Enriching Video Captions With Contextual Text

Philipp Rimle
ETH Zürich
primle@ethz.ch

Pelin Dogan-Schönberger
ETH Zürich
pelin.dogan@inf.ethz.ch

Markus Gross
ETH Zürich
grossm@inf.ethz.ch

***Abstract*—Understanding video content and generating caption with context is an important and challenging task. Unlike prior methods that typically attempt to generate generic video captions without context, our architecture contextualizes captioning by infusing extracted information from relevant text data. We propose an end-to-end sequence-to-sequence model which generates video captions based on visual input, and mines relevant knowledge such as names and locations from contextual text. In contrast to previous approaches, we do not preprocess the text further, and let the model directly learn to attend over it. Guided by the visual input, the model is able to copy words from the contextual text via a pointer-generator network, allowing to produce more specific video captions. We show competitive performance on the News Video Dataset and, through ablation studies, validate the efficacy of contextual video captioning as well as individual design choices in our model architecture.**

I. Introduction

Understanding video content is a substantial task for many vision applications, such as video indexing/navigation [1], human-robot interaction [2], describing movies for the visually impaired people [3], or procedure generation for instructional videos [4]. There are many difficult challenges due to the open domain and diverse set of objects, actions, and scenes that may be present in the video with complex interactions and fine motion details. Furthermore, the required contextual information may not be present in the concerned video section at all, which needs to be extracted from some other sources.

While significant progress has been made in video captioning, stemming from release of several benchmark datasets [3], [5]–[8] and various neural algorithmic designs, the problem is far from being solved. Most, if not all, existing video captioning approaches can be divided into two sequential stages that perform visual encoding and text decoding respectively [9]. These stages can be coupled further by additional transformations [10], [11] where the models are limited by the input visual content or the vocabulary of a specific dataset. Some approaches [12] consider the preceding or succeeding video clips to extract contextual relation in the visual content to generate coherent sentences in a storytelling way. In general, these approaches focus on a domain specific dataset not reflecting the whole real world, but only a subset that is missing a lot of information needed to produce human comparable results. Consequently, most captions still tend to be generic like "someone is talking to someone" and the knowledge about *who*, *where* and *when* is missing. We try to overcome this issue by providing contextual knowledge in addition to the video representation. This allows us to produce more specific

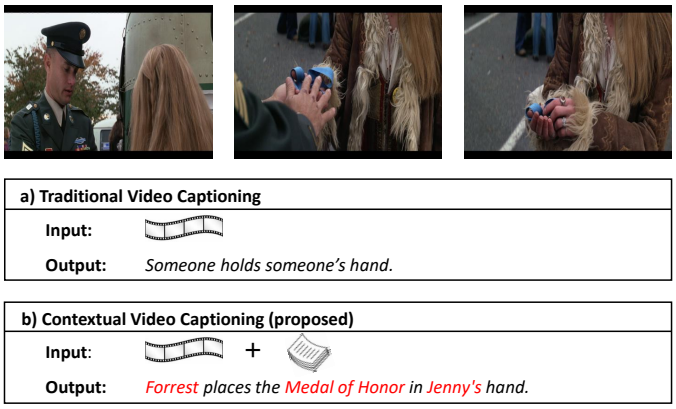

Fig. 1. Captions for a video clip from the movie *Forrest Gump*. a) with traditional methods, b) with contextual video captioning that exploits the movie script as an additional input.

captions like "Forrest places the Medal of Honor in Jenny's hand." instead of just "Someone holds someone's hand." as illustrated in Figure 1.

To address these limitations, we propose an end-to-end differentiable neural architecture for contextual video captioning, which exploits the required contextual information from a relevant contextual text. Our model extends a sequence-to-sequence architecture [9] by employing temporal attention in the visual encoder, and a pointer generator network [13] at the text decoder which allows for extraction of background information from a contextual text input to generate rich contextual captions. The contextual text input can be any text that is relevant to the video up to some degree without strict limitations. This could be a part of the script for a movie section, an article for a news video, or a user manual for a section of an instructional video.

Contributions. The contributions of this paper are three-fold. First, we propose a method for contextual video captioning which learns to attend over the context in raw text and generates out of vocabulary words by copying via pointing. The source code for the full framework will be publicly available[1]. Second, we augment the LSMDC dataset [8] by pairing video sections with the corresponding parts in the movie scripts, and share this new split with the community[2]. Third, we show competitive performance both with respect to the prior

[1]https://github.com/primle/S2VT-Pointer
[2]https://github.com/primle/LSMDC-Context

state-of-the-art and ablation variants of our model. Through ablations we validate the efficacy of contextual captioning as well as individual design choices in our model.

II. Related Work

Our goal of contextual caption generation is related to multiple topics. We briefly review the most relevant literature below.

