End-to-end Concept Word Detection
for Video Captioning, Retrieval, and Question Answering

Youngjae Yu Hyungjin Ko Jongwook Choi Gunhee Kim
Department of Computer Science and Engineering
Seoul National University, Seoul, Korea
{yj.yu, hj.ko}@vision.snu.ac.kr, {wookayin, gunhee}@snu.ac.kr

Abstract

We propose a high-level concept word detector that can be integrated with any video-to-language models. It takes a video as input and generates a list of concept words as useful semantic priors for language generation models. The proposed word detector has two important properties. First, it does not require any external knowledge sources for training. Second, the proposed word detector is trainable in an end-to-end manner jointly with any video-to-language models. To maximize the values of detected words, we also develop a semantic attention mechanism that selectively focuses on the detected concept words and fuse them with the word encoding and decoding in the language model. In order to demonstrate that the proposed approach indeed improves the performance of multiple video-to-language tasks, we participate in four tasks of LSMDC 2016 [20]. Our approach achieves the best accuracies in three of them, including fill-in-the-blank, multiple-choice test, and movie retrieval. We also attain comparable performance for the other task, movie description.

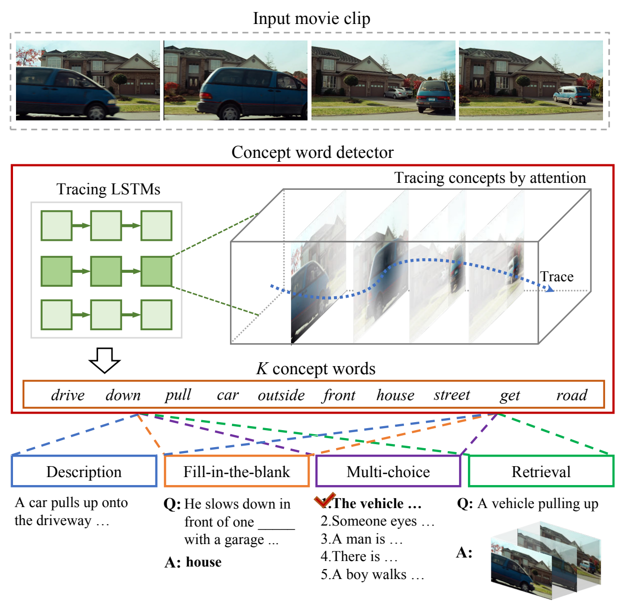


Figure 1. The key idea of the proposed concept word detector. Given a video clip, we use a set of tracing LSTMs to extract multiple concept words that consistently appear across frame regions. We then employ semantic attention to combine visual features with the detected concepts in the LSTM networks for several video-to-language tasks of LSMDC 2016, such as captioning, retrieval, and question answering.

1. Introduction

Video-to-language tasks, including video captioning [6, 11, 19, 29, 34, 37] and video question answering (QA) [25], are recent emerging challenges in computer vision research. This set of problems is interesting as one of frontiers in artificial intelligence; beyond that, it can also potentiate multiple practical applications, such as retrieving video content by users' free-form queries or helping visually impaired people understand the visual content. Recently, a number of large-scale datasets have been introduced as a common ground for researchers to promote the progress of video-to-language research (*e.g*. [4, 18, 20, 25]).

The objective of this work is to propose a *concept word detector*, as shown in Fig.1, which takes a video as input and generates a list of high-level concept words as useful semantic priors for a variety of video-to-language tasks, including video captioning, retrieval, and question answering. We design our word detector to have the following two characteristics, to be easily integrated with any video-to-language models. First, it does not require any external knowledge sources for training. Instead, our detector learns the correlation between words in the captions and video regions from the whole training data. To this end, we use continuous soft attention mechanism that traces consistent visual information across frames and associates them with concept words from captions. Second, the proposed word detector is trainable in an end-to-end manner jointly with any video-to-language models. The loss function for learning the word detector can be plugged as an auxiliary term into the model's overall cost function; as a result, we can reduce efforts to separately collect training examples and learn both models.

We also develop language model components to maximize the values of detected words. Inspired by *semantic attention* in image captioning research [36], we develop an attention mechanism that selectively focuses on the detected concept words and fuse them with the word encoding and decoding in the language model. That is, the detected concept words are combined with input words to better represent the hidden states of encoders, and with output words to generate more accurate word prediction.

In order to demonstrate that the proposed word detector and attention mechanism indeed improve the performance of multiple video-to-language tasks, we participate in four tasks of LSMDC 2016 [20]. The LSMDC (*Large Scale Movie Description Challenge*) is one of the most active and successful challenge series that advances the progress of video-to-language research. The challenges include *movie description* and *multi-choice test* as video captioning, *fill-in-the-blank* as video question answering, and *movie retrieval* as video retrieval. Following the evaluation protocol of LSMDC 2016 exactly, our approach achieves the best accuracies in the three tasks (*fill-in-the-blank*, *multiple-choice test*, and *movie retrieval*), and comparable performance in the other task (*movie description*).

1.1. Related Work

Our approach can be uniquely positioned in the context of recent two research directions in image/video captioning as follows.

Image/Video Captioning with Word detection. Image and video captioning has been actively studied in recent vision and language research, including [5, 6, 11, 19, 21, 29, 30], to name a few. Among them, there have been several attempts to detect a set of concept words or attributes from visual input to boost up the captioning performance. In image captioning research, Fang *et al*. [7] exploit an multiple instance learning (MIL) approach to train visual detectors that identify a set of words with bounding boxed regions of an image. Based on the detected words, they retrieve and re-rank the best caption sentence for the image. Wu *et al*. [31] use a CNN to learn a mapping between an image and semantic attributes. They then exploit the mapping as an input to the language decoder for captioning the image. They also extend the framework to explicitly leverage external knowledge base such as DBpedia for question answering tasks. Venugopalan *et al*. [28] generate description with novel words beyond the ones in the training set, by leveraging external sources, including object recognition datasets like ImageNet and external text corpus like Wikipedia. You *et al*. [36] also exploit weak labels and tags on Internet images or train additional parametric visual classifiers for image captioning.

In the video domain, it is more ambiguous to learn the relation between descriptive words and visual patterns. There have been only few work in video captioning; Rohrbach *et al*. [19] propose a two-step approach for video captioning on the LSMDC dataset. They first extract verbs, objects, and places from movie description, and separately train SVM-based classifiers for each group. They then learn the LSTM-based decoder that generates text description based on the responses of these visual classifiers.

While almost all previous captioning methods exploit external classifiers for concept or attribute detection, the novelty of our work lies in that we use only captioning training data with no external sources to learn the word detector, and propose an end-to-end design for learning both word detection and caption generation simultaneously. Moreover, compared to video captioning work of [19] where only *movie description* is addressed, this work is more comprehensive in that we validate the usefulness of our method for all the four tasks of LSMDC.

Attention for Captioning. Attention mechanism has been successfully applied to caption generation. One of the earliest works is [33] that selectively focuses on different image regions to produce an output word sequence. Later this soft attention has been extended as temporal attention over video frames [35, 37] for video captioning.

Beyond the attention on spatial or temporal structure of visual input, recently You *et al*. [36] propose an attention on attribute words for image captioning. That is, the method enumerates a set of important object labels in the image, and then dynamically switch attention among these concept labels. Although our approach also exploit the idea of semantic attention, it bears two key differences. First, we extend the semantic attention to video domains for the first time, not only for video captioning but also for retrieval and question answering tasks. Second, the approach of [36] relies on the classifiers that are separately learned from external datasets, whereas our approach is learnable end-to-end with only training data of captioning. It significantly reduces efforts to prepare for additional multi-label classifiers.

