Recipe2Video: Synthesizing Personalized Videos from Recipe Texts

Prateksha Udhayanan[1], Suryateja BV[2]*, Parth Laturia[3]*,
Dev Chauhan[4]*, Darshan Khandelwal[5]*, Stefano Petrangeli[1], and Balaji Vasan Srinivasan[1]
[1]Adobe Research; [2]Avanti Fellows; [3]Morgan Stanley; [4]Graviton Research Capital LLP; [5]Goldman Sachs
udhayana@adobe.com, suryateja@avantifellows.org, dev.chauhan@gravitontrading.com
parthlaturia@gmail.com, darshankhandelwal1218@gmail.com, {petrange,balsrini}@adobe.com

Abstract

Procedural texts are a special type of documents that contain complex textual descriptions for carrying out a sequence of instructions. Due to the lack of visual cues, it often becomes difficult to consume the textual information effectively. In this paper, we focus on recipes - a particular type of procedural document and introduce a novel deep-learning driven system - Recipe2Video that automatically converts a recipe document into a multimodal illustrative video. Our method employs novel retrieval and re-ranking methods to select the best set of images and videos that can provide the desired illustration. We formulate a Viterbi-based optimization algorithm to stitch together a coherent video that combines the visual cues, text and voice-over to present an enhanced mode of consumption. We design automated metrics and compare performance across several baselines on two recipe datasets (RecipeQA, Tasty Videos). Our results on downstream tasks and human studies indicate that Recipe2Video captures the semantic and sequential information of the input in the generated video.

1. Introduction

Documents are rich sources of information and we consume a wide range of these in our day-to-day lives - novels, technical reports, manuals, etc. Procedural documents are a special type of documents that are used as a reference for carrying out a sequence of instructions (e.g., an Ikea assembly manual providing a step-by-step guide towards a furniture assembly). The presence of complex textual descriptions and absence of appropriate illustrations can make the consumption of such documents difficult. For instance, in a recipe, a user may find it difficult to identify certain ingredients or to visualize intricate cooking procedures.

Edgar Dale's 'Cone of Experience' ('Learning Pyramid') and the other study [37] indicate that visual content improves the cognition of information. Motivated by these studies, we propose ***Recipe2Video***, that alleviates the challenges in the consumption of such documents by automatically converting them into illustrative videos. While our algorithms are generic by design, we specifically focus on recipes as domain, and convert recipe texts into illustrative videos. Given a recipe document, we synthesize an explanatory video tailored to the expertise of the user, thus enhancing the consumption experience. Our illustrative video not only provides users with distinctive information modes, but also with an opportunity to engage in self-correction via comparison with visual outcomes in the generated video.

Our key contributions are: **(1)** a novel end-to-end pipeline that synthesizes different video variants for a procedural document; **(2)** novel mechanism to retrieve, re-rank, and efficiently select the right combination of assets (text, images, and videos) for a given procedural instruction; **(3)** a novel optimization framework based on Viterbi algorithm to create a seamlessly transitioning video that can take into account the overall relevance and coherence across multiple frames; **(4)** evaluation metrics based on cognitive models of procedural text understanding.

The remainder of this paper is organized as follows. Sec 2 presents related work in this domain. Sec 3 details our Recipe2Video framework from a systemic and algorithmic point of view. Extensive quantitative results are presented in Sec 4. An in-depth analysis of human studies is presented in Sec 5 and Sec 6 presents limitations and future work.

2. Related Work

Synthesizing videos from procedural texts by converting complex text into consumable multi-modal combinations is a novel problem that has not been tackled in its entirety yet. Therefore, we outline the prior works that address components of the general problem we are interested in.

Url2Video [10] converts an input webpage into a short video representing the contents of that webpage. This solution focuses on visual display by utilizing CSS elements from the webpage and assigns importance to content using a combination of keywords and CSS attributes. A re-

*Work done while at Adobe Research

cent work [21] synthesizes audiovisual slideshows using hardcoded word concreteness from input text. We instead learn to process the input text by understanding semantics and synthesize coherent multi-modal combinations that enhance information consumption. Some video creation startups [4, 2] pick text from an input article and add audio-visual components from a predefined library to synthesize videos. However, there are no attempts to decide what is the right combination of modalities to display for a given context which is one of our core contributions.

DOC2PPT [14] converts a document into a slide-deck/PPT. Their method combines document summarization, image and text retrieval, slide structure and layout prediction to arrange key elements in a form suitable for presentation. However, the output slides do not support modalities other than images and text. Further, the slides are not optimized for an end-to-end visual coherence, which is a key aspect in our problem. Also, Doc2PPT relies on supervision from a corpus of research papers and the corresponding slide decks, and hence is limited to academic papers.

CookGan [43] focuses on synthesising the image of a cooked dish based on the input ingredient list. It accounts for the changes in the appearance of the dish due to different cooking methods and captures final appearance in the generated image. However, we are more interested in generating a video rather than individual multi-modal components.

Li et al. [22] generate videos from text by training a conditional generative model to extract both static and dynamic information from text using a hybrid framework [22]. Their method focuses on general textual description and the generation is limited to natural scenes, hence does not naturally extend to procedural documents. However, they do provide insights around several models for text-to-video conversion, which we have leveraged in our work.

Another related work is **B-Script** [17], which determines the right content and positions of B-roll and inserts it within the main footage. Recent works in multi-modal summarization generate a summary text along with the most relevant images [44, 37]. While these works are not directly applicable to our problem, they provide key insights towards generating multimodal outputs and evaluating the generation.

