems: accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI), defined as, e.g., in Sheng and Huber (2020). We compare to two baselines. The first, denoted 'uniform', divides each episode into n equally-sized scenes, where n is the average number of scenes per episode in the dataset. The second, denoted 'uniform oracle', does the same except sets n to the true number of scenes in that episode.

Our method produces more accurate scene splits than both baselines, despite taking no information from the ground-truth labels. Moreover, many of the occasions on which it differs from the ground-truth scene splits appear at least as reasonable as the ground-truth splits when we observe only the character names in the transcript. This suggests that errors in the predicted splits are due not to the algorithm itself but to the fact that, at present, it only uses character names and ignores the speech itself. A future extension is to use named entities or all nouns from the speech as well as character names.

	acc	nmi	ari
ours	0.890	0.881	0.766
uniform oracle	0.759	0.819	0.538
uniform	0.723	0.809	0.489

Table 9: The accuracy of the scene splits produced by our method, benchmarked against two methods that split the transcript uniformly: 'uniform' splits into the average number of scenes in the dataset; 'uniform oracle' into the ground-truth number of scenes for each episode.

E.2 BERTScore

Tables 10 and 11 report results on summary quality according to BERTScore (Zhang et al., 2019). Tables 10 shows BERTScore for our model and comparison models, the analogue of Table 1 in the main paper, and Table 11 shows BERTScore for the ablation settings, the analogue of Table 2 in the main paper.

We find this metric does not distinguish well between the different settings, and scores all models very similarly. Even the "w/o transcript" setting, which qualitatively misses most of the important information in the episode, and scores poorly on ROUGE and PREFS, gets a high BERTScore. Moreover, even within each setting, the scores are very similar across different episodes, again in contrast to ROUGE, PREFS and qualitative evaluation (some episodes appear much easier to summarize than others). The average standard deviation across episodes, within each setting, is 2.2, 1.39, and 1.41 for BERTScore-precision, BERTScore-recall and BERTScore-f1 respectively. In contrast, the same standard deviation for ROUGE-1, ROUGE-2, ROUGE-Lsum are 6.66, 3.32, 6.52, while for factscore-precision, factscore-recall and PREFS, they are 14.92, 15.14 14.94, respectively.

This suggests that BERTScore always returns a similar score, regardless of the input. This inability of BERTScore to adequately distinguish different settings was also reported by Papalampidi and Lapata (2023).

F Significance Calculations

The improvements in our model over comparison models, as shown in Table 1 are statistically significant at $\alpha < 0.01$. We now calculate an upper bound on all metrics to establish this, for an unpaired unequal variances T-test, i.e.. a Welch test. The t-value for such a test is calculated as

	bs-precision	bs-recall	bs-f1
llama-7b	75.48 (1.22)	78.99 (0.21)	77.13 (0.74)
mistral-7b	79.85 (0.04)	81.15 (0.07)	80.47 (0.05)
central	79.33 (0.39)	82.46 (0.19)	80.82 (0.24)
startend	80.09 (0.98)	82.58 (0.30)	81.29 (0.54)
unlimiformer	82.69 (0.64)	83.27 (0.43)	82.96 (0.53)
adapter-e2e	78.54 (0.09)	81.39 (0.01)	79.91 (0.05)
modular-swinbert	83.29 (0.23)	83.58 (0.24)	83.42 (0.20)
modular-kosmos	82.46 (0.81)	83.54 (0.28)	82.98 (0.33)
upper-bound	83.40 (1.87)	84.45 (2.35)	83.91 (1.82)

Table 10: BERTScore for our model as well as all comparison models.

	bs-precision	bs-recall	bs-f1
nocaptions	83.30 (0.20)	83.68 (0.22)	83.48 (0.10)
w/o video	83.62 (0.09)	83.55 (0.23)	83.57 (0.13)
w/o reorder	82.82 (0.18)	83.52 (0.06)	83.16 (0.10)
w/o transcript	83.10 (0.15)	81.45 (0.09)	82.25 (0.08)
modular-kosmos	82.46 (0.81)	83.54 (0.28)	82.98 (0.33)

Table 11: BERTScore for our our ablation settings.