1.2. Contributions

We summarize contributions of this paper as follows.

(1) We propose a novel end-to-end learning approach for detecting a list of concept words and attend on them to enhance the performance of multiple video-to-language tasks. The proposed concept word detection and attention model can be plugged into different models of video captioning, retrieval, and question answering. Our technical novelties can be seen from two recent trends of image/video captioning research. First, our work is a first end-to-end trainable model not only for concept word detection but also for lan-

guage generation. Second, our work is a first semantic attention model for video-to-language tasks.

(2) To validate the applicability of the proposed approach, we participate in all the four tasks of LSMDC 2016. Our models achieve the best accuracies in three of them, including *fill-in-the-blank*, *multiple-choice test*, and *movie retrieval*. We also attain comparable performance for the other task *movie description*.

2. Detection of Concept Words from Videos

We first explain the pre-processing steps for representation of words and video frames. Then, we explain how we detect concept words for a given video.

2.1. Preprocessing

Dictionary and Word Embedding. We define a vocabulary dictionary $\mathcal{V}$ by collecting the words that occur more than three times in the training set. The dictionary size is $|\mathcal{V}| = 12\,486$, from which our models sequentially select words as output. We train the word2vec skip-gram embedding [16] to obtain the word embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times |\mathcal{V}|}$ where d is the word embedding dimension and V is the dictionary size. We set $d = 300$ in our implementation.

Video Representation. We first equidistantly sample one per ten frames from a video, to reduce the frame redundancy while minimizing loss of information. We denote the number of video frames by N. We limit the maximum number of frames to be $N_{max} = 40$; if a video is too long, we use a wider interval for uniform sampling.

We employ a convolutional neural network (CNN) to encode video input. Specifically, we extract convolutional feature map of each frame from the res5c layer (*i.e.* $\mathbb{R}^{7\times7\times2,048}$) of ResNet [12] pretrained on ImageNet dataset [22], and then apply 2×2 max-pooling followed by 3×3 convolution to reduce dimension to $\mathbb{R}^{4\times4\times500}$. Reducing the number of spatial grid regions to 4×4 helps the concept word detector get trained much faster, while not hurting detection performance significantly. We denote resulting visual features of frames by $\{\mathbf{v}_n\}_{n=1}^{N}$. Throughout this paper, we use n for denoting video frame index.

2.2. An Attention Model for Concept Detection

Concept Words and Traces. We propose the *concept word detector* using LSTM networks with soft attention mechanism. Its structure is shown in the red box of Fig.2. Its goal is, for a given video, to discover a list of *concept words* that consistently appear across frame regions. The detected concept words are used as additional references for video captioning models (section 3.1), which generates output sentence by selectively attending on those words.

We first define a set of candidate words with of size V from all training captions. Among them, we discover K concept words for each video. We set $V = 2,000$ and $K = 10$ in our implementation. We first apply the automatic POS tagging of NLTK [3], to extract nouns, verbs and adjectives from all training caption sentences [7]. We then compute the frequencies of those words in a training set, and select the V most common words as word candidates.

Since we do not have groundtruth bounding boxes for concept words in videos, we cannot train individual concept detectors in a standard supervised setting. Our idea is to adopt a soft attention mechanism to infer words by tracking regions that are spatially consistent. To this end, we employ a set of *tracing LSTMs*, each of which takes care of a single spatially-consistent meaning being tracked over time, what we call *trace*. That is, we keep track of spatial attention over video frames using LSTM, so that adjacent spatial attentions resemble the spatial consistency of a single concept (*e.g.* a moving object, or an action in video clips; see Fig.1). We use a total of L tracing LSTMs to capture out L traces (or concepts), where L is the number of spatial regions in the visual feature (*i.e.* $L = 4\times4 = 16$ for $\mathbf{v} \in \mathbb{R}^{4\times4\times D}$). Fusing these L concepts together, we finally discover K concept words, as will be described next.

Computation of Spatial Attention. For each trace l, we maintain spatial attention weights $\boldsymbol{\alpha}_n^{(l)} \in \mathbb{R}^{4\times4}$, indicating where to attend on (4×4) spatial grid locations of $\mathbf{v}_n$, through video frames $n = 1 \ldots N$. The initial attention weight $\boldsymbol{\alpha}_0^{(l)}$ at $n = 0$ is initialized with an one-hot matrix, for each of L grid locations. We compute the hidden states $\mathbf{h}_n^{(l)} \in \mathbb{R}^{500}$ of the LSTM through $n = 1 \ldots N$ by:

$$\mathbf{c}_n^{(l)} = \boldsymbol{\alpha}_n^{(l)} \otimes \mathbf{v}_n \tag{1}$$

$$\mathbf{h}_n^{(l)} = \text{LSTM}(\mathbf{c}_n^{(l)}, \mathbf{h}_{n-1}^{(l)}). \tag{2}$$

where $A \otimes B = \sum_{j,k} A_{(j,k)} \cdot B_{(j,k,:)}$. The input to LSTMs is the context vector $\mathbf{c}_n^{(l)} \in \mathbb{R}^{500}$, which is obtained by applying spatial attention $\boldsymbol{\alpha}_n^{(l)}$ to the visual feature $\mathbf{v}_n$. Note that the parameters of L LSTMs are shared.

The attention weight vector $\boldsymbol{\alpha}_n^{(l)} \in \mathbb{R}^{4\times4}$ at time step n is updated as follows:

$$\mathbf{e}_n^{(l)}(j,k) = \mathbf{v}_n(j,k) \odot \mathbf{h}_{n-1}^{(l)}, \tag{3}$$

$$\boldsymbol{\alpha}_n^{(l)} = \text{softmax}\left(\text{Conv}(\mathbf{e}_n^{(l)})\right), \tag{4}$$

where $\odot$ is elementwise product, and $\text{Conv}(\cdot)$ denotes two convolution operations before softmax layer in Fig.2. Note that $\boldsymbol{\alpha}_n^{(l)}$ in Eq.(3) is computed from the previous hidden state $\mathbf{h}_{n-1}^{(l)}$ of the LSTM.

The spatial attention $\boldsymbol{\alpha}_n^{(l)}$ measures how each spatial grid location of visual feature is related to the concept being tracked through tracing LSTMs. By repeating these two steps of Eq.(1)–(3) from $n = 1$ to N, our model can continuously find important and temporally consistent meanings