3. Recipe2Video: System Architecture

Given a recipe document, Recipe2Video first retrieve different assets including clips and images for each instruction and arrive at a combination of assets that best depicts every component and action in that instruction. We score and re-rank the retrieved assets to capture their ability in covering the information presented in the instruction. The ranking also accounts for the temporal aspects of the components or actions in the instruction. Next, we perform a modality choice (clip or image) for every instruction to generate a frame that minimizes the cognitive load of the user. Figure 1 displays a schematic of the Recipe2Video architecture.

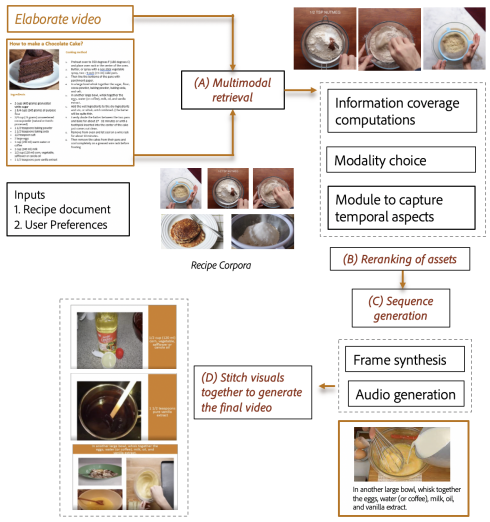

Figure 1. A schematic of the Recipe2Video system. Given a a procedural recipe text along with a communicative goal, our framework synthesizes a tailored video catering to the goal.

We consider fine-grained variations of video for catering to different user needs. In our work, we consider two possible variants - the *elaborate variant* for users who prefer a detailed multimodal depiction of contents in the input document. Such users could be novices or careful users who do not want to miss out on any detail and use the video for self-correction. This typically contains larger number of visual assets with longer duration. The *succinct variant* caters to users who prefer quicker depictions of contents in the document. Such users could be experts who want a quick reference. This variant contains a smaller number of assets that cover larger chunks of information within a short duration.

3.1. Multimodal Retrieval:

Any procedural text, including recipes, typically contains an enumeration of different components used, followed by a sequence of instructions. As a first step, we retrieve visuals (referred to as assets from here on) such as images and clips from a large corpus that illustrate the components and actions. In Recipe2Video, the corpus is built by combining images and videos from the RecipeQA [41], TastyVideos [38, 39], and YouCook2 [42] datasets. We extract and store short clips (from full recipe videos) based on the available ground truth timestamp annotations. The objective is to arrive at unitary clips that illustrate a specific process that can be used independently of the full video. For every instruction, we gather a set of relevant assets by combining the retrievals obtained from three different mechanisms, to improve robustness and guarantee enough richness in the selected assets for further processing.

(1) Textual Indexing-based Retrieval: We use the descriptions associated with the assets to index them using a weighting-based ensemble of models [25, 31]. We use

hyper-geometric divergence from randomness weighting model [6] to score, rank and retrieve the indexed assets.

(2)Textual Embeddings-based Retrieval: We compute a similarity score between the pretrained word2vec embeddings [27] of asset description and instruction to rank assets.

(3) Cross-modal Semantics-based Retrieval: Since the first 2 approaches do not focus on the semantics of the retrieved modalities, we use recently proposed deep networks for multimodal representations [32, 26] to project assets and text instructions into a common representation space and use the similarity in this space to rank the assets. For images, we use the CLIP (Contrastive Language-Image Pre-Training) embeddings [32] pretrained on 400 million image-text pairs. We retrieve the images from our corpus whose embeddings have maximum cosine similarity to the text instruction embeddings. For videos, we use the model from [26] which learns a joint text-video embedding by leveraging video clip-caption pairs. Since the model is pretrained on HowTo100M dataset [26], we finetune it on our dataset and use it to extract video and text embeddings.

3.2. Ranking Assets and their Combinations:

To prune the set of assets retrieved, it is vital to consider the relevance and value of each asset towards illustrating the given instruction beyond semantics as described in the previous subsection. Often, an instruction might need a combination of image(s) and clip(s) to be completely illustrated. Hence, we evaluate every retrieved asset for its ability to depict the instruction and arrive at a combination (if required) to best cover the entire instruction. Our system uses the following computations to rank assets and their combinations.

(1) Ranking with Information Coverage Scores: In this step, we focus on scoring and ranking assets based on the extent to which they depict the key aspects of the instruction. We extract the key phrases of an instruction [36] and then compute a set of scores that indicate the affinity of each of these key phrases to the retrieved assets using a Zero-shot Classifier extended from the CLIP [32] model. For videos, we compute and aggregate the affinity of a list of representative keyframes. More formally, let $t_1, t_2, \ldots, t_K$ be the key phrases extracted from an instruction. For each image I (or aggregation of keyframes), we compute the distribution over key phrases P_C given by $P_C\left(I,k\right) = \frac{\exp\left(e^{I\,T} e_k^t\right)}{\Sigma_{i=1}^{K} \exp\left(e^{I\,T} e_i^t\right)}$, where, $e_k^t = \text{CLIP\_TEXT}\left(t_k\right)$, $e^I = \text{CLIP\_IMAGE}\left(I\right)$, with $\text{CLIP\_TEXT}(.)$ and $\text{CLIP\_IMAGE}(.)$ being the text [33] and image [12] encoders. We assume that an ideal asset (combination) should uniformly cover all aspects of the instruction and compute the KL divergence between the coverage distribution obtained above and a uniform distribution and use it to rank assets. Let $P_U \sim \text{Unif}\left(K\right)$ be the uniform distribution over K key phrases. The asset with the highest rank (thus maximizing information coverage) corresponds to a such that,

$$a = argmin_{a \in A}\, KL\left(P_c\left(a\right) || \, P_U\right)$$

$$= argmax_{a \in A} \Sigma_{k=1}^{K} P_C\left(a,k\right) \frac{\log\left(P_C\left(a,k\right)\right)}{log(P_U\left(k\right))} \quad (1)$$

where $KL\left(P_c\left(a\right) || \, P_U\right)$ serves as a measure for information coverage. An example-based explanation for this approach is provided in the supplementary material.