Unimodal Representations. It has been observed that deep neural networks such as VGG [14], ResNet [15], GoogLeNet [16] and even automatically learned architectures [17], can learn suitable image features to be transferred to various vision tasks [18], [19]. Generic representations for video and text have been receiving considerable attention. Pooling and attention over frame features [20]–[22], neural recurrence between frames and spatiotemporal 3D convolution are among the common video encoding techniques [23]–[25]. On the language side, distributed word representations [26], [27] and recent attention-based architectures [28], [29] provide effective and generalisable representations modeling sentential semantics.

Joint Reasoning of Video and Text. Popular research topics in joint reasoning of image/video and text include video captioning [21], [30], [31], retrieval of visual content [32], [33] and text grounding in images/videos [32], [34]–[36]. Most approaches along these lines can be classified as belonging to either (i) joint language-visual embeddings or (ii) encoder-decoder architectures. The joint vision-language embeddings facilitate image/video or caption/sentence retrieval by learning to embed images/videos and sentences into the same space [31], [37]. The encoder-decoder architectures [43] are similar, but instead attempt to encode images into the embedding space from which a sentence can be decoded [10], [38], [39]. Most of these approaches yield generic video captions without any context due to lack of background knowledge.

Contextual video captioning has not received great attention yet besides few attempts [40]–[42] which might be due to lack of suitable datasets. [43] presents a dataset of news videos and captions that are rich in knowledge elements and employs *Knowledgeaware Video Description Network (KaVD)* that incorporates entities from topically related text documents. Similar to [43], we incorporate relevant text data, with the use of pointer networks [13], for a given video to produce richer and contextual captions. In contrast to KaVD, we propose a model which directly operates on raw contextual text data. Our model learns to attend over the relevant words, based on visual input which allows the model not only to learn contextual entities and events, but also the interaction between them. Further, it allows both video captioning with background knowledge, as well as text summarization based on visual information. We also eliminate the additional preprocessing overhead of name/event discovery and linking systems.

III. Approach

We now present our neural architecture for contextual video captioning. An overview of our model is shown in Figure 2. The input video clip consists of a number of consecutive frames $V = \{V_i\}_{i=1...N}$. The contextual text sequence consists of a number of consecutive words $Z = \{Z_i\}_{i=1...M}$. Our task is to find a function π that encodes the input sequences and decodes a contextual caption as a sequence of consecutive words $Y = \pi(V, Z) = \{Y_i\}_{i=1...L}$. We rely on a sequence-to-sequence architecture to handle variable input and output length. A stack of two LSTM [44] blocks as proposed in [9] is used for both encoding and decoding, which allows parameter sharing between the two stages. The stack consists of a bidirectional and a unidirectional LSTM which are mainly effective in encoding and decoding, respectively. During decoding, the bottom LSTM layer additionally uses a temporal attention over the hidden states of the top LSTM layer to identify the relevant frames. Next to the visual input, a contextual text input of variable size is encoded using another bidirectional LSTM. We use a pointer generator network [13] to attend over the contextual text and build a visual and context aware vocabulary distribution. In addition, the pointer generator network allows us to copy context words directly into the output caption, which enables extracting specific background knowledge not available from only visual input.

A. Encoder-Decoder Network

The baseline architecture consists of two main blocks: a bidirectional LSTM block stacked on top of a unidirectional LSTM block, modeling the input frame and output word sequences, respectively. The top LSTM takes an embedded video feature vector v_t at time step t as input, and passes its hidden state s_t^{top} concatenated with the embedding of the previously predicted word input x_t and a frame context vector s_t^* to the bottom LSTM block:

$$s_t^{top}, c_t^{top} = BiLSTM(v_t, s_{t-1}^{top}, c_{t-1}^{top}) \quad (1)$$

$$s_t^{bottom}, c_t^{bottom} = LSTM([x_t, s_t^{top}, s_t^*], s_{t-1}^{bottom}, c_{t-1}^{bottom}) \quad (2)$$

where c_t^{top}, c_t^{bottom} are the respectively memory cells for the top and bottom LSTM.

The time axis of the stacked LSTMs can be split into the encoding and decoding stage. During encoding, each video frame is passed into a pretrained Convolutional Neural Network (CNN) to obtain frame level features, from where the linear embedding v_t to a lower dimensional space is learned. Since there is no previously predicted word input and no frame context vector during this stage, a padding vector of zeros is used for x_t and s_t^*.