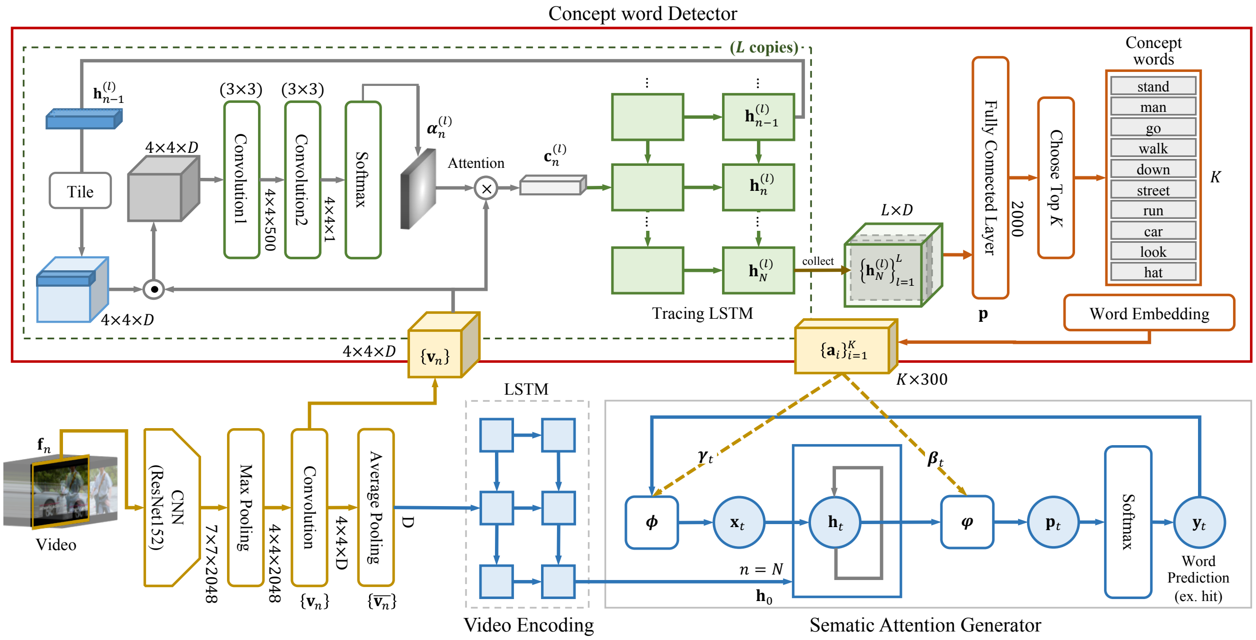


Figure 2. The architecture of attention model for the concept word detection (Top, section 2.2), and a model for video description using semantic attention based on the detected concept words (Bottom, section 3.1).

over time, that are closely related to a part of video, rather than focusing on each video frame individually.

Finally, we predict the concept confidence vector $\mathbf{p}$:

$$\mathbf{p} = \sigma \left(\mathbf{W}_p \left[\mathbf{h}_N^{(1)}; \cdots ; \mathbf{h}_N^{(L)} \right] + \mathbf{b}_p \right) \in \mathbb{R}^V, \qquad (5)$$

that is, we first concatenate the hidden states $\{\mathbf{h}_N^{(l)}\}_{l=1}^L$ at the last time step of all tracing LSTMs, apply a linear transform parameterized by $\mathbf{W}_p \in \mathbb{R}^{V\times(500L)}$ and $\mathbf{b}_p \in \mathbb{R}^V$, and apply the elementwise sigmoid activation σ.

Training and Inference. For training, we obtain a reference concept confidence vector $\mathbf{p}^* \in \mathbb{R}^V$ whose element p_i^* is 1 if the corresponding word exists in the groundtruth caption; otherwise, 0. Then we minimize the following sigmoid cross-entropy cost $\mathcal{L}_{con}$, which is often used for multi-label classification [32] where each class is independent and not mutually exclusive:

$$\mathcal{L}_{con} = -\frac{1}{V} \sum_{i=1}^{V} \left[p_i^* \log(p_i) + (1-p_i^*) \log(1-p_i) \right]. \quad (6)$$

Strictly speaking, since we apply end-to-end learning approach, the cost of Eq.(6) is used as an auxiliary term for the overall cost function, which will be discussed later.

For inference, we compute $\mathbf{p}$ for a given query video, and find top K words by score $\mathbf{p}$ (*i.e.* $\text{argmax}_{1:K}\,\mathbf{p}$). Finally, we denote these K concept words by their word embedding representations $\{\mathbf{a}_i\}_{i=1}^K$.

3. Video-to-Language Models

We design a different base model for each of LSMDC tasks, while they share the concept word detector and the semantic attention mechanism. That is, we aim to validate that the proposed concept word detection is useful to a wide range of video-to-language models. For base models, we take advantage of state-of-the-art techniques, for which we do not argue as our contribution. We refer to our video-to-language models leveraging the concept word detector as *CT-SAN* (*Concept-Tracing Semantic Attention Network*).

For better understanding of our models, we briefly summarize the four LSMDC tasks as follows: (i) *Movie description*: generating a single descriptive sentence for a given movie clip, (ii) *Fill-in-the-blank*: given a video and a sentence with a single blank, finding a suitable word for the blank from the whole vocabulary set, (iii) *Multi-choice test*: given a video query and five descriptive sentences, choosing the correct one out of them, and (iv) *Movie retrieval*: ranking 1,000 movie clips for a given natural language query.

We defer more model details to the supplementary. Especially, we skip the description of multi-choice and movie retrieval models in Figure 3(b)–(c), which can be found in the supplementary.

3.1. A Model for Description

Figure 2 illustrates the proposed video captioning model. It takes video representation $\{\mathbf{v}_n\}_{n=1}^N$ and predicted concept words $\{\mathbf{a}_i\}_{i=1}^K$ as input, and generates a sequence of words as an output sentence $\{\mathbf{y}_t\}_{t=1}^T$. The model comprises video encoding and caption decoding LSTMs, and two semantic attention models. The two LSTM networks have two layers in depth, with layer normalization [1] and dropout [24] with a rate of 0.2.

Video Encoder. The *video encoding LSTM* encodes a video into a sequence of hidden states $\{\mathbf{s}_n\}_{n=1}^{N} \in \mathbb{R}^D$.

$$\mathbf{s}_n = \text{LSTM}(\overline{\mathbf{v}_n}, \mathbf{s}_{n-1}) \tag{7}$$

where $\overline{\mathbf{v}_n} \in \mathbb{R}^D$ is obtained by $(4,4)$-average-pooling $\mathbf{v}_n$.

Caption Decoder. The *caption decoding LSTM* is a normal LSTM network as follows:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}), \tag{8}$$

where the input $\mathbf{x}_t$ is an intermediate representation of t-th word input with semantic attention applied, as will be described below. We initialize the hidden state at $t = 0$ by the last hidden state of video encoder: $\mathbf{h}_0 = \mathbf{s}_N \in \mathbb{R}^D$.

Semantic Attention. Based on [36], our model in Fig.2 uses the semantic attention in two different parts, which are called as *input* and *output* semantic attention, respectively.

The *input semantic attention* ϕ computes attention weights $\gamma_{t,i}$, which is assigned to each predicted concept word $\mathbf{a}_i$. It helps the caption decoding LSTM focus on predicted concept words differently at each step t.

The attention weight $\gamma_{t,i} \in \mathbb{R}^K$ and input vector $\mathbf{x}_t \in \mathbb{R}^D$ to the LSTM are obtained by

$$\gamma_{t,i} \propto \exp((\mathbf{E}\mathbf{y}_{t-1})^\top \mathbf{W}_\gamma \mathbf{a}_i), \tag{9}$$

$$\begin{aligned} \mathbf{x}_t &= \phi(\mathbf{y}_{t-1}, \{\mathbf{a}_i\}) \\ &= \mathbf{W}_x(\mathbf{E}\mathbf{y}_{t-1} + \text{diag}(\mathbf{w}_{x,a}) \sum_i \gamma_{t,i}\mathbf{a}_i). \end{aligned} \tag{10}$$

Since the previous word $\mathbf{y}_{t-1}$ is $|\mathcal{V}|$-dimensional, we multiply it by the word embedding matrix $\mathbf{E}$ to make it d-dimensional. The parameters to learn include $\mathbf{W}_\gamma \in \mathbb{R}^{d\times d}$, $\mathbf{W}_x \in \mathbb{R}^{D\times d}$ and $\mathbf{w}_{x,a} \in \mathbb{R}^d$.

