Creating Multimedia Summaries Using Tweets and Videos

Anietie Andy,1 Siyi Liu,2 Daphne Ippolito,2 Reno Kriz,2,3 Chris Callison-Burch,2 Derry Wijaya4
1Penn Medicine, 2University of Pennsylvania, 3Human Language Technology Center of Excellence, Johns Hopkins University,
4Boston University

Anietie.Andy@pennmedicine.upenn.edu, {siyiliu,daphnei,ccb}@seas.upenn.edu, rkriz1@jh.edu, wijaya@bu.edu

Abstract
While popular televised events such as presidential debates or TV shows are airing, people provide commentary on them in real-time. In this paper, we propose a simple yet effective approach to combine social media commentary and videos to create a multimedia summary of televised events. Our approach identifies scenes from these events based on spikes of mentions of people involved in the event and automatically selects tweets and frames from the videos that occur during the time period of the spike that talk about and show the people being discussed.

Introduction
Televised events like presidential debates and TV shows capture the attention of vast numbers of people–some of whom tweet or discuss about them in real-time. Using user generated contents–UGCs (in social media or discussion forums data such as Twitter or Reddit) to detect important sequential scenes in a televised event can help with summarizing such events (Andy, Wijaya, and Callison-Burch 2019), gaining insights about the events (Shamma, Kennedy, and Churchill 2009, 2011), or capturing how people react to different parts of an event (Shamma, Kennedy, and Churchill 2011). When such events happen, news media often have analysts or commentators discuss various parts of the event; however, automatically generated summaries from UGCs can be useful to capture how individuals are reacting to the event on social media platforms in real-time. In this paper, we propose a novel, simple yet effective approach that creates multimedia summaries of televised events by selecting social media data (tweets) and video frames of the events that occur in the same timeframe and show the people/characters being mentioned in the tweets. Taken together, they provide a multimedia summary (descriptions and pictures) of important parts of a televised event as shown in Figure 1.

In televised events, a scene is defined as "composing of groups of shots that emphasize a specific concept such as a fixed setting" (Panda, Kuamar, and Chowdhury 2018). Scenes vary depending on the type of event; for example in a presidential debate, an example of a scene is a candidate responding to a question posed by the moderator, however, in a TV show, a scene could be an interaction between two characters. During the broadcast of many televised events, Twitter users create a huge amount of time-stamped and temporally ordered data (Andy, Wijaya, and Callison-Burch 2019; Shamma, Kennedy, and Churchill 2009, 2011; Andy et al. 2017; Nichols, Mahmud, and Drews 2012; Gillani et al. 2017; Andy, Callison-Burch, and Wijaya 2020). We use this tweet-stream data to identify scenes.

Some previous works have developed algorithms to detect scenes from videos related to events (Li and Xu 2020; Vats et al. 2020; Einfalt and Lienhart 2020). In this work, rather than use videos, we are interested in using Twitter data related to events to identify the sequential scenes related to the events. We are interested in this approach because previous works that identified scenes of events from Twitter data related to events focused on identifying the important scenes of these events (Shamma, Kennedy, and Churchill 2009; Andy, Wijaya, and Callison-Burch 2019). However, these events also have scenes that may not be as important. In this work, we aim to identify all the scenes (important or not) in a given event from Twitter data related to the event. In (Andy et al. 2017), it was determined that in tweets (related to an event) published during the time-period of a scene, there is an uptick in the number of mentions of characters involved in the scene that can be used to summarize the event (Huang, Shen, and Li 2018). Taking advantage of this, our approach first identifies scenes by identifying changes in the minute-by-minute frequency of character mentions in tweets. After identifying a time-range for a scene, our approach selects the tweet that best describes the scene and the key frame from

Figure 1: Our approach selects tweets paired with key video frames for a multimedia summary of important events in a TV show. Here are tweets about Game of Thrones and their corresponding top-3 video frames selected by our algorithm.
the video that have faces of the characters mentioned in the scene tweets.

The main contributions of this paper are:

- We develop an approach that uses contemporaneous Twitter commentary about a televised event and the people involved to segment the event into sequential scenes.
- We present a novel approach that uses weak supervision data and incremental learning to effectively recognize faces in video frames. Our method outperforms previous weakly supervised approaches on the task of face recognition. We use this method to select video frames for the identified scenes, thus creating multimedia summaries of the event.
- We collect two new datasets to test our approach. They include video frames and tweets corresponding to night 1 of the first United States (US) Democratic party presidential debates in 2019 and 7 episodes of the Game of Thrones (GoT). We will release these datasets and code upon publication.

Related Work

This paper addresses the task of multimedia summarization of events using tweets and videos. Although previous works have addressed the summarization of videos (Wang et al. 2012, Ajmal et al. 2012, Zhang et al. 2016) and Twitter data (Huang, Shen, and Li 2018) separately, ours is the first to address them together.