The decoding stage begins after the fixed amount of encoding time steps T, by feeding the *beginning of sentence* (BOS) tag to the model. The BOS tag is used to signal the model to start decoding its latent representation of the video as a sentence. Since there is no video frame input in this stage, a padding vector is passed to the top LSTM. To obtain the frame context vector s_t^* at decoding timestep t, a temporal attention

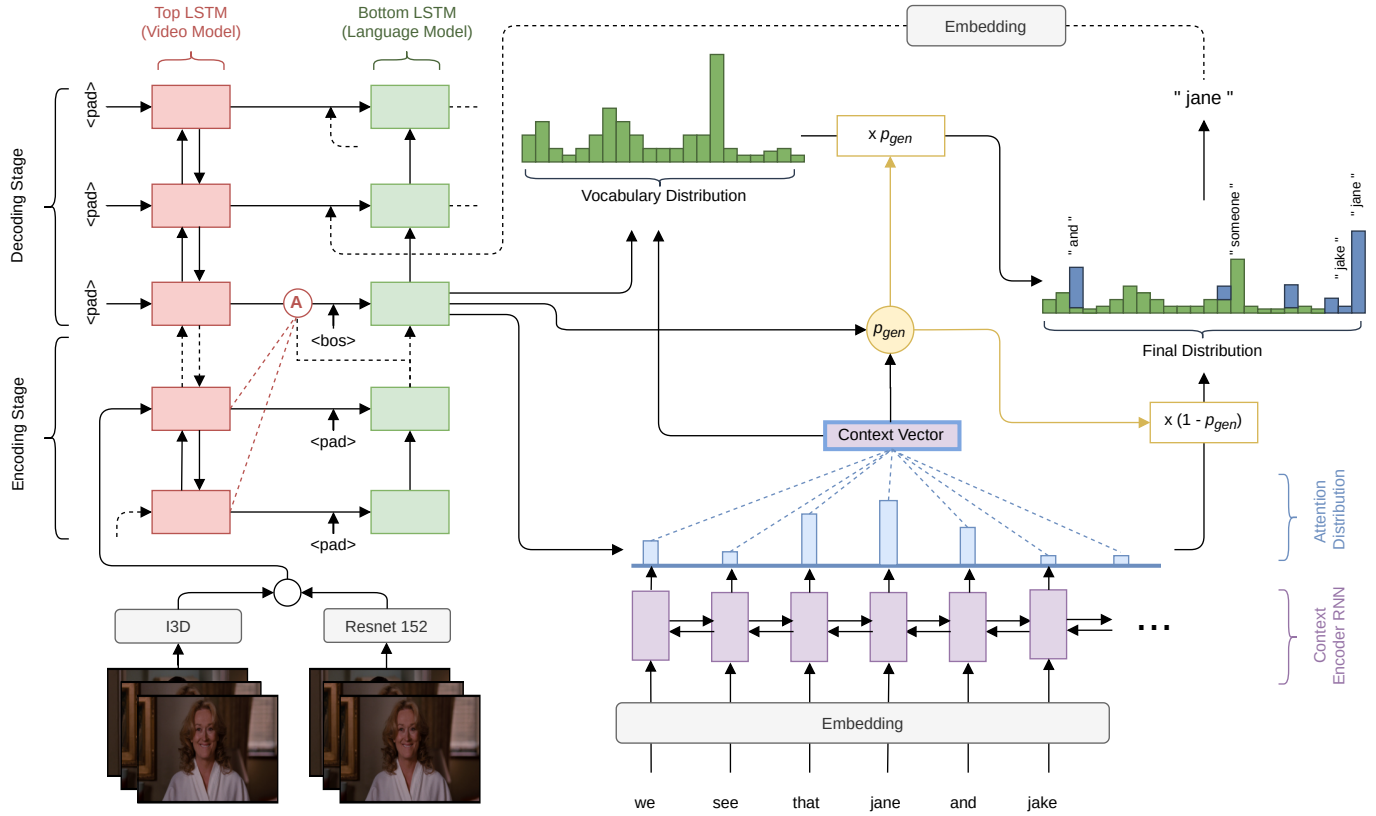

Fig. 2. Model Overview. A stack of two LSTM blocks is used for both encoding (red) and decoding (green) the visual input and textual output respectively. The bottom LSTM (green) layer additionally uses a temporal attention to identify the relevant frames. The contextual text input of variable size is encoded using another bidirectional LSTM to build a visual and context aware vocabulary distribution with the use of a pointer generator network.

with an additive alignment score function [45] over the hidden states of the top LSTM $\{s_0^{top}, \ldots, s_T^{top}\}$ is applied:

$$\alpha_{t,j} = score(s_j^{top}, s_{t-1}^{bottom}) \tag{3}$$

$$\eta_{t,j} = softmax(\alpha_{t,j}) \tag{4}$$

$$s_t^* = \sum_j \eta_{t,j} s_j^{top} \tag{5}$$

The output of the bottom LSTM s_t^{bottom} is then passed to the pointer generator network, generating the output word y_t. During the encoding stage, no loss is computed and the output of the LSTM is not passed to the pointer generator network.