(2) Ranking with Temporal Aspect Scores: Finding visual assets that integrate well with recipe texts is challenging because these texts describe several temporal aspects like the change in state of the components, etc. To tackle this, we leverage the CITE (Corpus of Image-Text Relations) dataset [5], which contains human-annotated answers to temporal questions on image-text pairs derived from RecipeQA [41]. We use the following subset of questions from CITE: (1) Does the image show how to prepare before carrying out the instruction? (2) Does the image show results of the action described in the instruction? (3) Does the image depict an action in progress described in the instruction? We believe that the ability of an asset to answer these questions helps in providing information about the preparation, execution, or results of an instruction, thus embedding the temporal aspects of the instruction. We train a feed-forward neural network, called *Temporal Classifier*, on this dataset using the CLIP embeddings [32] of assets and texts as input. The trained model is run on all the retrieved assets to obtain a set of confidence scores for each of the temporal aspects introduced before. For videos, we take the average of the scores computed on all key frames. Akin to information coverage, we score all the assets and their combinations to arrive at the aggregated scores that indicate their ability to capture different temporal aspects. For each (instruction, retrieved asset) pair, we compute $s_{bef}, s_{aft}, s_{dur}$ that indicate the scores corresponding to the three temporal aspects.

The characterization of the temporal aspects into 3 categories also us to synthesize a video emphasizing on different aspects of the procedure. By default, we give equal weights to all 3 questions in our ranking. We compose 2-asset and 3-asset combinations for each of the elaborate and succinct variants using $s_{bef}, s_{aft}, s_{dur}$ scores. In the succinct case, we pick the top-k ($k = 2, 3$) assets that rank the highest on the average of the 3 scores so that the higher-ranked assets contain all 3 temporal aspects, leading to fewer assets with wider information range.

In the elaborate case for 3-asset combination, we first pick the top-n ($n = 5$) assets that rank the highest in each individual temporal aspect. Then, we consider all n^3 combinations of assets and pick the combination that ranks highest on the summation of their temporal aspect scores. In the elaborate case for 2-asset combination, we again pick the top-n ($n = 5$) assets that jointly rank the highest in two

aspects ($[s_{bef}, s_{aft}]$ or $[s_{dur}, s_{aft}]$) and iterate over all n^2 combinations, picking the one that ranks the highest on the summation of their joint scores.

(3) Ranking with Modality Appropriateness Scores: While information coverage and temporal aspect based rankings provide us a list of asset combinations that cover information and key temporal questions on the procedure, they do not address whether these combinations are the ideal modalities to represent the instruction. To determine the appropriate modalities for each instruction, we utilize the concept of weak supervision [20, 29], which captures supervisory signals such as heuristics, constraints, or data distributions on a small sample and extends it to a larger corpus. Given an unlabeled dataset, akin to recipe instructions, weak supervision enables programmatic creation of labels for this dataset via labelling functions. We design multiple labelling functions (LFs) based on cognitive models for procedural text understanding [15] that capture domain expertise and simple intuitions of human annotator behavior. Each LF labels a subset of the data, and multiple LFs ensure that a large proportion of data is labelled entailing high coverage. A single data point can be labelled by multiple LFs, thereby reducing noise, and making the process robust. We use the following LFs in our system to compute models for computing the modality appropriateness:

a. Action Type: We identify verbs (actions) [9] from instructions and classify them into categories based on our inductive biases and cognitive studies [7, 15]. These categories are then mapped to their appropriate modality. For e.g., one-time actions -> image modality: e.g., *bake in oven*; general actions -> text modality – e.g., *leave for* 5 *minutes*; Repetitive actions -> short clips: e.g., *whisking*.

b Action Count above a threshold -> Video modality: Instructions containing multiple verbs cannot be illustrated with a single image, hence videos should be preferred.

c. Instructions with Numerical Quantities -> Textual modality: Quantitative information, e.g., *3 spoons of sugar*, *Some wheat flour* in recipes, is illustrated better via text as it provides accurate and immediate actionable knowledge of the material to collect [7].

We verify that our LFs cover the entire instruction dataset via the coverage metric provided by [34]. We use a majority label consensus to resolve conflicts when different LFs label an instruction differently, which also serves as a denoiser to our weak labelling. Thus, we arrive at a weak-labelled dataset that contains textual instructions mapped to one of the four labels (text, image, image-list, video), which determines the appropriate modality for each instruction. We train a multi-class classifier on this dataset using CLIP embeddings of instructions with a cross-entropy loss. At inference, the trained classifier predicts a 4-dimensional vector for every instruction, with each dimension representing a score for each of the labels. We use one of these scores as $Mod\,(y_i)$ based on the asset combination y_i.

3.3. Sequence Generation & Video Synthesis:

Prior works in automatic video editing/generation [28, 24, 23] state that visual and semantic coherence of the output video is a key requisite for user consumption. We make decisions at an instruction level to ensure a coherent video. We start with the CLIP embeddings of each asset combination and use the cosine similarity between them as a measure of coherence of their transitions, similar to works that compute textual coherence [40].