Past work determined that televised events often garner considerable attention from the public and that Twitter captures large volumes of discussions and messages related to these events, in real-time (Tumasjan et al. 2010, Starbird and Palen 2012, Paul and Dredze 2011, Sakaki, Okazaki, and Matsuo 2010, Guo, Chang, and Kiciman 2013, Sakaki, Okazaki, and Matsuo 2010, Andy, Callison-Burch, and Williams 2020). To detect important moments of events from social media data, prior work focused on identifying the highlights of these events by using the increase in the volume of published tweets around these events (Nichols, Mahmud, and Drews 2012, Gilliani et al. 2017, Shamma, Kennedy, and Churchill 2009).

The challenge with the moment detection models described in prior work is that some scenes are not as exciting as other scenes and thereby do not attract as many tweets. Our approach is different from the previous work in that it identifies the character mentions per minute and based on the frequency of character mentions, it determines a change in scenes. Even though the scene does not generate a lot of tweets, our approach is able to detect the scene.

Some work has been done to capture visual context (video) as well as dialogue exchanges among multiple speakers (Barbieri et al. 2017) Pasumaru and Bansal 2018 and use Twitter feeds related to live coverage of events to annotate the sentiment of videos associated with the live coverage (Sinha, Choudhury, and Agrawal 2014). Related to our work, there have also been previous approaches on multimodal summarization. In (Zhu et al. 2018, 2020), known news events discussed by news articles and images collected from these news articles are summarized. In (Li et al. 2018), a multimodal method for generating text summaries from asynchronous documents, images, audio recordings, and video of the same news event is proposed. These previous approaches assume the domain of news text, while ours assume the domain of UGCs which are shorter, noisier, often informal, idiomatic and sentiment-laden, which are excellent for summarizing real time reactions about the event; but on which summarization methods trained on clean text such as news have been observed to not work well (Jing, Lopresti, and Shih 2003, Meechan-Maddon 2019). These approaches also assume that the event is known, whereas we assume a live event that is unfolding and consisting of multiple yet unknown scenes that first need to be detected before summarized. Further, previous multimodal approaches summarize from multiple sources and modalities that are all narration of the event. In our case, although the UGCs contain narration of the event, the videos contain dialogues which are part of, but do not narrate the event. There is a bigger semantic gap between the UGCs and videos. For example, transcription of videos in our case cannot be directly used for summaries unlike in (Li et al. 2017), where sentences from video transcriptions are added to the summary to improve informativeness. Lastly, although previous multimodal summarization approaches assume asynchronous sources, the images are still found within news text and the videos are accompanied with narration about the same news event. In our case, there is no such obvious alignment between video and text. Instead, we make use of temporal information for aligning the reaction in tweets with the video being broadcasted at the same time frame. To alleviate the problem of time lag between the video and the reactions in tweets, we also use the change in the frequency of character mentions in tweets with the characters shown in the video to align tweets and video frames automatically and produce a multimodal timeline summary of scenes.

With regards to face recognition, the state-of-the-art in face recognition trains a neural net in a supervised manner to push the embeddings of faces from the same person near to each other, and those of different people far apart (Schroff, Kalenichenko, and Philbin 2015). We use these pre-trained face embeddings as feature vectors to represent faces we extract from video frames. However, different from these supervised methods that employ manually annotated faces, we use textual annotation of the show in the form of subtitles and transcripts to generate weak labels for faces. These have been shown to be useful for identifying character faces in TV shows (Tapaswi, Bäuml, and Stiefelhagen 2015, Miech et al. 2017, Everingham, Stivic, and Zisserman 2009, 2006b). Previous works add other signals such as face tracking (Everingham, Stivic, and Zisserman 2006a, Parkhi, Rahtu, and Zisserman 2015), coreference resolution from video description (Ramanathan et al. 2014), lip movement detection (Tapaswi, Bäuml, and Stiefelhagen 2015), person tracking through face recognition, clothing appearance, speaker recognition, and contextual constraints (Tapaswi, Bäuml, and Stiefelhagen 2012), textual contents of the subtitles (Azab et al. 2018), activity recognition (Bojanowski et al. 2013), and freely available image resources in the web (Nagrani and Zisserman 2018). In our work, we have not used
other information outside of subtitles and transcripts, which we can pursue in future work. Instead, we have concentrated on improving the learning with weak labels using different loss functions and incremental strategies of relabeling ambiguous labels. Contrary from previous works that assume all data from all the episodes to be available at the beginning of training, we perform multiple stage learning where we expose our system to training data from more episodes at each stage.

Many previous works on face recognition with weak/no labels have used EM-like algorithms (Franc and Cech[2017]), clustering algorithms (Sharma et al.[2019]) (we use k-means clustering as one of our baselines), and multi-instance learning (MIL) algorithms that require bags of faces during train and test time (Haarilet et al.[2016]). In our case, these bags of faces are not given during test time. Other work has learned to generate descriptions and jointly ground mentioned characters in videos (Rohrbach et al.[2017]). In contrast, we learn to detect scenes from tweets and describe them using descriptions from tweets and frames from the video.