B. Pointer Generator Network

We use a bidirectional LSTM to learn a representation of the contextual text. At each context encoder timestep i, the embedded word z_i is passed to the LSTM layer, producing a sequence of context encoder hidden states h_i. These hidden states are used to build a soft attention distribution ξ_t over the context word representations per decoder timestep t, similar to [46]:

$$\beta_{t,i} = u^T tanh(W_h h_i + W_s s_t^{bottom} + b_{attn}) \tag{6}$$

$$\xi_{t,i} = softmax(\beta_{t,i}) \tag{7}$$

where u, W_h, W_s and b_{attn} are learned parameters. To overcome the general issue of tendency to produce repetition in sequence to sequence models, [13], [47] proposed a coverage model, which keeps track of the attention history. At each decoder timestep t, we follow the same procedure by introducing a coverage vector c_t, which is the sum of the previous attention distributions:

$$c_t = \sum_{t'=0}^{t-1} \xi_{t'} \tag{8}$$

This vector informs the model about the degree of attention that the context words have received so far, and helps the model not to attend over the same words repeatedly. The coverage vector is fed to the pointer-generator network as an additional input, and the attention score calculation from Equation 6 is modified as:

$$\beta_{t,i} = u^T tanh(W_h h_i + W_s s_t^{bottom} + w_c c_{t,i} + b_{attn}) \tag{9}$$

where w_c is a learned parameter vector of the same shape as u. The resulting context vector h_t^*, computed as

$$h_t^* = \sum_i \xi_{t,i} h_i \tag{10}$$

is then concatenated with the decoder hidden state and passed to two fully connected linear layers to produce the vocabulary output distribution P_{vocab}:

$$P_{vocab} = softmax(W'(W[s_t^{bottom}, h_t^*] + b) + b') \quad (11)$$

where W, W', b and b' are learned parameters. At each decoder timestep t, we additionally calculate a generation probability p_{gen}, as proposed in [13], based on the context vector h_t^*, the decoder hidden state s_t^{bottom}, and the embedded decoder word input x_t:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t^{bottom} + w_x^T x_t + b_{ptr}) \quad (12)$$

where σ is the sigmoid function and the vectors w_{h^*}, w_s, w_x and the scalar b_{ptr} are learned parameters. The generation probability is used to weight the vocabulary distribution P_{vocab} and the attention distribution ξ_t at timestep t. For a word y, the final distribution is given as:

$$P(y) = p_{gen} P_{vocab}(y) + (1 - p_{gen}) \sum_{i:z_i=y} \xi_{t,i} \quad (13)$$

Note that if a word y is not in the contextual text, $\sum_{i:z_i=y} \xi_{t,i}$ is zero, and similarly if y is not in the global vocabulary, $P_{vocab}(y)$ is zero.

The loss function per decoder timestep t is given as:

$$loss = -logP(y_t^*) + \lambda \sum_i min(\xi_{t,i}, c_{t,i}) \quad (14)$$

where y_t^* is the target word and λ is a parameter of the model to weight the additional coverage loss [13], used to penalize attending over the same contextual word representation multiple times.

As the coverage mechanism penalizes repeated attention on the contextual text, but not on the global vocabulary, we introduce an additional penalization at inference time. At timestep t, the output probability $P(y)$ of a word y is multiplied by the factor $\beta \in [0, 1]$, if it already occurs in the predicted sentence $y_0 \dots y_{t-1}$.

IV. Datasets

We test our approach on two datasets that provide both video and contextual text input.

A. News Video

To the best of our knowledge, the News Video Dataset [43] is the only publicly available dataset consisting both visual and contextual background information for video captioning. The dataset is composed of 2883 news videos from the AFP YouTube channel[3] with the given descriptions as groundtruth captions. The videos cover a variety of topics such as protests, attacks, natural disasters and political movements from October, 2015 to November, 2017. Furthermore, the authors retrieved topically related news documents using the video meta-data tags. The official release comes with the video URLs only. However, upon our request, they kindly shared their collected news articles with us, which we use as an contextual text input in our experiments.

[3]https://www.youtube.com/user/AFP

B. LSMDC-Context

The Large Scale Movie Description Challenge (LSMDC) dataset [8] is a combination of the MPII-MD [3] and the M-VAD [6] datasets, consisting of a large set of video clips taken from Hollywood movies with paired audio description (AD) sentences as groundtruth captions. Sentences in the original AD are filtered and manually aligned to the corresponding portions for a better precision by the authors. The released dataset comes with the original captions as well as a pre-processed version, where all the character names were replaced with *someone* or *people*. The latter version is also the most used in related research and benchmarks as the character names come from the movie context rather than the visual input.

To adapt it to our problem, we augmented the LSMDC dataset with additional contextual text by using publicly available movie scripts. The scripts were downloaded from the Internet Movie Script database[4], parsed in a similar way to the public code of Adrien Luxey[5]. The extracted text from the scripts were stored in a location-scene structure and later used to narrow down the contextual text input while generating a caption for a short video clip within the movie. Next, we downloaded the movie subtitles[6] and built a coarse <*script scene, video time*> mapping using the dialogues in scripts. Note that public movie scripts are rare and can be either a draft, a final, or a shooting version. Therefore the stage directions and especially the dialogues may differ from the subtitles a lot. To overcome this issue, we built the mapping in multiple rounds and eliminated the movies and scripts which do not have sufficient correspondences between the video and the script. In the end, we assign a coarse time interval from the movie for each scene in the script which could be used as contextual text input.