Apart from the visual and semantic coherence, the chosen combination of assets for each instruction should also optimize for information coverage, temporal coverage, and modality appropriateness. We solve this by formulating a Viterbi-style dynamic programming problem [19], used in sequence prediction tasks to generate outputs that maximize local independent scores (coverage & modality appropriateness) and transition scores between consecutive elements (visual & semantic coherence). For the chosen assets in each step, we assign a score for the sequence $\bar{y}$ as follows:

$$F\,(\bar{y}) = \sum_{y_i \in \bar{y}} S\,(y_i) + \sum_{y_i,\ y_{i+1}} T\,(y_i,\ y_{i+1}) \qquad (2)$$

where $S\,(y_i) = \text{weighted\_sum}(Rel\,(y_i)\,, IC\,(y_i)\,, TC\,(y_i)\,, Mod\,(y_i))$, all three scores are normalized and given equal weights, $Rel\,(y_i)$ = similarity score between the text embedding and the asset embedding, $IC\,(y_i)$ is the information coverage score (given by $1 - \sigma\,(kld\,(y_i))\,, \sigma(.)$ is the sigmoid function), $TC\,(y_i)$ is the temporal coverage score, $Mod\,(y_i)$ is the modality appropriateness score, and $T\,(y_i, y_{i+1})$ is the semantic similarity between y_i and y_{i+1}. With this approach, maximizing $F\,(\bar{y})$ will output sequences such that the inter-coherence of frames is high, making it smooth for users to follow a video. Our novelty is not Viterbi algorithm itself but using it with our measures and transitions in the context of multi-modal content which has not been done in prior work. In [28], transition probabilities are computed across uniform frames and audio clips which are then used in inference (similar to Viterbi) to stitch optimal frames. We use a similar transition scheme $T(.)$ with multi-modal assets instead. While [28] uses embedding similarity, we use our ranking measures (coverage, temporal score) in $S(.)$ that add to the novelty of our approach. Finally, to produce consumable video from our selected visuals, we chose an optimal template from a set of predefined templates for a frame. We utilize [3] to generate voice-over for the input instructions, overlay it with the corresponding frames and merge all such clips into our final video [45, 1].

4. Experimental Results

Evaluating our synthesized videos at large scale considering the overall users' experience is a non-trivial task.

We therefore design metrics to capture specific aspects of Recipe2Video. We consider two datasets for our evaluation – (1) RecipeQA [41] (test set), containing 960 recipe texts along with task-specific question-answer pairs; (2) Tasty Videos [39], containing 1000 recipe texts along with recipe categories. For each recipe text, we synthesize elaborate and succinct video variants from Recipe2Video.

Given the novelty of our end-to-end system, it is not straight-forward to compare our system with different baselines. Moreover, we do not have ground-truth frame sequences to compare our outputs with. We therefore adapt the following baselines that are closely related to our work to ensure a fair and exhaustive comparison. ***Audiovisual Slideshows*** [21] uses the notion of word concreteness to obtain search query from input text and uses it to retrieve assets. We test our retrieval module independently to replicate this baseline. ***Multimodal Summarization*** [44] aims to generate a multimodal (text-image) summarization of a multimodal document, while ensuring faithfulness to the input document. It is equivalent to our system, which contains the retrieval module and the information coverage component of the Ranking module. This also serves as an ablation for our Ranking module with regards to temporal aspects and modality appropriateness scoring. ***Doc2PPT*** [14] aims to generate slides sequentially from academic documents, by using a Hierarchical RNN with Progress Tracker (PT). However, it does not account for coherence. Since the code is not publicly available, we consider a variant of our model that replaces the Viterbi Decoding (Section 3.3) with a greedy decoding approach that does not consider optimizing the inter-frame transitions. We retain our ranking module in its entirety to match the strength of their hierarchical RNN model. Finally, ***Random Sampling*** is a naïve baseline where we sample assets for each step using a randomly generated query and combine into a video using greedy decoding, removing all other modules.

Note that, to the best of our knowledge, no previous work considers semantic video variants (as elaborate/succinct in our case) to meet different users' consumption needs . Thus, we report values on the standard video output of each baseline to compare with the elaborate variant synthesized by Recipe2Video, and consider a sped-up baseline video with fewer frames to compare with the succinct variant. We believe all the proposed baselines are relevant and competitive adaptations of existing approaches to better tackle the problem at hand. We reiterate that, unlike our proposed system, none of the baselines solve the problem in its entirety.

We adapt standard metrics to capture the performance of the different modules of our proposed system [15]. Note: all metrics (such as Visual Relevance) can be computed for text documents. We put a blank symbol (–) in these scenarios. We describe the considered metrics in the following.

Visual Relevance measures how visually close the assets in synthesized videos are to the corresponding input texts. We take pairwise cosine similarity of ViT representations [12] of assets and input document images and average over all videos. Note that the document images are used by Recipe2Video and are used only for evaluation. Since Tasty Videos recipes do not have images in the input document, we use this measure only for RecipeQA documents.

Textual Relevance measures how verbally close the assets in synthesized videos are to the input document. We take pairwise cosine similarity of sentence BERT [35] of video text and input document text and average over all videos. Video text is obtained using dense captioning [18] of extracted keyframes. A high value indicates that our method retains the verbal information of the procedure, and the assets are not obfuscating this information.

Action Coverage measures the number of verbs in the input document that are visually encoded in the final video. We count the number of verbs in the final video using dense captioning and compute the ratio with the input document's verbs. A high value shows that our method encodes verbs behaviorally into the visuals [15].

Video Quality measures the visual quality of the synthesized videos via Inception Score (IS) [30]. We use the pretrained Inception-v3 network to compute IS score, which is given by the exponential of averaged KL divergence between conditional $p(y|x)$ and marginal $p(y)$ probability distributions. A high video quality score indicates that our video frames are diverse and visually pleasing to the user.