The *output semantic attention* φ guides how to weight the concept words $\{\mathbf{a}_i\}$ when generating an output word $\mathbf{y}_t$ at each step. We use $\mathbf{h}_t$, the hidden state of decoding LSTM at t as an input to the output attention function φ. We then compute $\mathbf{p}_t \in \mathbb{R}^D$ by attending the concept words set $\{\mathbf{a}_i\}$ with the weight $\beta_{t,i}$:

$$\beta_{t,i} \propto \exp(\mathbf{h}_t^\top \mathbf{W}_\beta \sigma(\mathbf{a}_i)), \tag{11}$$

$$\begin{aligned} \mathbf{p}_t &= \varphi(\mathbf{h}_t, \{\mathbf{a}_i\}) \\ &= \mathbf{h}_t + \text{diag}(\mathbf{w}_{h,a}) \sum_i \beta_{t,i} \mathbf{W}_\beta \sigma(\mathbf{a}_i), \end{aligned} \tag{12}$$

where σ is the hyperbolic tangent, and parameters include $\mathbf{w}_{h,a} \in \mathbb{R}^D$ and $\mathbf{W}_\beta \in \mathbb{R}^{D\times d}$.

Finally, the probability of output word is obtained as

$$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \text{softmax}(\mathbf{W}_y \mathbf{p}_t + \mathbf{b}_y), \tag{13}$$

where $\mathbf{W}_y \in \mathbb{R}^{|\mathcal{V}|\times D}$ and $\mathbf{b}_y \in \mathbb{R}^{|\mathcal{V}|}$. This procedure loops until $\mathbf{y}_t$ corresponds to the <EOS> token.

Training. To learn the parameters of the model, we define a loss function as the total negative log-likelihood of all the words, with regularization terms on attention weights $\{\alpha_{t,i}\}$, $\{\beta_{t,i}\}$, and $\{\gamma_{t,i}\}$ [36], as well as the loss $\mathcal{L}_{con}$ for concept discovery (Eq.6):

$$\mathcal{L} = -\sum_t \log p(\mathbf{y}_t) + \lambda_1(g(\beta) + g(\gamma)) + \lambda_2 \mathcal{L}_{con} \tag{14}$$

where λ_1, λ_2 are hyperparameters and g is a regularization function with setting to $p = 2, q = 0.5$ as

$$\begin{aligned} g(\alpha) &= \|\alpha\|_{1,p} + \|\alpha^\top\|_{1,q} \\ &= \left[\sum_i \left[\sum_t \alpha_{t,i}\right]^p\right]^{1/p} + \left[\sum_t \left[\sum_i \alpha_{t,i}\right]^q\right]^{1/q}. \end{aligned} \tag{15}$$

For the rest of models, we transfer the parameters of the concept word detector trained with the description model, and allow the parameters being fine-tuned.

3.2. A Model for Fill-in-the-Blank

Figure 3(a) illustrates the proposed model for the fill-in-the-blank task. It is based on *Bidirectional LSTM network* (BLSTM) [23, 13], which is useful in predicting a blank word from an imperfect sentence, since it considers the sequence in both forward and backward direction. Our key idea is that we employ the semantic attention mechanism on both input and output of the BLSTM, to strengthen the meaning of input and output words with the concept words which are obtained from the concept word detector.

The model takes word representation $\{\mathbf{c}_t\}_{t=1}^T$ and concept words $\{\mathbf{a}_i\}_{i=1}^K$ as input. Each $\mathbf{c}_t \in \mathbb{R}^d$ is obtained by multiplying the one-hot word vector of a word by $\mathbf{E}$. Suppose that the t-th text input is a blank word for which we use a special token <blank>. Then, we add the word prediction module only on top of the t-th step of the BLSTM.

BLSTM. The input video is represented by the *video encoding LSTM* in Figure 2. The hidden state of the final video frame $\mathbf{s}_N$ is used to initialize the hidden states of the BLSTM: $\mathbf{h}_{T+1}^b = \mathbf{h}_0^f = \mathbf{s}_N$, where $\{\mathbf{h}_t^f\}_{t=1}^T$ and $\{\mathbf{h}_t^b\}_{t=1}^T$ are the forward and backward hidden states of BLSTM, respectively. The BLSTM is represented as:

$$\mathbf{h}_t^f = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}^f), \tag{16}$$

$$\mathbf{h}_t^b = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t+1}^b). \tag{17}$$

We also use the layer normalization [1].

Semantic Attention. The input and output semantic attention of this model is almost identical to those of the captioning model in section 3.1, only except that the word representation $\mathbf{c}_t \in \mathbb{R}^d$ is used as input at each time step,

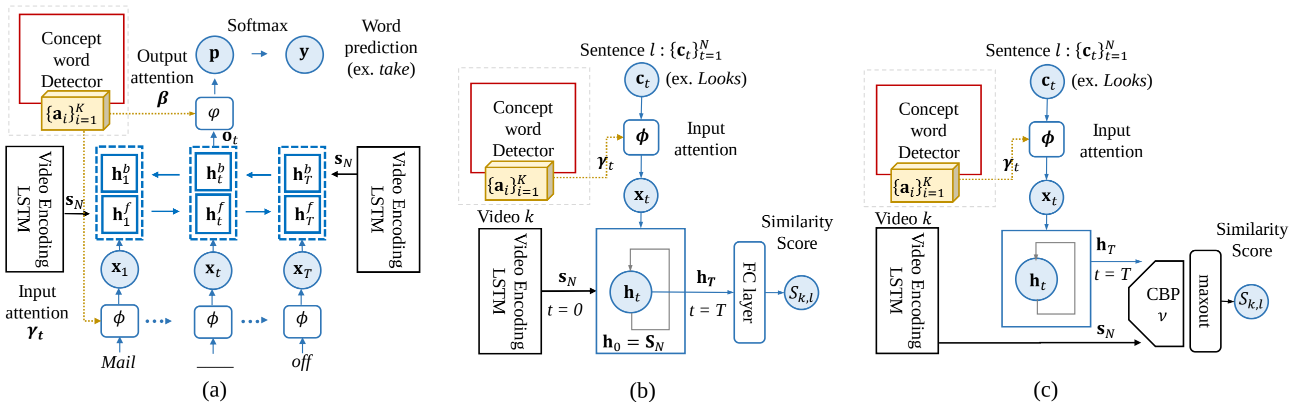


Figure 3. The model architectures for (a) fill-in-the-blank (section 3.2), (b) multi-choice, and (c) movie retrieval task. The description of models for (b)–(c) can be found in the supplementary. Each of the models take advantage of the concept word detector described illustrated in Figure 2, and semantic attention for the sake of its objective.

instead of previous word vector $\mathbf{y}_{t-1}$. Then the attention weighted word vector $\{\mathbf{x}_t\}_{t=1}^T$ is fed into the BLSTM.