1) AD-Captions with Context: Roughly 40 movies from the LSMDC dataset with AD sentences have an available movie script in the form of a draft, a shooting or a final version. In the first step, we analyzed how many words of the AD-captions can be recovered by the provided movie script context. In the second step, we removed the movies with an average caption/context overlap less than 33.3% to create a smaller split with better context richness. This way we can improve the average overlap by 7% in trade of a smaller but higher quality dataset. The resulting dataset contains 23 movies with a total of $14'464$ video clips. As one expects, experiments have shown that keeping the movies with almost no useful additional context is rather obstructive than helpful in the training process.

2) Script-Captions with Context: A part of LSMDC dataset is composed of movies that are paired with script sentences as groundtruth captions, instead of AD sentences. We used this split as our *toyset* to see how well our model can recover a caption when the ground truth caption is in the contextual

[4]https://www.imsdb.com

[5]https://github.com/Adrien-Luxey/Da-Fonky-Movie-Script-Parser

[6]https://subscene.com/

TABLE I
SUMMARY STATISTICS OF THE DATASETS

Dataset	Domain	# Videos	# Clips	Avg. Duration	# Sentences	Vocab Size
MPII-MD [3]	Movie	94	68'337	3.9s	68'375	21'700
LSMDC [8]	Movie	202	118'114	4.8s	118'081	23'442
News Video [43]	News	–	2'883	52.5s	3'302	9'179
LSMDC*	Movie	177	114'039	4.1	114'039	25'204
LSMDC-Context-AD	Movie	23	14'464	4.2	14'464	8'162
LSMDC-Context-Script	Movie	26	17'954	3.9	17'954	11'997

text. We select the movies with an available movie script and filter out the movies with a caption/context overlap less than 90%. The resulting dataset contains 26 movies with a total of $17'954$ video clips.

3) LSMDC:* The splits above cover a small percentage of the original LSMDC dataset. We denote the bigger split of remaining movies as LSMDC*, which is to be used to pretrain the encoder-decoder network (without contextual text input) to apply transfer learning for the relatively small splits. The new split contains all video-sentence pairs from the original dataset except the test set, since the groundtruth sentences are not available for the original test set.

For these three splits, we created our own training, test, and validation sets considering the number of clips per movie as well as the movie genres. Table I shows the statistics of the datasets used.

V. EXPERIMENTS

A. Video and Text Representation

In all experiments, text data is lower-cased and tokenized into words. For the News Video Dataset, numbers, dates and times are replaced with special tokens following [43]. A vocabulary is built for each respective dataset and clipped by taking account of the occurrence frequency of words. Each word is mapped to an index and the text input to our model is represented as one-hot vector. Further, we use a pretrained Word2Vec model [26], [48], trained on a subset of Google News dataset[7], to have good initialization to our word embedding layer.

We perform video representation differently depending on the used dataset due to different content and style of videos.

1) News Video Dataset: For each video clip, we sample one frame per second, as the video clips from the News Video Dataset are longer (up to two minutes) and short-term temporal information is less significant due to the news video style of rapid scene changes. All frames (RGB images) are smoothed with a Gaussian filter before down-scaling to the size of 224×224, to avoid aliasing artifacts. The preprocessed frames are fed into the VGG-16 [49] pretrained on the ImageNet dataset [50], and the output of the second dense layer (fc2-layer, after applying the ReLU non-linearity and before the softmax layer) is fed into the top LSTM of our model.

[7]https://code.google.com/archive/p/word2vec

2) LSMDC Dataset: The Large Scale Movie Description Challenge published precomputed video features, which we directly use in all our experiments. They provide two types of features: the output of ResNet-152 [51] pretrained on ImageNet [50] before applying softmax, and the output of the I3D model [52] pretrained on ImageNet and Kinetics [53]. I3D makes use of multiple frames and optical flow using 3D CNN, therefore a single feature vector input to our LSTM captures a segment of multiple frames. The concatenation of the two feature vectors is fed into the top LSTM of our model.

B. Training Setup

In all our experiments, the video features and text (word) inputs are embedded into a 500-dimensional and 300-dimensional space respectively. The LSTMs in the encoder-decoder network have a hidden state size of 512, and the LSTM block used to encode the contextual text in the pointer generator network has a hidden state size of 256. During training, dropout [54] rate of 0.5 is applied on the video feature input, embedded word input, embedded context input, and all LSTM outputs. The training is performed with the Adam [55] optimizer using a learning rate of 10^{-4}.