### Task Definition

The goal in this work is as follows: given a televised event while an event is ongoing

**Our Approach**

In this section, we describe our approach to this task. Our approach consists of 3 stages i.e. (1) identify sequential scenes (2) select tweet indicative of each scene, and (3) select video frames to represent each indicative tweet. Figure[2] shows an overview of our approach.

#### Algorithm 1 Scene identification from Twitter data

**Input:** Tweets, \( T \) (related to a televised event) published while an event is ongoing

**Output:** All sequential scenes in the event from the tweets

1. procedure **SceneDetection**\( (T) \)
2. for tweets in \( T \) published in a minute do
3. create a bin to store these tweets
4. for each bin identify the (%) of each character mentions do
5. if \( (\%) > k \% \) and character not from previous scene i.e., mentioned less than \( m \% \) time in previous scene then
6. mark as start of new scene
7. Return SceneChange(minute of scene change and characters/persons)

#### Stage 1 - Identifying sequential scenes: The first step in our pipeline is to tag mentions of people or characters. In televised events, these are often known prior to the start of the event. Nevertheless tagging known people is still a challenging task, since tweets often refer to characters using nicknames, actors’ names, or other aliases. For each character in an event, we construct an alias list. Our alias lists consist of their first names, last names, and the nickname listed in the first paragraph of the character’s Wikipedia page. For example, the the alias list for the Presidential candidate **Beto O’Rourke** is Robert (his given name), O’Rourke, and Beto (his nickname). The alias list for the GoT character, **Petyr Baelish** is Petyr, Baelish, and Littlefinger.

To identify scenes, our approach collects then groups tweets about an event chronologically into one minute bins. For each character and each bin, the percentage of tweets in the bin that mention the character is calculated. Our scene detection approach (Algorithm[1]) chooses when to create a new scene based on changes in character mentions. A new scene is triggered if a character bin spikes: that is if the fraction of tweets where the character is mentioned in the current minute exceeds a hyperparameter \( k \), while the fraction of tweets mentioning them in the previously determined scene is lower than a hyperparameter \( m \). The values of \( k \) and \( m \) can be optimized on a development set. Our approach outputs the start times of scenes and the character(s) that determined the change. The start time of a new scene also signifies the end time of the previous scene.

#### Stage 2 - Select tweet indicative of each scene: To find the tweet that best describes each scene, we use BERT (Devlin et al.[2018]) to create embeddings for each tweet by averaging its token embeddings. For the period of a predicted scene, we compute the centroid of the tweets and select the tweet whose embedding is closest, in terms of cosine similarity, to the centroid and which contains the names of characters that spiked during the scene. Our approach selects this tweet as the description of the scene.

#### Stage 3 - Select video frames to represent each indicative tweet: Once a tweet is selected to describe a scene, our approach selects frames from videos of the event to visually represent this scene. Since our scene identification is driven by people/character mentions, we select frames from the video of the scene which show the people mentioned in the tweet describing the scene. To do so, we train a model that recognizes faces in the frames; given a face \( x \), the model produces its label \( y \) i.e., the person’s name. We describe our face recognition model in detail in Section Face Recognition Model.

For each identified scene and its tweet description, our approach uses our trained face recognition model to select video frames from that time period that contain faces of the characters mentioned in the tweet.

Similar to how we find the best tweet for each scene, for each character in the scene, we find the centroid of the frames that contain the character mentioned in the tweet. We use ResNet50 (He et al.[2016]) to obtain vector representations of the frames and select the frame that is closest to the centroid and has the highest confidence of containing the character. We also select the frame that contains all the characters in the same way. Intuitively, by selecting the frame that is closest to the centroid and contains the characters of the scene, we will be selecting the frame that is most similar to all the other frames published in that period and that contains the characters involved in the scene. We order the selected frames based on their time stamps and return this sequence of frames to visually represent the scene.

**Face Recognition Model:** Using video subtitles and tran-
scripts, we can label our video frame with weak labels for the faces shown in the frame—that is, the face labels are not fully specified since subtitles and transcripts only give us the information of who is speaking, but not necessarily who appears in the frame or who is who in the case of multiple faces in the frame. For every detected face in the frame, a character’s identity is hence ambiguous: each face is partially labeled with a set of characters speaking in the frame.

In our dataset, \( \sim 76\% \) of faces in GoT and \( \sim 73\% \) of faces in the debate are labeled with multiple names (see Datasets). Our goal is to learn a face recognition model that can refine and disambiguate the labels of the training faces and also generalize to unseen data. In contrast to the standard supervised setting where each training face is labeled with an unambiguous single label, in our dataset each training face \( x \) is labeled with a set of possible labels \( \mathcal{Y} \), only one of which is correct. \( \mathcal{Y} \) is a subset of \( Y_{\text{tot}} \), the set of all possible characters in the video.