The output semantic attention is also similar to that of the captioning model in section 3.1, only except that we apply the attention only once at t-th step when `<blank>` token is taken as input. We feed the output of BLSTM

$$\mathbf{o}_t = \tanh(\mathbf{W}_o[\mathbf{h}_t^f; \mathbf{h}_t^b] + \mathbf{b}_o), \tag{18}$$

where $\mathbf{W}_o \in \mathbb{R}^{D \times 2D}$ and $\mathbf{b}_o \in \mathbb{R}^D$, into the output attention function φ. It generates $\mathbf{p} \in \mathbb{R}^D$ as in Eq.(12) of the description model: $\mathbf{p} = \varphi(\mathbf{o}_t, \{\mathbf{a}_i\})$.

Finally, the output word probability $\mathbf{y}$ given $\{\mathbf{c}_t\}_{t=1}^T$ is obtained via softmax on $\mathbf{p}$ as

$$p(\mathbf{y} \mid \{\mathbf{c}_t\}_{t=1}^T) = \text{softmax}(\mathbf{W}_y \mathbf{p} + \mathbf{b}_y), \tag{19}$$

where parameters include $\mathbf{W}_y \in \mathbb{R}^{|\mathcal{V}| \times D}$ and $\mathbf{b}_y \in \mathbb{R}^{|\mathcal{V}|}$.

Training. During training, we minimize the loss $\mathcal{L}$ as

$$\mathcal{L} = -\log p(\mathbf{y}) + \lambda_1(g(\beta) + g(\gamma)) + \lambda_2 \mathcal{L}_{con}, \tag{20}$$

where λ_1, λ_2 are hyperparameters, and g is the same regularization function of Eq.(15). Again, $\mathcal{L}_{con}$ is the cost of Eq.(6) in the concept word detector.

4. Experiments

We report the experimental results of the proposed models for the four tasks of LSMDC 2016. More experimental results and implementation details can be found in the supplementary. We plan to make public our source code.

4.1. The LSMDC Dataset and Tasks

The LSMDC 2016 comprises four video-to-language tasks on the LSMDC dataset, which contains a parallel corpus of 118,114 sentences and 118,081 video clips sampled from 202 movies. We exactly follow the evaluation protocols of the challenge. We defer more details of the dataset and challenge rules to [20] and the challenge homepage[1].

Movie Description. This task is related to video captioning; given a short video clip, its goal is to generate a single descriptive sentence. The challenge provides a subset of LSMDC dataset named *LSMDC16*. It is divided into training, validation, public test, and blind test set, whose sizes are 91,941, 6,542, 10,053, and 9,578, respectively. The performance metrics include BLEU-1,2,3,4 [17], METEOR [2], ROUGE-L [15] and CIDEr [27].

Multiple-Choice Test. Given a video query and five candidate captions, from which its goal is to find the best option. The correct answer is the GT caption of the query video, and four other distractors are randomly chosen from the other captions that have different activity-phrase labels from the correct answer. The evaluation metric is the percentage of correctly answered test questions from 10,053 public-test data.

Movie Retrieval. The objective is, given a short query sentence, to find its corresponding video out of 1,000 candidate videos, sampled from the LSMDC16 public-test data. The evaluation metrics include Recall@1, Recall@5, Recall@10, and Median Rank (MedR). The Recall@k means the percentage of GT videos in the first k retrieved videos, and the MedR indicates the median rank of GT videos. Each algorithm predicts $1,000 \times 1,000$ pairwise rank scores between phrases and videos, from which all the evaluation metrics are calculated.

Movie Fill-in-the-Blank. This task is related to visual question answering. The task is, given a video clip and a sentence with a blank in it, to predict a single correct word to fill in the blank. The test set includes 30,000 examples from 10,000 clips (*i.e.* about 3 examples per sentence). The

[1] https://sites.google.com/site/describingmovies/

Movie Description	B1	B2	B3	B4	M	R	Cr
EITanque	0.144 (4)	0.042 (5)	0.016 (3)	0.007 (2)	0.056 (7)	0.130 (7)	0.098 (2)
S2VT [29]	**0.162** (1)	**0.051** (1)	**0.017** (1)	0.007 (2)	0.070 (4)	0.149 (4)	0.082 (4)
SNUVL	0.157 (2)	0.049 (2)	0.014 (4)	0.004 (6)	0.071 (2)	0.147 (5)	0.070 (6)
sophieag	0.151 (3)	0.047 (3)	0.013 (5)	0.005 (4)	**0.075** (1)	0.152 (2)	0.072 (5)
ayush11011995	0.116 (8)	0.032 (7)	0.011 (7)	0.004 (6)	0.070 (4)	0.138 (6)	0.042 (8)
rakshithShetty	0.119 (7)	0.024 (8)	0.007 (8)	0.003 (8)	0.046 (8)	0.108 (8)	0.044 (7)
Aalto	0.070 (9)	0.017 (9)	0.005 (9)	0.002 (9)	0.033 (9)	0.069 (9)	0.037 (9)
Base-SAN	0.123 (6)	0.038 (6)	0.013 (5)	0.005 (4)	0.066 (6)	0.150 (3)	0.090 (3)
CT-SAN	0.135 (5)	0.044 (4)	**0.017** (1)	**0.008** (1)	0.071 (2)	**0.159** (1)	**0.100** (1)

Fill-in-the-Blank	
Methods	Accuracy
Simple-LSTM	30.9
Simple-BLSTM	31.6
Base-SAN	34.5
amirmazaheri	34.2
SNUVL (Single)	38.0
SNUVL (Ensemble)	40.7
CT-SAN (Single)	41.9
CT-SAN (Ensemble)	**42.7**

Table 1. **Left**: Performance comparison for the movie description task on the LSMDC2016 public test dataset. For language metrics, we use BLEU (B), METEOR (M), ROUGE (R), and CIDEr (Cr). We also show the ranking in parentheses. **Right**: Accuracy comparison (in percentage) for the movie fill-in-the-blank task.

Tasks	Multi-Choice	Movie Retrieval			
Methods	Accuracy	R@1	R@5	R@10	MedR
Aalto	39.7	–	–	–	–
SA-G+SA-FC7 [26]	55.1	3.0	8.8	13.2	114
LSTM+SA-FC7 [26]	56.3	3.3	10.2	15.6	88
C+LSTM+SA-FC7 [26]	58.1	4.3	12.6	18.9	98
Base-SAN	60.1	4.3	13.0	18.2	83
SNUVL (Single)	63.1	3.8	13.6	18.9	80
EITanque	63.7	4.7	15.9	23.4	64
SNUVL (Ensemble)	65.7	3.6	14.7	23.9	50
CT-SAN (Single)	63.8	4.5	14.1	20.9	67
CT-SAN (Ensemble)	**67.0**	**5.1**	**16.3**	**25.2**	**46**

Table 2. Performance comparison for the multiple-choice test (accuracy in percentage) and movie retrieval task: Recall@k (R@k, higher is better) and Median Rank (MedR, lower is better).

evaluation metric is the prediction accuracy, which is the percentage of predicted words that match with GTs.

We compare with the results on the public dataset in the official evaluation server of LSMDC 2016 as of the submission deadline (*i.e*. November 15th, 2016 UTC 23:59).

4.2. Quantitative Results

Movie description. Table 1 compares the performance of movie description between different algorithms. Among comparable models, our approach ranks (5, 4, 1, 1)-th in the BLEU language metrics, and (2, 1, 1)-th in the other language metrics. That is, our approach ranks first in four metrics, which means that our approach is comparable to the state-of-the-art methods. In order to quantify the improvement by the proposed concept word detection and semantic attention, we implement a variant (Base-SAN), which is our model of Fig.2 without those two components. As shown in Table 1, the performance gaps between (CT-SAN) and (Base-SAN) are significant.