Abrupt Info Gain measures the abruptness of information gained after each frame in the video. We calculate the distance between consecutive encoded representations of each frame and average the distances over the entire video. A high standard deviation of these distances indicates that the information conveyed to the user over the entire duration of the video is not smooth, thereby increasing cognitive load. Abruptness is given by $\sqrt{\Sigma_{t=1}^{N}\left(d_t - \hat{d}\right)^2/N}$ where $d_t = 1 - f_t^T f_{t-1}$ and $\hat{d} = \Sigma_{t=1}^{N} d_t/N$, with N being the number of frames and f_t the encoded representation of the frame at time t.

Summarization Score measures the ability of our videos to convey the same overall summary that the document conveys. We compute the sentence embeddings of input document and video text (from dense captions of extracted keyframes) and take the cosine similarity of all possible sentence combinations in each domain. We then use LexRank [13] to find the most central sentences that represent the extracted summaries. Comparing the summaries of input documents with generated video yields the required score.

Additionally, we also evaluate the capabilities of our synthesized videos on various downstream tasks. Note that Recipe2Video is not explicitly trained to perform well on these tasks. Instead, we hypothesize that Recipe2Video's

Table 1. Performance Comparison of various baselines against Recipe2Video on the RecipeQA and TastyVideos. All values are averaged over 962 input texts for Recipe QA and 1000 input texts for Tasty Videos. First row corresponds to input text documents. (↑) arrow indicates that a higher score on the metric is better. Some columns are left blank (–) since the input text document we consider does not have visuals / categories. The first column refers to the downstream task of Visual Coherence and is limited to the Recipe QA evaluation. The second column is the task of predicting categories from the context encoded from system outputs and is restricted to Tasty Videos.

Variant	System	Visual Relevance (↑)	Category Prediction (↑)	Textual Relevance (↑)	Action Coverage (↑)	Video Quality (↑)	Abrupt Info Gain (↓)	Summ. Score (↑)
Recipe QA	Text Document	–	–	1.00	–	–	0.52 (± 0.13)	1.00
Elaborate (Recipe QA)	Random Sampling	0.36	–	0.42	0.25	4.24 (± 0.54)	0.86 (± 0.22)	0.49
	Audiovis. Slides	0.52	–	0.55	0.51	4.09 (± 0.50)	0.38 (± 0.11)	0.62
	MSMO	0.78	–	0.84	0.56	4.04 (± 0.51)	0.41 (± 0.16)	0.73
	Doc2PPT	**0.81**	–	0.85	0.63	**4.31 (± 0.18)**	0.41 (± 0.10)	**0.71**
	Recipe2Video (Ours)	0.80	–	**0.85**	**0.72**	4.16 (± 0.46)	**0.26 (± 0.04)**	0.70
Succinct (Recipe QA)	Random Sampling	0.23	–	0.40	0.25	4.14 (± 0.35)	0.96 (± 0.34)	0.36
	Audiovis. Slides	0.42	–	0.45	0.51	4.28 (± 0.44)	0.59 (± 0.18)	0.58
	MSMO	0.56	–	0.63	0.56	4.16 (± 0.37)	0.53 (± 0.24)	0.65
	Doc2PPT	0.55	–	0.64	0.63	**4.25 (± 0.12)**	0.47 (± 0.21)	0.65
	Recipe2Video (Ours)	**0.78**	–	**0.85**	**0.68**	4.24 (± 0.23)	**0.34 (± 0.05)**	**0.73**
Tasty Videos	Text Document	–	0.52	1.00	–	–	0.58 (± 0.18)	1.00
Elaborate (Tasty Videos)	Random Sampling	–	0.45	0.44	0.26	4.40 (± 0.33)	0.58 (± 0.19)	0.47
	Audiovis. Slides	–	0.58	0.66	0.55	4.58 (± 0.60)	0.48 (± 0.13)	0.58
	MSMO	–	0.62	0.72	0.62	4.69 (± 0.59)	0.42 (± 0.20)	**0.72**
	Doc2PPT	–	0.63	0.77	0.71	**4.91 (± 0.58)**	0.44 (± 0.23)	0.71
	Recipe2Video (Ours)	–	**0.65**	**0.81**	**0.88**	4.78 (± 0.68)	**0.25 (± 0.05)**	0.68
Succinct (Tasty Videos)	Random Sampling	–	0.32	0.42	0.25	4.34 (± 0.28)	0.64 (± 0.28)	0.45
	Audiovis. Slides	–	0.38	0.47	0.54	4.19 (± 0.44)	0.55 (± 0.18)	0.58
	MSMO	–	0.49	0.55	0.53	4.72 (± 0.25)	0.46 (± 0.17)	0.62
	Doc2PPT	–	0.50	0.55	0.68	**4.88 (± 0.52)**	0.47 (± 0.18)	0.61
	Recipe2Video (Ours)	–	**0.63**	**0.72**	**0.82**	4.75 (± 0.61)	**0.31 (± 0.08)**	**0.71**

representations are strong enough to effectively solve these tasks, unlike other textual or baseline representations.

Given a context and a set of question images, the *Visual Coherence* task (in RecipeQA) predicts the best image (out of four available options) that best relates to the question images. We vary the context to compare our baselines. For videos, we compute the average of frame representations and concatenate them to ViT representations [12] of both question images and option images. We then reduce the dimensionality of these representations using Singular Value Decomposition (SVD) and compute the cosine similarity. We predict the option that has the highest cosine similarity with the set of question images as the final image.

In *Visual Cloze* task (for RecipeQA), given a context and a sequence of images with a placeholder, the task is to predict which image out of four available options fits well in the placeholder position. We again vary the context across baselines and compute SVD representations as explained earlier. We then replace the placeholder position with each of the option images and predict the option that leads to the lowest abruptness in information gain across the ordering.

In *Textual Cloze* task (for RecipeQA), given a context and a sequence of texts with a placeholder, the task is to predict which text out of four available options fits well in the placeholder position. We follow the previous computations replacing the frame representations with BERT representations. Note that both cloze tasks not only capture the representative strength of videos but also the strength of sequential information encoded in them.