In a fully-supervised multiclass setting, where \((x, y)\) pairs are given, we can learn our model parameters \( \theta \) by minimizing the negative log likelihood of \( y \) given the input \( x \) with respect to \( \theta \): 

\[
\mathcal{L}_{\text{sup}}(\theta) = -\log P(y|x; \theta).
\]

In our partially-labeled training scenario, the model has access to \( x \) and \( \mathcal{Y} = \{y_1, y_2, \ldots, y_n\} \).

We experiment with a hard expectation-maximization (EM) algorithm from (Min et al. 2019), which minimizes the (negative) marginal likelihood, while attempting to assign a high probability to only one label:

\[
\mathcal{L}_{\text{hardEM}}(\theta) = -\log \max_{y_i \in \mathcal{Y}} P(y_i|x; \theta)
\]

We also experiment with an average categorical cross entropy (aveCE) loss (Mahajan et al. 2018; Joulin et al. 2016; Cour et al. 2009):

\[
\mathcal{L}_{\text{aveCE}}(\theta) = -\frac{1}{|\mathcal{Y}|} \sum_{y_i \in \mathcal{Y}} \log P(y_i|x; \theta)
\]

where \( P(y_i|x; \theta) \) is a softmax function \( \frac{e^{\theta_i^T x}}{\sum_{y_k \in \mathcal{Y}} e^{\theta_k^T x}} \) and \( \theta_i \) is the parameters of the model for character \( y_i \). We use stochastic gradient descent to learn parameters to minimize the loss.

If the set \( \mathcal{Y} \) contains a single label \( y \), then this loss reduces to the regular multiclass cross entropy loss. However, when \( \mathcal{Y} \) is not a singleton, this loss will drive up the average of the scores of the labels in \( \mathcal{Y} \). Intuitively this will mean that if the score of the correct label is large enough, the other labels in the set do not need to be positive. This loss has been shown to work well for some multi-label classification problems (Mahajan et al. 2018). In this work, we explore its use for our partial-label problem.

We employ different strategies for training our face recognition model: (1) by training the model with all our training data at once or (2) incrementally by sequentially exposing our model to training data from one episode at a time.

In the incremental strategy, we retrain our model after each episode with all the data it has seen so far. We then use the trained model to explicitly disambiguate each face \( x \) in our training data with its current most likely solutions \( y \in \mathcal{Y} \). Specifically, similar to the idea of Prototypical Networks (Snell, Swersky, and Zemel 2017) that has been shown effective for incremental learning, we use the model to compute a centroid (prototype) for each character label: the average embedding of faces assigned by the model to the label; and relabel the faces based on the distance to each prototype. This relabeled data is then used to retrain the model in the next iteration.

For the debate dataset, since we only have one “episode” (night 1 of the first debate), we train our model for several iterations on this “episode” with relabeling.

Datasets

We demonstrate our approach on data from two types of televised events: presidential debates and a TV show. 

Presidential Debate Dataset: The first 2019 US Democratic party Presidential debate was held on two nights. We collected tweets and video of night 1 of the debate. Night 1 of the debate had 10 candidates debating with 5 moderators. Similar to prior works (Andy, Wijaya, and Callison-Burch 2019; Andy, Callison-Burch, and Wijaya 2020), using the Twitter streaming API, we collected over 50,000 timestamped and temporally ordered tweets and re-tweets...
Table 1: Example of New York Times reporters’ live chats we use as our ground truth scenes for the Presidential Debate event.

| Time       | Candidate  | New York Time reporters’ live chats                                                                 |
|------------|------------|-----------------------------------------------------------------------------------------------------|
| 9:11 PM    | Julian Castro | “Castro gets the next question, about the pay gap and what he would do to ensure women are paid fairly. He says he wants to pass the Equal Rights Amendment and gets some loud cheers from the audience.” |
| 9:23 PM    | Beto O’Rourke | “O’Rourke is asked why he no longer wants to abolish private insurance. He explains that if you’re a member of a union, you should be able to keep your health care plan. He says directly that he would not get rid of private insurance.” |

that mentioned the word “debate” during the time periods in which night 1 of the debates aired. For ground truth scenes, we segmented the debate based on when each Presidential candidate spoke. We obtained these from the New York Times (NYT) reporters’ live chats about the debates. These live chats include the timestamp in which each candidate spoke and a brief description of what the candidate said as shown in Table 1. One of the co-authors reviewed the NYT reporters’ live chats and marked the timestamp each candidate started speaking as the beginning of a new scene; the timestamp of a new scene also indicates the end time of the previous scene.

**Game of Thrones Dataset:** For our Game of Thrones (GoT) dataset, we collected tweets for all the 7 episodes of GoT season 7. Each episode lasted for approximately an hour. Similar to prior works (Andy, Wijaya, and Callison-Burch 2019; Andy, Callison-Burch, and Wijaya 2020), we used the Twitter streaming API to collect time-stamped and temporally ordered tweets containing the “#gots7”, a popular hashtag for the show, while each episode aired. We collected over 87,000 tweets (averaging 12,439 per episode). Character names and alias were collected from Wikipedia. For ground-truth scenes, we used the scene timestamps from the Amazon Prime video streaming service DVD chapter summaries. The DVD chapter summaries of GoT also contain a short description of each scene, and an image key frame for each scene.