Movie Fill-in-the-Blank. Table 1 also shows the results of the fill-in-the-blank task. We test an ensemble of our models, denoted by CT-SAN (Ensemble); the answer word is obtained by averaging the output word probabilities of three identical models trained independently. Our approach outperforms all the participants with large margins. We also compare our model with a couple of baselines: (CT-SAN) outperforms the simple single-layer LSTM/BLSTM variants with the scoring layer on top of the blank location, and (Base-SAN), which is the base model of (CT-SAN) without the concept detector and semantic attention.

Movie Multiple-Choice Test. For the multiple-choice test, our approach also ranks first as shown in Table 2. As in the fill-in-the-blank, the multiple-choice task also benefits from the concept detector and semantic attention. Moreover, an ensemble of six models trained independently further improves the accuracy from 63.8% to 67.0%.

Movie Retrieval. Table 2 compares Recall@k (R@k) and Median Rank (MedR) metrics between different algorithms on 1,000 video/sentence test pairs. We also achieve the best movie retrieval performance with significant margins from baselines. Our CT-SAN (Ensemble) obtains the video-sentence similarity matrix with an ensemble of two different types of models. First, we train six retrieval models with different parameter initialization. Second, we obtain the similarity matrix using the multi-choice version of (CT-SAN) because it can also generate a similarity score for a video-sentence pair. Finally, we average the seven similarity matrices into the final similarity matrix.

4.3. Qualitative Results

Figure 4 illustrates qualitative results of our algorithm with correct or wrong example for each task. In each set, we show sampled frames of a query video, groundtruth (GT), our prediction (Ours), and the detected concept words. We provide more examples in the supplementary.

Movie Description. Figure 4(a)-(b) shows some examples of our movie description. As shown in the examples, our predicted sentences are often related to the content of clips well, but the words themselves are not always identical to the GTs. For instance, the generated sentence for Figure 4(b) reads *the clock shows a minute*, which is relevant to the video clip although its GT sentence much focuses on *awards on a shelf*. Nonetheless, concept words relevant to

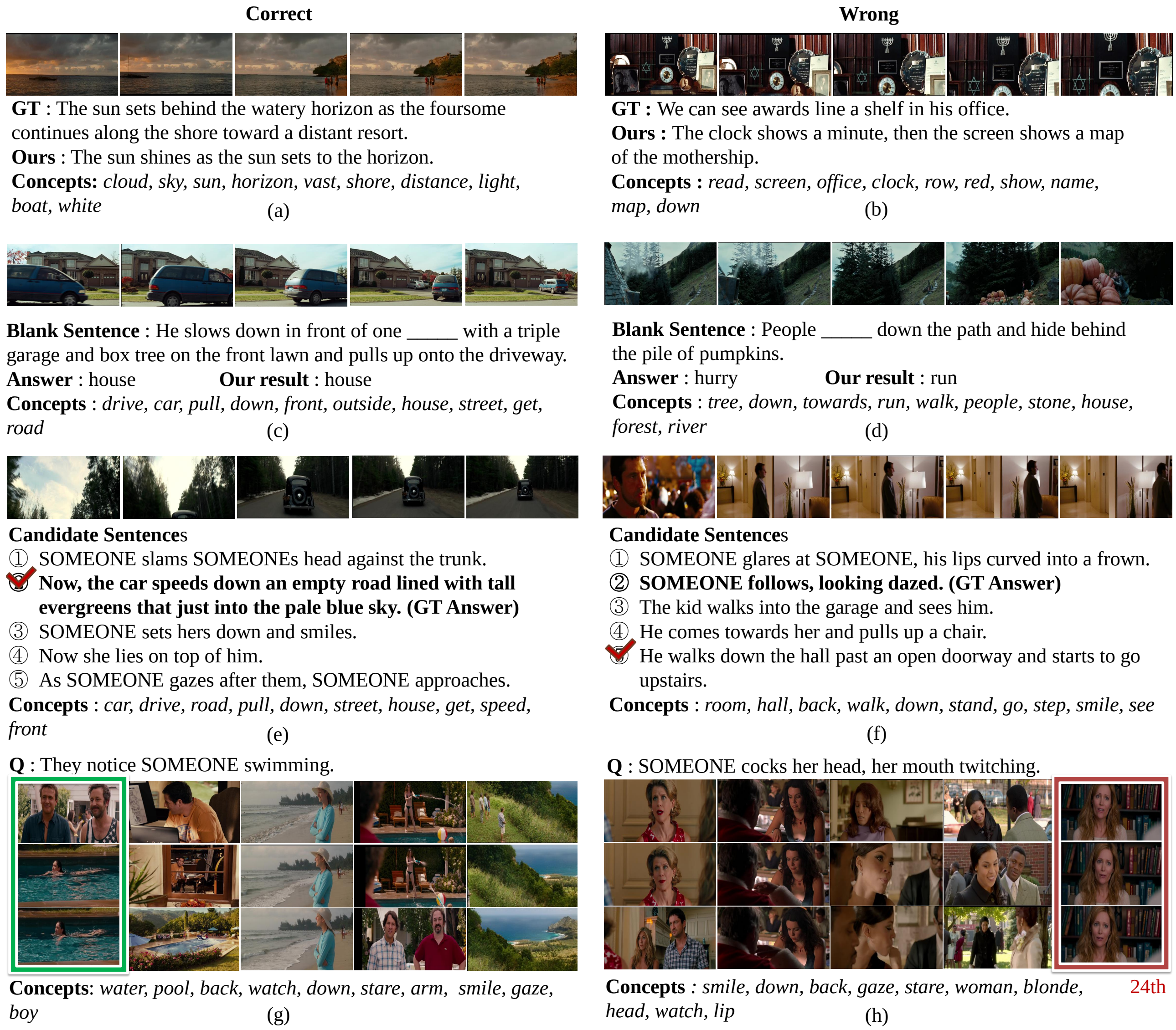


Figure 4. Examples of the four visual to language task: (a)-(b) movie description, (c)-(d) fill-in-the-blank, (e)-(f) multi-choice, and (g)-(h) movie retrieval. The left column shows correct examples and the right column shows wrong examples. In (h), we also show our retrieval ranks of the GT clips (the red box), 24th. We present more, clearer, and larger examples in the supplementary.

the GT sentence such as *office* or *clock* are well detected.

Movie Fill-in-the-Blank. Figure 4(c) shows that the detected concept words are indeed well matched with the content of the clip, and possibly help predict the correct answer. Figure 4(d) is a near-miss case where our model also predict a plausible answer (*e.g*. *run* instead of *hurry*).

Movie Multiple-Choice Test. Figure 4(e) shows that our concept detection successfully guides the model to select the correct answer. Figure 4(f) is an example of failure to understand the situation; the fifth candidate is chosen because it is overlapped with much of detected words such as *hall*, *walk*, *go*, although the correct answer is the second.

Movie Retrieval. Interestingly, the concept words of Figure 4(g) capture the abstract relation between *swimming*, *water*, and *pool*. Thus, the first to fifth retrieved clips include *water*. Figure 4(h) is a near-miss example in which our method fails to catch rare word like *twitch* and *cocks*. The first to fourth retrieved clips contain a woman's head and mouth, yet miss to catch subtle movement of mouth.