1) News Video Dataset: We unroll the stacked LSTMs to 120 timesteps: 60 for video encoding and 60 for caption decoding. Note that the News Video Dataset contains longer reference captions than the LSMDC* dataset and mostly includes several sub sentences. Further, we unroll the LSTM for the contextual text to a fixed size of 400 timesteps, following [13]. Articles are sentence-wise cropped at the end to fit the maximum length of 400 tokens. For video clips with multiple articles, we create a sample per article and train on all of them. During inference, we take the prediction of the sample/article pair with the highest probability (i.e. most confident). To use transfer learning in some experiments, the complete News Video vocabulary and the most frequent words of the CNN/Daily Mail dataset [46], [56] were combined together and cropped at $20'000$. We first train the pointer-generator network on the bigger CNN/Daily Mail Dataset, and the sequence-to-sequence model on the News Video Dataset. Secondly, we combine the pretrained models and train on the News Video Dataset. For the final model, we use a coverage loss weight λ of 0.2. At inference time, we use beamsearch with a beamwidth of 2 and a repetition penalization β of 0.2.

2) LSMDC-Context Dataset: We unroll the stacked LSTMs to 40 timesteps: 10 for video encoding and 30 for caption

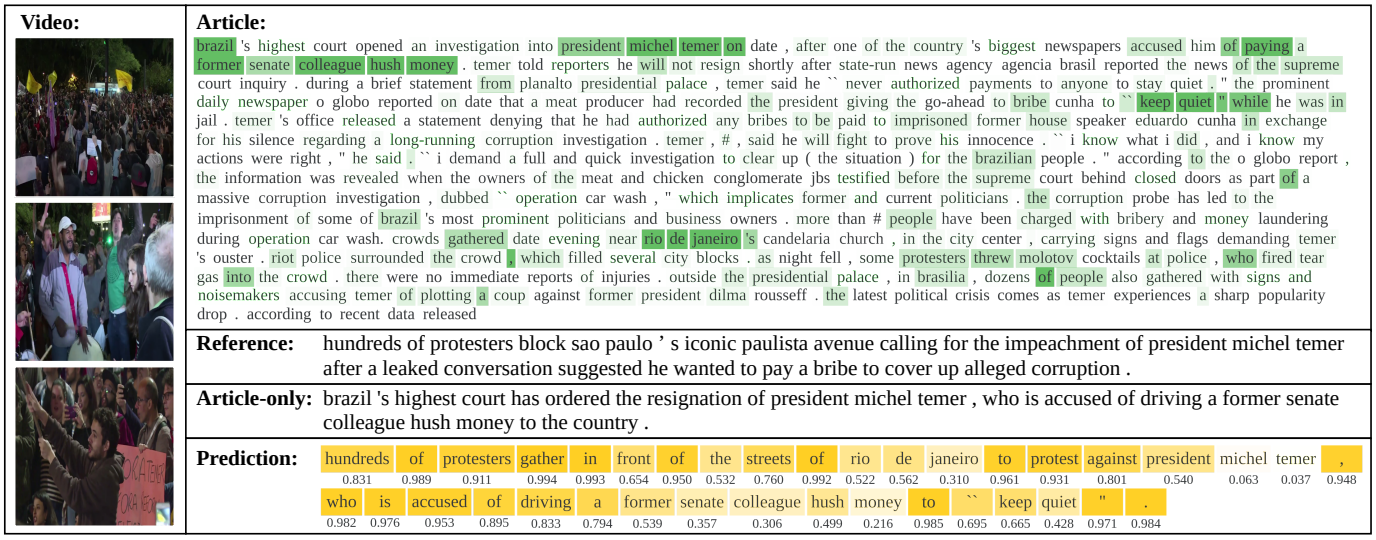

Fig. 3. Sample Prediction on News Video Dataset. Article: the green shading represents the final value of the coverage vector (the sum of the attention distribution for each timestep). A more intense green corresponds with a higher coverage value. Prediction: the yellow shading and the number below represent the generation probability p_{gen}.

decoding. Further, we unroll the LSTM for the contextual text to a fixed size of 400 timesteps for AD-captions and 600 timesteps for script-captions. Movie script scenes are cropped sentence-wise from the beginning and end, to fit the maximum length of tokens. The complete LSMDC* vocabulary is used for the final models that are trained on LSMDC-Context splits. We first train the sequence-to-sequence model on the bigger LSMDC* dataset with *someone-captions*. Next, we fix the weights of the top LSTM (modeling the video), while training on LSMDC-Context-AD (LSMDC-Context-Script, respectively) with *name-captions*. This procedure provides a good initialization for the pointer-generator network. In a last step, we release all the weights and train the full framework end-to-end. Coverage loss weight $\lambda = 1.0$ is used in the final models. At inference time, we do not use beamsearch (i.e. beamwidth of 1), but a repetition penalization β of 0.2.