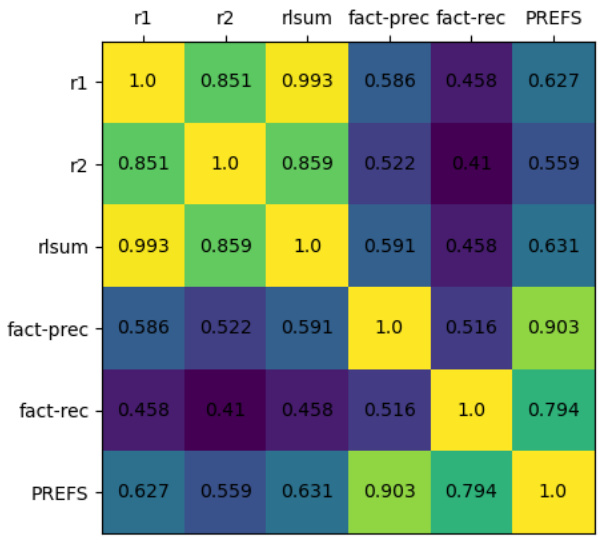

Figure 3: Correlation between all pairs of metrics that we report in Section 5. All are weakly correlated, with a stronger correlation between the different varieties of ROUGE.

$$t_w = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}, \quad (3)$$

where μ_i and σ_i^2 are the mean and variance, respectively, for the ith sample. Let μ_1, σ_1^2 denote the mean and variance for our model, and μ_2, σ_2^2 those for some comparison model. Excluding Llama-7B as a special case, which we consider below, then, for ROUGE1, $\mu_1 0.6$, and $\sigma_2 \leq 0.42$. Therefore, the denominator in (3) is $\leq\approx 0.328$. Also, $\mu = 44.86$ $\mu_2 \leq 42.24$. Thus,

$$t_w \geq \frac{44.86 - 42.24}{\sqrt{\frac{0.6^2}{5} + \frac{0.42^2}{5}}} \approx 4.47\,.$$

Similarly, for ROUGE2

$$t_w \geq \frac{11.83 - 10.32}{\sqrt{\frac{0.15^2}{5} + \frac{0.34^2}{5}}} \approx 4.78\,.$$

and for ROUGE-Lsum

$$t_w \geq \frac{42.49 - 40.40}{\sqrt{\frac{0.56^2}{5} + \frac{0.34^2}{5}}} \approx 4.72\,.$$

Using the Welch-Satterthwaite equation, the degrees of freedom ν can also be lower-bounded at 5, giving a critical value, for $\alpha = 0.1$ of 3.37, which is less than our t-value for each for ROUGE1, ROUGE2 and ROUGE-Lsum.

For FActScore, we do not run multiple random seeds because of the financial cost, but if we compare distributions over individual facts, we see all comparisons are very highly significant. Each setting involves $\sim 15,000$ facts, with the 'no transcript" ablation having fewer, $\sim 4,000$. Expressing fact-precision and fact-recall as fractions rather than percentages, and recalling that Bernoulli distribution with mean p has variance $p(1-p)$, we see that the denominator in (3) is upper-bounded by $\frac{2}{4000}$, so the t-value is lower-bounded by $2000(\mu_1 - \mu_2)$. This is in the hundreds for all μ_1, μ_2 from the factscore-precision and factscore-recall columns in Table 1, which is far above the critical value of 2.33.

G Correlation Between Metrics

Figure 3 shows the pairwise correlation (Pearson) between all metrics that we report in Section 5. This is taken across all data points in all settings. All metrics are at least weakly correlated with each other, with the strongest correlations between the different varieties of ROUGE. This suggests that our PREFS metric captures information outside of what is captured by ROUGE.