5. Conclusion

We proposed an end-to-end trainable approach for detecting a list of concept words that can be used as semantic priors for multiple-video-to-language models. We also developed a semantic attention mechanism that maximizes the values of discovered concept words. We implemented our approach into multiple video-to-language models to participate in four tasks of LSMDC 2016. We demonstrated that our approach indeed improved the performance of captioning, retrieval, and question answering; specifically, our approach achieved the best accuracies in three tasks in LSMDC 2016, including *fill-in-the-blank*, *multiple-choice test*, and *movie retrieval*.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. *arXiv:1607.06450*, 2016. 5, 12

[2] S. Banerjee and A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL*, 2005. 6

[3] S. Bird, E. Loper, and E. Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. 3

[4] D. L. Chen and W. B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *ACL*, 2011. 1

[5] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. In *CVPR*, 2013. 2

[6] J. Donahue, L. A. 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. 1, 2

[7] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig. From Captions to Visual Concepts and Back. In *CVPR*, 2015. 2, 3

[8] A. Fukui, D. Park, A. Rohrbach, T. Darrel, and M. Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *EMNLP*, 2016. 11

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 12

[10] I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML*, 2013. 11

[11] 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. 1, 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3

[13] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. In *IEEE*, 1997. 5

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 12

[15] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *WAS*, 2004. 6

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 2013. 3

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002. 6

[18] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent Multi-Sentence Video Description with Variable Level of Detail. In *GCPR*, 2014. 1

[19] A. Rohrbach, M. Rohrbach, and B. Schiele. The Long-Short Story of Movie Description. In *GCPR*, 2015. 1, 2

[20] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie Description. *arXiv:1605.03705*, 2016. 1, 2, 6

[21] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating Video Content to Natural Language Descriptions. In *ICCV*, 2013. 2

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 3

[23] M. Schuster and K. K. Paliwal. Bidirectional Recurrent Neural Networks. In *IEEE TSP*, 1997. 5

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*, 2014. 5, 12

[25] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *CVPR*, 2016. 1

[26] A. Torabi, N. Tandon, and L. Sigal. Learning Language-Visual Embedding for Movie Understanding with Natural-Language. *arXiv:1609.08124*, 2016. 7

[27] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based Image Description Evaluation. *arXiv:1411.5726*, 2014. 6

[28] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. Captioning Images with Diverse Objects. *arXiv:1606.07770*, 2016. 2

[29] S. Venugopalan, R. Marcus, D. Jeffrey, M. Raymond, D. Trevor, and S. Kate. Sequence to Sequence - Video to Text. In *ICCV*, 2015. 1, 2, 7, 15

[30] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *HLT-NAACL*, 2015. 2

[31] Q. Wu, C. Shen, A. v. d. Hengel, P. Wang, and A. Dick. Image Captioning and Visual Question Answering Based on Attributes and Their Related External Knowledge. *arXiv:1603.02814*, 2016. 2

[32] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, 2016. 4

[33] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015. 2

[34] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. In *AAAI*, 2015. 1

[35] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing Videos by Exploiting Temporal Structure. In *ICCV*, 2015. 2, 15

[36] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 2, 5

[37] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks. In *CVPR*, 2016. 1, 2

Appendix

A. Details of Video-to-Language Models

In this section, we describe the further details of video-to-language models (Section 3).

A.1. A Model for Multiple-Choice Test

Figure 5(b) illustrates the proposed model for the multiple-choice test. It takes a video and five choice sentences among which only one is the correct answer. Hence, our model computes the compatibility scores between the query video and five sentences, and selects the one with the highest score.

The multiple-choice model shares much resemblance to the model for fill-in-the-blank in Figure 5(a). First, it is based on the LSTM network, although it is not bi-directional. Second, it inputs the query video into the video encoding LSTM, and use its last hidden state $\mathbf{s}_N$ to initialize the following LSTM. Third, it uses the same word representation $\{\mathbf{c}_t\}_{t=1}^T$ for each candidate sentence. Finally, it exploits the same *input semantic attention* of Eq.(9)–(10), although it does not apply the *output semantic attention* because output is not a word but a score in this task.

We obtain a joint embedding of a pair of a single video and a sentence using the LSTM network:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{21}$$

where $\mathbf{x}_t = \phi(\mathbf{c}_t, \{\mathbf{a}_i\}) \in \mathbb{R}^D$ is obtained via the input semantic attention ϕ of Eq.(9)–(10), from the input sentence representation $\{\mathbf{c}_t\}_{t=1}^T$. We also initialize the hidden state $\mathbf{h}_0 = \mathbf{s}_N$ by the final hidden state of video representation. Once the sentence is fed into the LSTM, we obtain a multi-modal embedding of a video-sentence pair as the final hidden state $\mathbf{h}_T$ of the LSTM.

Alignment Objective. The objective of the multiple-choice model is to assign high scores for the correctly matched video-sentence pairs but low scores for incorrect pairs. Therefore, we predict a similarity score S_{kl} between a movie clip k and a sentence l as follows:

$$S_{kl} = (\mathbf{W}_s)^\top \text{ReLU}(\mathbf{W}_a \mathbf{h}_T + \mathbf{b}_a), \tag{22}$$

where $\mathbf{W}_a \in \mathbb{R}^{D\times D}$, $\mathbf{b}_a \in \mathbb{R}^D$ and $\mathbf{W}_s \in \mathbb{R}^D$ are parameters. We train the model using a max-margin structured loss objective:

$$\mathcal{L} = \sum_k \sum_{l=1}^{5} \max(0, S_{k,l} - S_{k,l^*} + \Delta) + \lambda_1 \cdot g(\gamma) + \lambda_2 \mathcal{L}_{con} \tag{23}$$

where l^* denotes the answer sentence among the five candidates. This objective encourages a positive video-sentence pair to have a higher score than a misaligned negative pair by a margin Δ. We use $\Delta = 1$ in our experiments.

At test, for a query video k, we compute five scores $\{S_{k,l}\}_{l=1}^5$ of the candidate sentences, and select the one with maximum score $S_{k,l}$ as the answer.

A.2. A Model for Retrieval

Figure 5(c) illustrates our model for movie retrieval. The basic idea is to compute a score for a query text and video pair, by learning a joint representation between two modalities (*i.e.* query text and video) using the CBP (Compact Bilinear Pooling) layer [8].

For the video encoding, we use the final hidden state $\mathbf{s}_N$ of the video encoding LSTM as done in other models. We also obtain a query representation via input semantic attention like as in section A.1, through the LSTM network:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{24}$$

Similarly, $\mathbf{x}_t = \phi(\mathbf{c}_t, \{\mathbf{a}_i\}) \in \mathbb{R}^D$ is obtained via the input semantic attention of Eq.(9)–(10), from the input query sentence representation $\{\mathbf{c}_t\}_{t=1}^T$. Then, we use the final hidden state $\mathbf{h}_T$ of query encoding LSTM as query representation.

To measure a similarity score $S_{k,l}$ between a movie k and a sentence l as follows (see Figure 5(c)):

$$S_{k,l} = (\mathbf{W}_s)^\top \text{maxout}(\mathbf{W}_p^\top \nu(\mathbf{s}_N, \mathbf{h}_T)) \tag{25}$$

where $\nu(\cdot)$ denotes the CBP (Compact Bilinear Pooling) layer [8], which captures the interactions between different modalities better than simple concatenation. That is, we learn the multimodal space for common features between video encoding LSTM and query encoding LSTM. The joint representation extracted from the MCB layer is multiplied by $\mathbf{W}_p \in \mathbb{R}^{8{,}000\times 1{,}500}$, and further processed by a consequent maxout layer [10], which yields non-sparse activations while mitigating overfitting. Finally, we obtain the score $S_{k,l}$ by multiplying the output by $\mathbf{W}_s \in \mathbb{R}^{1500\times 1}$.