C. Evaluation

We use METEOR [57] as our quantitative evaluation metric. It is based on the harmonic mean of unigram precision and recall scores, and considers how well the predicted and the reference sentences are aligned. METEOR improves the shortcomings of BLEU [58] and makes use of semantic matching like stemmed word matches, synonym matches and paraphrase matches next to exact word matches. In all experiments, we use METEOR 1.5[8] as done in [9].

D. Results and Analysis

We report the performance of our model on the News Video Dataset in Table II. In order to understand the benefits of the individual components of our model, we also present an ablation study where blocks stacks are removed. Our full model performs significantly better than the video-only and the article-only model which are missing the pointer generator network and the video encoder respectively. Comparing the results between KaVD [43] and our full model is difficult as the authors of KaVD and News Video Dataset only published the ratio of train, validation and test splits, but not the exact sets. The authors did not report the CIDEr score in [43].

[8]http://www.cs.cmu.edu/~alavie/METEOR

TABLE II
PERFORMANCE EVALUATION ON THE NEWS VIDEO DATASET.

Model	METEOR [%]	ROUGE-L [%]	CIDEr [%]
KaVD [43]	10.2	**18.9**	–
Video-only	7.1	16.4	10.2
Article-only	9.3	17.2	20.5
S2VT-Pointer	**10.8**	18.6	**25.7**

We show a qualitative result in Figure 3 to highlight the capabilities of our model which presents a semantically correct summary of the article based on the visual input. While the article focuses on the *hush money investigation*, the model correctly uses this information to augment the visual caption of *protesters doing a demonstration in a street*. This can be seen in the weighting (p_{gen}) of the attention distribution and the global vocabulary distribution: words related to the event of *protesting* are taken from the global vocabulary and entities like *rio de janeiro* or *michel temer*, as well as additional information are successfully extracted from the article.

The performance of our model on LSMDC-Context-AD is shown in Table III. The model is able to recover 37.4% of the character names on average. Figure 4 shows an example where the model correctly extracts the name and scene location from the movie script. The difference between the predicted caption (visually correct) and the groundtruth caption shows the difficulties of the LSMDC dataset in general. Analyzing some example prediction shows that the model occasionally

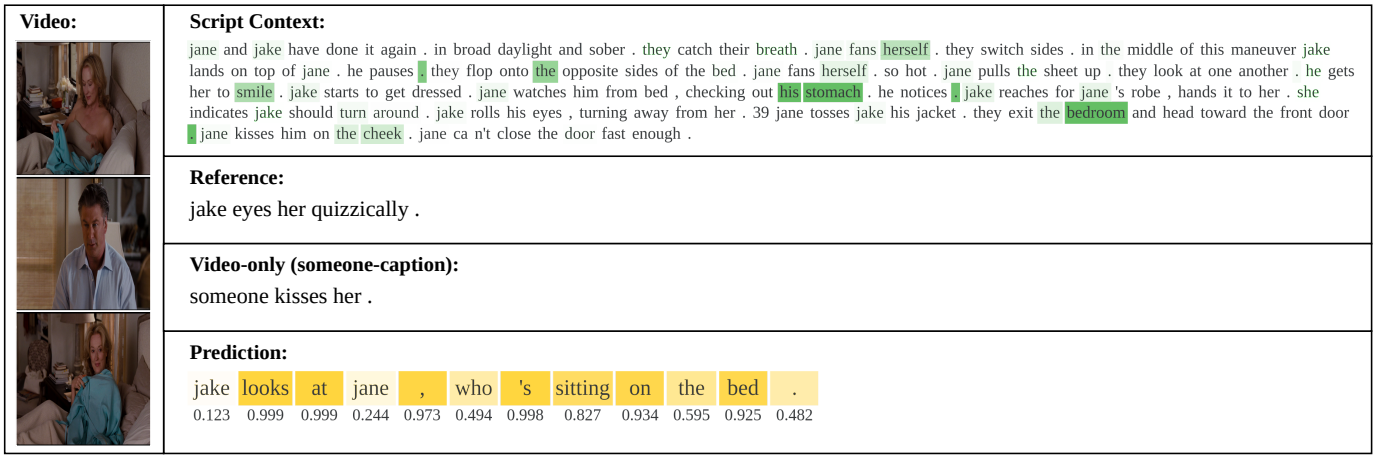

Fig. 4. Sample Prediction on LSMDC-Context-AD. Article: the green shading represents the final value of the coverage vector (the sum of the attention distribution for each timestep). A more intense green corresponds with a higher coverage value. Prediction: the yellow shading and the number below represent the generation probability p_{gen}

TABLE III
PERFORMANCE EVALUATION ON THE LSMDC-CONTEXT-AD DATASET.