**Video Frames and Faces:** For each of these datasets, in addition to the Twitter data, we extracted faces for the debate and for each GoT episode. We first sample frames, once per second, from the videos of the debate and of each GoT episode. Then, we extract faces from each frame with the OpenFace library (Amos, Ludwiczuk, and Satyanarayanan 2016), and obtain embeddings of the faces using the pre-trained FaceNet embeddings (Schroff, Kalenichenko, and Philbin 2015). We discard faces from frames that contain more than 5 faces. In total, using OpenFace we obtained 10,318 faces from GoT episodes (averaging 1,474 faces per episode) and 9,979 faces from the debate.

We obtain weak labels for the faces via subtitles (which record what is said and when, but not by whom) and transcripts (which record who says what but not when), following previous work on face recognition in TV-Shows (Everingham, Sivic, and Zisserman 2009). By matching what is said in subtitles and transcripts, we create labels of a face based on who is speaking when the face is shown. However, knowledge that a character is speaking gives only a very weak cue that the character is in the frame. The speaking character may not always be visible and other characters who are not speaking or who speak before or after the character may be shown. Hence, we use all speaking character names in and within 15 seconds window of each second frame as labels. Each face in our dataset may thus be labeled with multiple names. On average, for each episode of GoT, ~76% of the faces are labeled with multiple names (a total of 7,895 faces). For the debate, ~73% of the faces are labeled with multiple names (a total of 7,301 faces).

Since transcripts (who says what) are only available for episode 1-6 of GoT season 7, to obtain weak face labels for the remaining episode i.e., episode 7, we train a deep speaker recognition model (a 3-layer feed forward neural network) on speech from episode 1-6 that are labeled with character names by matching the speech time stamps and when characters are speaking from the subtitles and transcripts of these episodes. Our speaker recognition model achieves 96.17% 10-fold cross validation accuracy for recognizing speaker from speech in training. We use this model to predict who speaks when in episode 7 and use these predictions to obtain weak labels for faces in this episode.

**Evaluation**

Similar to prior work (Bekoulis et al. 2019), a predicted scene is considered correct if the predicted start time is within the timestamp boundary of the ground-truth reference. We compute precision, recall, and F1 scores of our approach for identifying scenes. In the future, to evaluate the scene detection algorithm, we will set a margin of error so if the predicted start time of a scene is between the ground truth start time and the margin of error, the predicted time is considered correct otherwise it is not correct.

We evaluate our face recognition model on our presidential debate and GoT datasets. For the presidential debate dataset, we create a test set with 113 randomly sampled faces, which we manually annotated with 15 labels (the candidates and moderator names). We use the rest of the ~9k faces for training. Similarly, for, GoT we created a test set with 1,079 randomly sampled faces from episode 7, which we manually annotated with 45 labels (the character names). We use the rest of the faces from GoT season 7 (~11k faces) for training. We compute the micro accuracies of our face recognition on these test sets (Section Face Recognition)

1 https://www.NYT.com/interactive/2019/06/26/us/politics-democratic-debate-live-chat.html

2 We use pre-trained speech representation provided by https://www.voicebiometry.org/
Scene Identification Results: From the ground truth data (i.e., NYT reporters’ live chats and Amazon prime video streaming service DVD chapter summaries), there were 47 scenes in the presidential debate and 90 scenes in GoT.

We compare our scene identification approach to two baselines:

**Baseline 1:** This baseline identifies a scene if there is a peak in the volume of tweets and records the start time of the ascent of the peak and the time of the descent of the peak as the start and end times of the scene.

**Baseline 2:** This spike-based baseline uses the moment identification model from Gillani et al. (2017), which uses the mean and standard deviation to determine the threshold for selecting scenes.

Tables 2 and 3 show that our entity-spike approach for identifying scenes outperforms these two baselines for identifying scenes in terms of precision, recall, and F1 scores when evaluated against the ground truth.

### Table 2: Our approach outperforms the baselines for identifying scenes around the Presidential debate on the NYT ground truth scenes

| Approaches | Precision | Recall | F1   |
|------------|-----------|--------|------|
| Our Model  | 0.79      | 0.71   | 0.74 |
| Baseline 1 | 0.77      | 0.55   | 0.64 |
| Baseline 2 | 0.75      | 0.37   | 0.49 |

### Table 3: Our approach outperforms the baselines for identifying scenes of GoT season 7 on the Amazon Prime DVD chapter ground truth scenes

| Approaches | Precision | Recall | F1   |
|------------|-----------|--------|------|
| Our Model  | 0.80      | 0.58   | 0.67 |
| Baseline 1 | 0.77      | 0.35   | 0.48 |
| Baseline 2 | 0.58      | 0.30   | 0.39 |

Multimedia Summary Results: For GoT, we compare our model’s multimedia summary results against the scene descriptions and pictures from Amazon Prime DVD Chapter Summaries and show quantitative and qualitative results. For the presidential debate, since the NYT reporters live chat describes scenes but does not have images of the scenes, we compare our model’s tweet summary to the last NYT reporters’ live chat in each identified scene; the intuition here is that the last blog post in a scene captures the summary of what happened in that scene. Side-by-side comparison of our automatically generated multimedia summaries to Amazon DVD Chapter Summaries and NYT reporters’ live chats are shown in Figure 3.