In *Category Prediction* task (for Tasty Videos), we use the set of categories that comes with every recipe and predict the categories from the context (varied across baselines). We measure the performance using multi-label accuracy by taking the set intersection of true labels and top-10 labels with the highest similarity scores. We reduce the 51 available unique labels to 10 commonly occurring labels and add an "Other" label for the remaining 41 categories.

Metric-based Evaluation: We look at different statistics across the RecipeQA and TastyVideos datasets. RecipeQA has longer instructions (average of 475.48 words per recipe) with fewer steps (6.62 steps on an average) leading to fewer frames and assets retrieved. Tasty Videos has shorter instructions (139.70 words per recipe) with almost double the number of steps (12.60 steps per recipe), leading to more frames. Recipe2Video enables an easy and quick consumption across these long texts, by synthesizing succinct videos less than a minute long (39.04s for RecipeQA and 44.36s for TastyVideos) and elaborate videos of about two minutes (100.08s for Recipe QA and 109.17s for TastyVideos)

The average duration of succinct videos is less than half the duration of the elaborate videos, making it suitable for quick concise consumption of the input document.

Table 1 compares the performance of Recipe2Video against the baselines on the RecipeQA and Tasty Videos datasets, across different metrics. The first row computes scores for a textual document and serves as a reference to verify our hypothesis of presenting an alternate video modality to consume procedural texts. We observe

Table 2. Downstream Task Performance of various baselines against our Recipe2Video (Elaborate variant) on RecipeQA test set. (↑) arrow indicates that a higher score is better. We use the context encoded from system outputs for the tasks.

System	Visual Coherence (↑)	Visual Cloze (↑)	Textual Cloze (↑)
Text Document	0.71	0.36	**0.58**
Random Sampling	0.28	0.29	0.24
Audiovis. Slides	0.73	0.28	0.35
MSMO	0.75	0.29	034
Doc2PPT	0.78	0.45	0.54
Recipe2Video (Ours)	**0.79**	**0.56**	0.53

that Recipe2Video performs significantly better in smoothing the video consumption experience as indicated by the *Abrupt Info Gain*. This can be attributed to the overall coherence that Recipe2Video imparts in choosing the relevant assets. Recipe2Video also scores very high in the *Action Coverage* metric, owing to a strong temporal aspect ranking. Our variants perform well on all other metrics, with values close to the best baselines. High values on *Visual* and *Textual Relevance* show that our retrieval presents good performance. However, these values, along with *Video Quality* are lower than the Doc2PPT baseline, perhaps due to per-frame optimizations performed in Doc2PPT.

Succinct variant scores the highest on *summarization score* along with *Multimodal Summarization* baseline. This variant also has a higher visual quality as compared to the elaborate variant. Nevertheless, the ease of consumption is lower due to quickly changing information-heavy frames. *Textual Relevance* and *Summarization Scores* are similar across baselines since they are text dependent and all baselines process the text in a similar way (except for the Audiovisual Slideshows baseline that uses word concreteness and thus scores lower). *Textual Relevance* and *Summarization Scores* are the highest by default for text document since no changes are made to the text. However, the input documents contain no illustrative visuals and are very abrupt for consumption, leading to a potential suboptimal user experience. Scores of both our variants on all other metrics confirm our hypothesis that video modality is a much better way to consume procedural document, providing visuals for reference and self-correction, and enabling a smooth consumption.

Table 1 reports the performance of Recipe2Video and other baselines on Tasty Videos across various evaluation metrics described earlier. As reported in Table 1, the performance on Tasty Videos dataset follows a similar trend overall, reinforcing the benefits of our work. These results indicate that Recipe2Video generalizes to a different procedural text dataset with more steps and shorter sentences as compared to RecipeQA. Our method achieves strong coherence values across all the steps and retrieves highly relevant assets despite fewer query words available per step. We however note that the domains of both datasets are recipes, and leave extensions to other domains for future work.

Performance on Downstream Tasks: Table 2 compares downstream task performance of various baselines with Recipe2Video on the RecipeQA dataset. Our representations achieve a $+0.08$ gain on accuracy for the Visual Coherence task as compared to representations of input texts (computed by taking an average of Bert-based sentence representations of each step [35]). This can be explained by the higher quality of the visual assets that are selected to generate the video. We achieve significant gains in Visual Cloze task ($+0.20$) as compared to textual documents. It is worth stressing that this is a challenging task as it requires an understanding of the *exact sequence* of the visuals too. Our encoded video representations contain such sequential information, despite not having been trained for it explicitly. While we do not beat the text document representations on Textual Cloze task, we achieve similar performance. Since the task does not require visuals (the input text does not contain any), the visual assets we retrieve could be obfuscating and reducing the discriminative strength of the text in the video. These results confirm the promise of our approach in preparing richer representations and replacing standard BERT-based sentence representations for procedural texts.

5. Human Evaluation

While the automated metrics and tasks in previous section provide an insight into our system's performance, consumption experience is subjective and innate to users. We hence conduct extensive human studies via MTurk to capture the consumption experience of users on various dimensions like Enjoyability, Retainability, etc.

Our evaluation consists of three experiments, answering questions for: (1) consuming recipes from RecipeQA Test Set as procedural texts; (2) consuming recipes from RecipeQA Test Set as videos synthesized either by our system or from Doc2PPT system. Since Doc2PPT system was the the most competitive baseline in our quantitative evaluation, we use it for all our experiments in the human evaluation; (3) viewing and comparing recipes in both text and video modalities. Each text/video and corresponding questionnaire (HIT or Human Intelligence Task on MTurk) is annotated by five annotators. We also add few sanity check questions to ensure that annotators have actually gone through the displayed content, and reject HITs when annotators get any two of the three sanity questions wrong.