Model	Name-Recovery	METEOR	ROUGE-L	CIDEr
Video-only	–	3.4	10.1	5.2
S2VT-Pointer	**37.4**	**5.8**	**14.0**	**15.3**

TABLE IV
PERFORMANCE EVALUATION ON THE LSMDC-CONTEXT-SCRIPT DATASET.

Model	Name-Recovery	METEOR	ROUGE-L	CIDEr
Video-only	–	3.5	10.1	4.7
S2VT-Pointer	**60.0**	**13.8**	**25.3**	**13.4**

substitutes *someone* with a wrong character name. There are many reasons for this behaviour. Firstly, the movie script context does not necessarily include the video scene, nor the character name. Secondly, the dataset is too small and does not let the model learn a good context model at the pointer generator network. In contrast to experiments on the News Video Dataset that are pretrained on CNN/Daily Mail Dataset, the pointer generator network is missing a good initialization due to lack of larger text corpora with similar content and style for the experiments on LSMDC-Context-AD.

Table IV shows the performance on LSMDC-Context-Script. The model is able to learn the mapping between the video and the groundtruth caption that is mostly available in the contextual text. Analyzing some example predictions has shown the issue of the script based captions and why the scores remain relatively low. In LSMDC, consecutive samples tend to have almost identical visual input. Yet, the reference sentences describe different levels of scene details (e.g. *lester, carolyn and jane are eating dinner by candlelight* vs. *red roses are bunched in a vase at the center of the table*). Without the awareness of the sequence of samples, a correct mapping between the script sentences and the reference sentences is ambiguous. This is because a reasonable system would always go for the most likely sentence.

As the ground truth captions from the LSMDC-Context splits highly depend on the respective video clip, we omit the results of the Movie-Script-only model. In contrast to the News Video Dataset, the captions do not reflect a possible summary of the text input and therefore the results are uninformative.

VI. CONCLUSION

In this paper, we proposed an end-to-end trainable contextual video captioning method that can extract relevant contextual information from a supplementary contextual text input. Extending a sequence-to-sequence model with a pointer generator network, our model attends over the relevant background knowledge and copy corresponding vocabulary from the given text input. Results on the News Video Dataset and LSMDC-Context validate the competitive performance of our model which directly operates on the raw contextual text data without the need of additional tools unlike prior methods. Furthermore, we make the source code of our framework and LSMDC-Context publicly available for other researchers. The performance of the presented method is naturally limited by the level of correspondence between the video and the chosen contextual text. In future, we plan to involve multiple contextual resources to extract the relevant contextual information with more confidence and precision.

REFERENCES

[1] O. Marques and B. Furht, *Content-based image and video retrieval*. Springer Science & Business Media, 2002, vol. 21.
[2] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *CVPR*, vol. 2, 2004.
[3] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *CVPR*, 2015.
[4] L. Zhou, C. Xu, and J. J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. ACL, 2011.

[6] A. Torabi, C. Pal, H. Larochelle, and A. Courville, "Using descriptive video services to create a large data source for video annotation research," *arXiv preprint arXiv:1503.01070*, 2015.

[7] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *ICCV*, 2013.

[8] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele, "Movie description," *IJCV*, 2017.

[9] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – video to text," in *ICCV*, 2015.

[10] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical lstms with adaptive attention for visual captioning," *PAMI*, 2019.

[11] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *CVPR*, 2018, pp. 7190–7198.

[12] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *German conference on pattern recognition*, 2014.

[13] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[17] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018.

[18] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NeurIPS*, 2014.

[20] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[21] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.

[22] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, 2015.

[23] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[24] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014.

[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[30] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[31] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *CVPR*, 2016.

[32] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual semantic search: Retrieving videos via complex textual queries," in *CVPR*, 2014.

[33] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *ICCV*, 2017.

[34] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[35] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *ECCV*, 2016.

[36] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *ICCV*, 2017.

[37] B. Xu, Y. Fu, Y.-G. Jiang, B. Li, and L. Sigal, "Heterogeneous knowledge transfer in video emotion recognition, attribution and summarization," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, 2016.

[38] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *ICCV*, 2015.

[39] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *CVPR*, 2016.

[40] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, "Improving lstm-based video description with linguistic knowledge mined from text," *arXiv preprint arXiv:1604.01729*, 2016.

[41] C. Chunseong Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," in *CVPR*, 2017.

[42] V.-K. Tran and L.-M. Nguyen, "Natural language generation for spoken dialogue system using rnn encoder-decoder networks," *arXiv preprint arXiv:1706.00139*, 2017.

[43] S. Whitehead, H. Ji, M. Bansal, S.-F. Chang, and C. Voss, "Incorporating background knowledge into video description generation," in *EMNLP*, 2018.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[46] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[47] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," *arXiv preprint arXiv:1601.04811*, 2016.

[48] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NeurIPS*, 2013.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[52] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[53] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, 2014.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *NeurIPS*, 2015.

[57] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation*, 2014.

[58] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. ACL, 2002.