Given that tweets are more informal and noisier compared to text from news articles or text written by experts, ROUGE scores, which are determined by comparing the overlapping n-grams between a generated summary and a reference summary, is not the best evaluation metric for tweet summaries, hence, we use humans who had watched these events to evaluate our models summaries.

GoT: For each identified scene, we ask 5 Amazon mechanical turk (MTurk) respondents - who had watched GoT, the following questions:

(A) Given the image from the Amazon Prime DVD chapter summary and the image from our predicted multimedia summary for the scene, select the image that best depicts the scene?
Figure 3: Multimedia summaries of GoT and Debate scenes generated automatically by our approach, with compared with human expert and professional summaries of key frames and scene descriptions from Amazon Prime and live chats from NYT respectively.

| MTurk question | % Prefer Our Model | % Prefer Amazon |
|----------------|--------------------|-----------------|
| (A) Image      | 65%                | 35%             |
| (B) Summary    | 30%                | 70%             |
| (D) Image & summary pair | 35% | 65% |

Table 5: Percentage of respondents who prefer our model’s automatically generated multimedia summaries over those of Amazon Prime DVD chapter human curated professional summaries.

Debate: For each identified scene in the presidential debate, similar to the GoT analysis, we ask 5 respondents - who had watched the presidential debate the following questions:

(A) Given the last NYT reporters live chat in a scene as a summary of the scene and the tweet summary from our multimedia model, select the summary that best previews the scene?

(B) State why you selected the summary that you did and what you think about the summary that was not selected (this question was marked as optional)?

For question (A), we select a summary if 3 or more of the respondents selected it. The annotator agreement (using intraclass correlation) was 0.78. Roughly similar to GoT tweet summaries, the respondents prefer our summaries 33% of the time over the NYT reporters’ live chats summaries. Table 7 shows the qualitative results from the presidential debates.

Discussion and Limitations

In this section we discuss the findings from this work and analyze the limitations and failure cases of our approach to inform future works on this task.

From the qualitative results, the recurring response as to why the respondents did not select the tweet summary from our model was that some of the tweet summaries were not detailed/descriptive enough or too opinionated when compared to the summaries from Amazon/NYT. However, the fact that tweets are able to capture “in the moment” feelings and opinions of viewers are also the reason why respondents prefer them over traditional summaries provided by Amazon/NYT.

Figure 3 shows representative examples of the output from our model. For GoT, our approach depicts scenes similarly to Amazon Prime. What’s more, the descriptions from tweets include not only what happen e.g., “The Hound! fighting with Jon!” but also provide viewers’ general feelings and responses e.g., “Arya Stark and Hot Pie ... this is so beautiful”, as well as background information towards what’s happening in the scene e.g., “Jon was like test me Sansa–I dare you”, in the background in which brother and sister,
### Table 6: Qualitative reasons why respondents select a particular summary and why they do not select it: GoT

| Method | Reason for selecting this summary | Reason for not selecting this summary |
|--------|-----------------------------------|--------------------------------------|
| Our model Summary | (1) I selected this summary as it was the most related to the depiction and while not written in a great way, best described the scene | (1) did not choose this summary because it did not provide enough detail on the scene |
| | (2) I selected this summary as it seems to better show the feelings and environment of what is being described | (2) I did not choose this summary as it was not descriptive enough |
| Amazon Professional Summary | (1) In this scene the summary best describes the moment and the pending intensity to come | (1) This summary was too specific to a single moment |
| | (2) This was the most descriptive | (2) This summary not selected portrayed a slightly different aspect of the event. |

### Table 7: Qualitative reasons why respondents select a particular summary.

| Method | Reason for selecting summary |
|--------|-----------------------------|
| Our model Summary | (1) Not opinionated by saying he was showing off |
| NYT journalist/political analyst live chat | (1) It was more detailed |
| | (2) gave more information about the sub-event |

### Limitations and Error Analysis:
Firstly, for our scene identification step, the most common scenes that this approach missed were sequential scenes that involved the same character; this happened 8 times in GoT. This is partly because our approach assumes that the character that determines a new scene will be different from the characters from the previous scene, however, in these events, some adjacent scenes involve the same characters.

Another failure case comes from when our face recognition model predicts the name of the face wrongly, which results in a wrong frame being selected. This happens in 9 of the 76 (∼12%) identified scenes in our GoT dataset. Another error comes from the wrong name being identified in the tweet of the scene: we mistake the Twitter user name as containing a mention of a character, which results in the wrong frame being selected. This happens in 1 of the 76 identified scenes (∼1%) in GoT dataset.