Here is one sample question. *[Retainable; 1, 2] How much of the recipe can you remember now without looking at it again?* (a) Cannot remember anything; (b) remember very little; (c) remember some of it; (d) remember most. Here, the question format indicates its usage in experiments (1) and (2) to gauge retainability of the displayed modality.

The purpose of Expt (1) and (2) is to keep the annotators unaware of a different form of consumption and arrive at the qualitative scores independently. We use recipes from RecipeQA Test Set as it already contains images for tasks

Table 3. Results of Human Evaluation for Expts (1) and (2) on RecipeQA test set. (↑) indicates that higher value is better. Note R2V stands for Recipe2Video.

Variant	System	Enjoyable (↑)	Retainable (↑)	Jarring (↓)	Task Performance (↑)	Pleasant (↑)	Intra Coherence (↑)	Inter Coherence (↑)
Text Document		3.02	2.55	2.27	0.58	–	–	–
Elaborate	Baseline	2.73	2,18	1.65	0.56	**2.79**	2.23	1.47
	R2V	**3.15**	**2.85**	**1.39**	**0.68**	2.77	**3.25**	**2.40**
Succinct	Baseline	2.77	2.21	1.73	0.53	2.36	2.35	1.60
	R2V	**3.11**	**2.84**	**1.64**	**0.68**	**2.81**	**3.28**	**2.44**

such as visual coherence, which serve as a proxy for informativeness provided by the consumption mode. Let N denote the number of tasks (data points) for each experiment. We consider ($N = 15$) RecipeQA texts as tasks for Expt (1), and ($N = 15 \times 4 = 60$) videos as tasks for Expt (2). Note that each recipe has two video variants synthesized by the baseline and two variants synthesized by Recipe2Video. Each option corresponds to a Likert scale [1-4] value. Table 3 shows the results of the two experiments. All values are averaged across the HITs, with a moderate to substantial inter-annotator agreement in the range $[0.48 - 0.68]$ (computed using Cohen's κ score). Expt (3) allows for a direct comparison between the two forms of consumption, text vs video. We use recipes from both RecipeQA (23) and Tasty Videos (25) amounting to $N = (23+25) \times 4 = 192$ videos, thus leading to robust results. Since each annotator interacts with one video, our study is between-subjects. We chose this mode because watching and analyzing more than one video increases cognitive load on annotators, leading to reduction in quality of responses.

Table 3 compares the performances of videos generated from baselines and Recipe2Video on various human annotated metrics. We find that respondents found our videos to be more enjoyable, more retainable, and less jarring as compared to traditional procedural texts. Videos synthesized by the baseline score worse than texts on most metrics showing that our system produces better videos. Interestingly, baseline videos score better on the *[Jarring]* metric (The *Jarring*-related question corresponds to the *Abrupt Info Gain* metric). This could indicate that, despite baseline videos being less enjoyable or retainable, the information flow is smoother to view. We find that videos synthesized by Recipe2Video have a greater intra- and inter-coherence as compared to baseline videos, thus confirming the strong sequence generation and optimization part of our system.

Furthermore, gains for the succinct variant are more pronounced than the elaborate one. This is expected since the succinct variant from Recipe2Video is not a sped-up elaborate video or a video with fewer frames as in the baselines, but a completely new video with optimal assets. This major novelty in preparing a semantically different variant is sufficiently reflected in the human study.

As a part of Expt (3), we ask respondents to rate whether the various modalities helped in understanding the recipe text better. Figure 2 show the results of this analysis, where

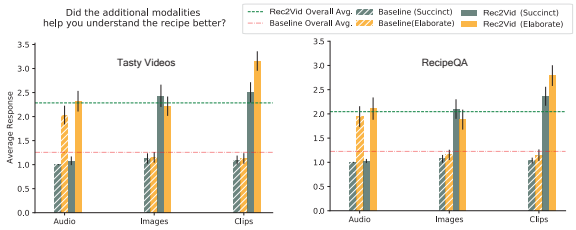

Figure 2. The bar graphs show the averaged values across all responses for each (modality, variant, system) triplet. Values are also averaged across all modalities and variants to obtain a system-level response, shown via dashed lines.

the values are averaged across all responses. We also average the responses across all modalities and display a horizontal line for each system. Overall, videos synthesized by Recipe2Video are moderately helpful (Average 2.28 for Tasty Videos, 2.04 for RecipeQA), and are significantly better than the baseline (Average = 1.22 for Tasty Videos, 1.26 for RecipeQA). Higher scores on Tasty Videos than RecipeQA could be due to shorter and crisper instructions in Tasty Videos, leading to better retrievals and shorter generated videos. Respondents find images in the succinct variant to be more helpful than images in the elaborate one. This points to the strength of our ranking module.

6. Conclusion

We introduce Recipe2Video, a novel deep learning-based system that converts procedural recipe texts into illustrative videos to enhance users' consumption experience. Recipe2Video uses various technologies to retrieve relevant multi-modal assets and rank them based on different dimensions such as temporality, information coverage and modality appropriateness. It then stitches them into an illustrative video using a Viterbi-inspired optimization scheme. While doing so, Recipe2Video also caters to user preferences - leading to semantically different variants (elaborate and succinct). Our quantitative and human evaluation demonstrate that the video variants from Recipe2Video: (i) are visually and textually relevant to the input recipe text; (ii) encode crucial actions in the form of clips for self-correction; (iii) provide a smooth information flow; (iv) are effective in capturing different user needs. Comparison with several baselines across two datasets shows significant gains of our system in producing coherent output videos.

Limitations & Future Work: Many of our system modules rely on the existence of specific datasets available in the recipe domain. There are very few equivalent datasets available in other domains. We intend to consider datasets like Tut-VQA dataset [11] in extending the proposed framework to other domains. We also look to optimize for layouts within the frames as incorporated in Url2Video [10] to enhance the video quality. For future work, we intend to analyze modalities and their combinations in greater detail to improve our modality appropriateness module [16, 8].

References

[1] Ffmpeg: A complete, cross-platform solution to record, convert and stream audio and video.

[2] Gliacloud. https//gliacloud.com. Accessed: 2021-01-28.

[3] Google text-to-speech. https://gtts.readthedocs.io/en/latest/. Accessed: 2021-01-28.

[4] Lumen 5. https://lumen5.com. Accessed: 2021-01-28.

[5] Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. CITE: A corpus of image-text discourse relations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 570–575. Association for Computational Linguistics, 2019.

[6] Gianni Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[7] Elisabeth André. The generation of multimedia presentations. *Handbook of natural language processing*, 12:305, 2000.

[8] John Bateman, Janina Wildfeuer, and Tuomo Hiippala. *Multimodality: Foundations, research and analysis–A problem-oriented introduction*. Walter de Gruyter GmbH & Co KG, 2017.

[9] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006.

[10] Peggy Chi, Zheng Sun, Katrina Panovich, and Irfan Essa. Automatic video creation from a web page. In Shamsi T. Iqbal, Karon E. MacLean, Fanny Chevalier, and Stefanie Mueller, editors, *UIST '20: The 33rd Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 20-23, 2020*, pages 279–292. ACM, 2020.

[11] Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Daisy Zhe Wang, and Doo Soon Kim. Tutorialvqa: Question answering dataset for tutorial videos. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5450–5455. European Language Resources Association, 2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[13] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128, 2011.

[14] Tsu-Jui Fu, William Yang Wang, Daniel J. McDuff, and Yale Song. DOC2PPT: automatic presentation slides generation from scientific documents. *CoRR*, abs/2101.11796, 2021.

[15] John T Guthrie, Stan Bennett, and Shelley Weber. Processing procedural documents: A cognitive model for following written directions. *Educational Psychology Review*, 3(3):249–265, 1991.

[16] Tuomo Hiippala and John A Bateman. Semiotically-grounded distant viewing of diagrams: insights from two multimodal corpora. *arXiv preprint arXiv:2103.04692*, 2021.

[17] Bernd Huber, Hijung Valentina Shin, Bryan C. Russell, Oliver Wang, and Gautham J. Mysore. B-script: Transcript-based b-roll video editing with recommendations. *CoRR*, abs/1902.11216, 2019.

[18] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4565–4574. IEEE Computer Society, 2016.

[19] G. David Forney Jr. The viterbi algorithm: A personal history. *CoRR*, abs/cs/0504020, 2005.

[20] Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. Self-training with weak supervision. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 845–863. Association for Computational Linguistics, 2021.

[21] Mackenzie Leake, Hijung Valentina Shin, Joy O. Kim, and Maneesh Agrawala. Generating audio-visual slideshows from text articles using word concreteness. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–11. ACM, 2020.

[22] Yitong Li, Martin Renqiang Min, Dinghan Shen, David E. Carlson, and Lawrence Carin. Video generation from text. *CoRR*, abs/1710.00421, 2017.

[23] Chang Liu, Han Yu, Yi Dong, Zhiqi Shen, Yingxue Yu, Ian Dixon, Zhanning Gao, Pan Wang, Peiran Ren, Xuansong Xie, Lizhen Cui, and Chunyan Miao. Generating engaging promotional videos for e-commerce platforms (student abstract). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020,*

The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 13865–13866. AAAI Press, 2020.

[24] Yong Liu, Min Wu, Chunyan Miao, Peilin Zhao, and Xiaoli Li. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Comput. Biol.*, 12(2), 2016.

[25] Craig Macdonald and Nicola Tonellotto. Declarative experimentation in information retrieval using pyterrier. *CoRR*, abs/2007.14271, 2020.

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *CoRR*, abs/1906.03327, 2019.

[27] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[28] Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A. Efros, and Trevor Darrell. Strumming to the beat: Audio-conditioned contrastive video textures. *CoRR*, abs/2104.02687, 2021.

[29] Trishala Neeraj. Data labeling using weak supervision: In action. *trishalaneeraj.github.io*, 2020.

[30] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

[31] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval, 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005, Proceedings*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer, 2005.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[33] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[34] Christopher Ré. Snorkel: Beyond hand-labeled data. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[36] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010.

[37] Fahim A Salim, Fasih Haider, Saturnino Luz, and Owen Conlan. Automatic transformation of a video using multimodal information for an engaging exploration experience. *Applied Sciences*, 10(9):3056, 2020.

[38] Fadime Sener, Rishabh Saraf, and Angela Yao. Learning video models from text: Zero-shot anticipation for procedural actions. *CoRR*, abs/2106.03158, 2021.

[39] Fadime Sener, Rishabh Saraf, and Angela Yao. Learning video models from text: Zero-shot anticipation for procedural actions. *CoRR*, abs/2106.03158, 2021.

[40] Ruixiao Sun, Jie Yang, and Mehrdad Yousefzadeh. Improving language generation with sentence coherence objective. *CoRR*, abs/2009.06358, 2020.

[41] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. *CoRR*, abs/1809.00812, 2018.

[42] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7590–7598. AAAI Press, 2018.

[43] Bin Zhu and Chong-Wah Ngo. Cookgan: Causality based text-to-image synthesis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5518–5526. Computer Vision Foundation / IEEE, 2020.

[44] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. MSMO: multimodal summarization with multimodal output. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4154–4164. Association for Computational Linguistics, 2018.

[45] Zulko. Moviepy: A python module for video editing.