We also observe that if characters appear together in all episodes, our approach can wrongly predict their labels. For example, Lyanna and Robert is wrongly predicted as each other as they only ever appear together in all GoT season 7 episodes. We believe exposing our model to more episodes can help in this case.

Lastly, in this work, we focused on popular events which generate a lot of Twitter data from viewers, hence, the findings from this work may not apply to events with fewer associated Twitter posts.

### Conclusion and Future Work
In conclusion, we propose a simple approach to identify and portray the scenes in televised events. We evaluate our approach on tweets collected around 7 episodes of a TV-show, GoT and night 1 of the first Democratic party presidential debates. To the best of our knowledge, our work is the first to combine knowledge from multiple modalities and platforms of different nature i.e., narration and reaction about the event in Twitter and dialogues in the televised video of the event, to obtain coherent and sequential portrayal of scenes in televised events.

This work uses English Twitter data, however, there are events such as the Olympics that attract users – from various countries and that speak different languages, some of whom publish Twitter data in their respective languages about the event in real-time. In the future, we envision the use of our method for creating real-time multilingual and multimedia summaries of events around the world using videos and social media data. Such multimedia summaries can also be useful for downstream applications such as sentiment analysis of users’ reactions toward events and machine translation (MT) of UGCs. As social media users may talk about the same event similarly in different languages, we can use the multilingual and multimedia summaries of their responses as comparable data, with shared images as additional information, for training future MT models for UGCs.

### References
Ajmal, M.; Ashraf, M. H.; Shakir, M.; Abbas, Y.; and Shah, F. A. 2012. Video summarization: techniques and classifica-
Proceedings of the 3rd
ences by modeling twitch emotes. In
region, H. 2017. Towards the understanding of gaming audi-
Barbieri, F.; Anke, L. E.; Ballesteros, M.; Soler, J.; and Sag-
guage understanding.
Einfalt, M.; and Lienhart, R. 2020. Decoupling video and
human motion: towards practical event detection in athlete
recordings. In Proceedings of the IEEE/CVF Conference on
Computer Vision and Pattern Recognition Workshops, 892–
93.
Everingham, M.; Sivic, J.; and Zisserman, A. 2006a. “Hello!
My name is... Buffy” – Automatic Naming of Characters in
TV Video. In Proceedings of the British Machine Vision
Conference.
Everingham, M.; Sivic, J.; and Zisserman, A. 2006b. Hello!
My name is... Buffy”--Automatic Naming of Characters in
TV Video. In BMVC, volume 2, 6.
Everingham, M.; Sivic, J.; and Zisserman, A. 2009. Taking
the bite out of automated naming of characters in TV video.
Image and Vision Computing 27(5): 545–559.
Franc, V.; and Cech, J. 2017. Learning CNNs for face recog-
nition from weakly annotated images. In Automatic Face &
Gesture Recognition (FG 2017), 2017 12th IEEE Interna-
tional Conference on, 933–940. IEEE.
Gillani, M.; Ilyas, M. U.; Saleh, S.; Alowibdi, J. S.; Aljo-
hani, N.; and Alotaibi, F. S. 2017. Post summarization of
microblogs of sporting events. In Proceedings of the 26th
International Conference on World Wide Web Companion,
59–68. International World Wide Web Conferences Steering
Committee.
Guo, S.; Chang, M.-W.; and Kiciman, E. 2013. To Link or
Not to Link? A Study on End-to-End Tweet Entity Linking.
In HLT-NAACL, 1020–1030.
Haurilet, M.-L.; Tapaswi, M.; Al-Halah, Z.; and Stiefelhagen,
R. 2016. Naming TV characters by watching and analyz-
ing dialogs. In 2016 IEEE Winter Conference on Appli-
cations of Computer Vision (WACV), 1–9. IEEE.
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep resid-
ual learning for image recognition. In Proceedings of the IEEE
conference on computer vision and pattern recognition,
770–778.
Huang, Y.; Shen, C.; and Li, T. 2018. Event summariza-
tion for sports games using twitter streams. World Wide Web
21(3): 609–627.
Jing, H.; Lopresti, D.; and Shih, C. 2003. Summarization of
noisy documents: a pilot study. In Proceedings of the HLT-
NAACL 03 Text Summarization Workshop, 25–32.
Joulin, A.; van der Maaten, L.; and Vasilache, N. 2016.
Learning visual features from large weakly supervised
data. In European Conference on Computer Vision, 67–84.
Springer.
Li, H.; Zhu, J.; Ma, C.; Zhang, J.; and Zong, C. 2017. Multi-
modal summarization for asynchronous collection of text,
image, audio and video. In Proceedings of the 2017 Confer-
ence on Empirical Methods in Natural Language Process-
ing, 1092–1102.
Li, H.; Zhu, J.; Ma, C.; Zhang, J.; and Zong, C. 2018. Read,
write, watch, listen, and summarize: Multi-modal summarization
for asynchronous text, image, audio and video. IEEE Trans-
actions on Knowledge and Data Engineering 31(5): 996–
1009.
Li, P.; and Xu, X. 2020. Recurrent Compressed Convolutional
Networks for Short Video Event Detection. IEEE Access 8:
114162–114171.
Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Puluri,
M.; Li, Y.; Bharame, A.; and van der Maaten, L. 2018. Ex-
ploring the limits of weakly supervised pretraining. In Pro-
cedings of the European Conference on Computer Vision
(ECCV), 181–196.
Meechan-Maddon, A. 2019. The effect of noise in the train-
ing of convolutional neural networks for text summarisation.
Miech, A.; Alayrac, J.-B.; Bojanowski, P.; Laptev, I.; and
Sivic, J. 2017. Learning from video and text via large-scale
discriminative clustering. In 2017 IEEE International Con-
ference on Computer Vision (ICCV), 5267–5276. IEEE.
Min, S.; Chen, D.; Hajishirzi, H.; and Zettlemoyer, L. 2019. A discrete hard em approach for weakly supervised question answering. arXiv preprint arXiv:1909.04849.

Nagrani, A.; and Zisserman, A. 2018. From benedict cumberbatch to sherlock holmes: Character identification in TV series without a script. arXiv preprint arXiv:1801.10442.

Nichols, J.; Mahmud, J.; and Drews, C. 2012. Summarizing sporting events using twitter. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, 189–198. ACM.

Panda, R.; Kuanar, S. K.; and Chowdhury, A. S. 2018. Nyström approximated temporally constrained multisimilarity spectral clustering approach for movie scene detection. IEEE transactions on cybernetics 48(3): 836–847.

Parkhi, O. M.; Rahtu, E.; and Zisserman, A. 2015. It’s in the bag: Stronger supervision for automated face labelling. In ICCV Workshop, volume 2, 6.

Pasunuru, R.; and Bansal, M. 2018. Game-Based Video-Context Dialogue. arXiv preprint arXiv:1809.04560.

Paul, M. J.; and Dredze, M. 2011. You are what you Tweet: Analyzing Twitter for public health. ICwsM 20: 265–272.

Ramanathan, V.; Joulion, A.; Liang, P.; and Fei-Fei, L. 2014. Linking people in videos with “their” names using coreference resolution. In European conference on computer vision, 95–110. Springer.

Rohrbach, A.; Rohrbach, M.; Tang, S.; Joon Oh, S.; and Schiele, B. 2017. Generating descriptions with grounded and co-referenced people. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4979–4989.

Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web, 851–860. ACM.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. CoRR abs/1503.03832. URL http://arxiv.org/abs/1503.03832.

Shamma, D. A.; Kennedy, L.; and Churchill, E. F. 2009. Tweet the debates: understanding community annotation of uncollected sources. In Proceedings of the first SIGMM workshop on Social media, 3–10. ACM.

Shamma, D. A.; Kennedy, L.; and Churchill, E. F. 2011. Peaks and persistence: modeling the shape of microblog conversations. In Proceedings of the ACM 2011 conference on Computer supported cooperative work, 355–358. ACM.

Sharma, V.; Tapaswi, M.; Sarfraz, M. S.; and Stiefelhagen, R. 2019. Self-supervised learning of face representations for video face clustering. arXiv preprint arXiv:1903.01000.

Sinha, P.; Choudhury, A. D.; and Agrawal, A. K. 2014. Sentiment Analysis of Wimbledon Tweets. In # MSM, 51–52. Citeseer.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In Advances in neural information processing systems, 4077–4087.

Starbird, K.; and Palen, L. 2012. (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In Proceedings of the acm 2012 conference on computer supported cooperative work, 7–16. ACM.

Tapaswi, M.; Bäuml, M.; and Stiefelhagen, R. 2012. “Knock! Knock! Who is it?” probabilistic person identification in TV-series. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2658–2665. IEEE.

Tapaswi, M.; Başüm, M.; and Stiefelhagen, R. 2015. Improved weak labels using contextual cues for person identification in videos. In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, volume 1, 1–8. IEEE.

Tumasjan, A.; Sprenger, T. O.; Sandner, P. G.; and Welpe, I. M. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM 10(1): 178–185.

Vats, K.; Fani, M.; Walters, P.; Claudi, D. A.; and Zelek, J. 2020. Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 882–883.

Wang, M.; Hong, R.; Li, G.; Zha, Z.-J.; Yan, S.; and Chua, T.-S. 2012. Event driven web video summarization by tag localization and key-shot identification. IEEE Transactions on Multimedia 14(4): 975–985.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016. Video summarization with long short-term memory. In European conference on computer vision, 766–782. Springer.

Zhu, J.; Li, H.; Liu, T.; Zhou, Y.; Zhang, J.; and Zong, C. 2018. MSMO: Multimodal summarization with multimodal output. In Proceedings of the 2018 conference on empirical methods in natural language processing, 4154–4164.

Zhu, J.; Zhou, Y.; Zhang, J.; Li, H.; Zong, C.; and Li, C. 2020. Multimodal summarization with guidance of multimodal reference. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 9749–9756.