We use the same max-margin structured loss objective with the multiple-choice model:

$$\mathcal{L} = \sum_k \sum_l \max(0, S_{k,l} - S_{k,l^*} + \Delta) + \lambda_1 \cdot g(\gamma) + \lambda_2 \mathcal{L}_{con} \tag{26}$$

which encourages a positive video-sentence pair to have a higher score than a misaligned pair by a margin Δ (*e.g.* $\Delta = 3$ in our experiments).

At test, for a query sentence k, we compute scores $\{S_{k,l}\}_l$ for all videos l in the test set. From the score matrix, we can rank the videos for the query. As mentioned in section 4.2, an ensemble of multiple score matrices is used in our final model, which yields much better retreival performance.

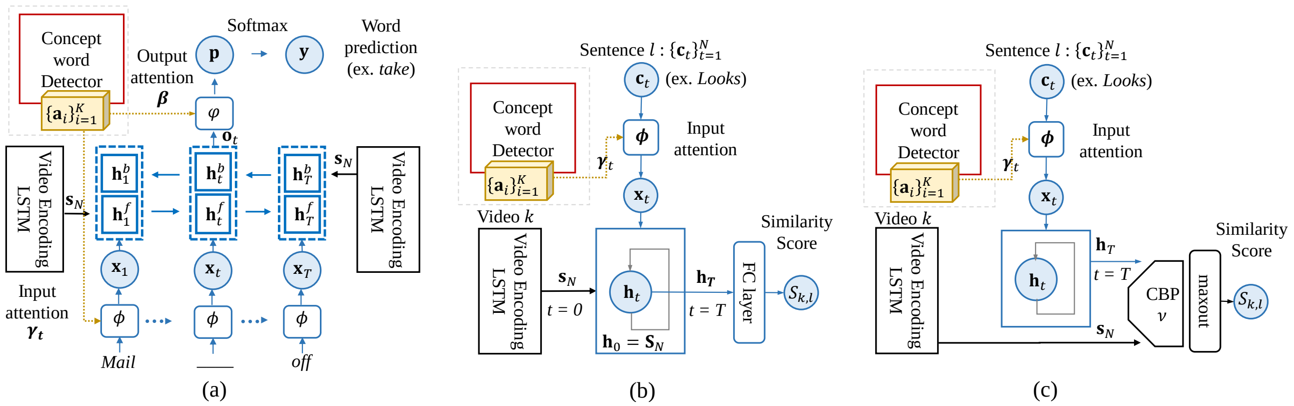


Figure 5. (Repeat of Figure 3) The model architectures for (a) fill-in-the-blank (section 3.2), (b) multi-choice (section A.1), and (c) movie retrieval task (section A.2). Each of the models take advantage of the concept word detector described illustrated in Figure 2, and semantic attention for the sake of its objective.

B. Experimental Details

B.1. Implementation Details

Optimization. We train all of our models using the Adam optimizer [14] to minimize the loss, with an initial learning rate in the range of 10^{-4} to 10^{-5}. We adopt the data augmentation of image mirroring. We also use batch shuffling in every training epoch. We use Xavier initialization [9] for initializing the weight variables. For all models, the LSTM (BLSTM) networks are two-layered in depth, and we apply layer normalization [1] and dropout [24] with a rate of 0.2 to reduce overfitting.

During training of fill-in-the-blank, multiple-choice, and retrieval models, we initialize the parameters in the concept word detector component with a pre-trained model of the movie description task. The new parameters (*e.g.* $\mathbf{W}_s, \mathbf{W}_a$ and the LSTM parameters for multi-choice test) are initialized randomly, and then the whole model is trained end-to-end using the provided training set.

Movie Description. The split of LSMDC16 dataset is provided by the challenge organizers: (training, validation, test, blind test set) $= (101079, 9578, 10053, 7409)$ video-sentence pairs respectively. We train our model using the training set of this split, and the Para-Phrase AD sentences additionally provided by the challenge organizers.

Fill-in-the-blank. The LSMDC16 dataset for the fill-in-the-blank is splitted into (training, validation, test set) $=$ $(296961, 98483, 30350)$. We also train our model using the officially provided training set only. To improve prediction accuracy, we use an ensemble of models; the answer word is obtained by averaging the output word probabilities of three copies of models trained with different initializations.

Multiple-choice test. The training/validation/test split of LSMDC16 dataset is same as in the movie description task. Although it is possible to include more negative sentences other than the provided four distractors (we also find that it leads to a better accuracy), we experiment the models trained using the four distractors only. we simply average the score matrix $S_{k,l}$ of individual models, to obtain the ensembled score matrix. In our experiments, an ensemble of six copies of model trained independently, denoted by CT-SAN (Ensemble), shows a considerable improvement of accuracy.

Movie Retrieval. Our video encoding LSTM and query encoding LSTM use the same parameter setting with the LSTM networks for movie description. We use the dropout [24] before the maxout layer with the rate of 0.5. The video-sentence similarity matrix $M \in \mathbb{R}^{1{,}000\times 1{,}000}$ is obtained with an ensemble of identical models and multiple-choice model. First, we train six retrieval models and one multiple-choice model with different parameter setting. Second, we obtain the similarity matrix of alignment score from all possible pair between 1,000 natural language sentences and 1,000 movie clip. To build an ensemble model, we average the multiple similarity matrices into the final similarity matrix.

C. More Experimental Results

In this section, we provide additional experimental results to support the validity of the proposed concept word detector and semantic attention models.

C.1. On the Quality of Concept Words

To study the effect of quality of concept words, we present two more baselines: (rand-SAN) and (no-ATT-SAN).

Random Concept Words. A baseline (rand-SAN) is a variant of the same structure as (CT-SAN), except that it uses *random* concept words instead of the ones detected by the concept word detector. We uniformly sample $K = 10$ words as concept words, from the V candidates.

Movie Description	B1	B2	B3	B4	M	R	Cr
rand-SAN	0.101	0.022	0.008	0.002	0.049	0.127	0.058
no-ATT-SAN	0.130	0.039	0.015	0.006	0.064	0.152	0.092
Base-SAN	0.123	0.038	0.013	0.005	0.066	0.150	0.090
CT-SAN	**0.135**	**0.044**	**0.017**	**0.008**	**0.071**	**0.159**	**0.100**

Tasks	Fill-in-the-Blank	Multi-Choice	Movie Retrieval			
Methods	Accuracy	Accuracy	R@1	R@5	R@10	MedR
rand-SAN	17.0	58.7	2.1	8.1	11.6	104
no-ATT-SAN	37.4	61.1	4.0	13.1	18.3	75
Base-SAN	34.5	60.1	4.3	13.0	18.2	83
CT-SAN (Single)	41.9	63.8	4.5	14.1	20.9	67
CT-SAN (Ensemble)	**42.7**	**67.0**	**5.1**	**16.3**	**25.2**	**46**

Table 3. Performance comparison of more baselines, for the movie description task (Top), and for the fill-in-the-blank, multiple-choice, and movie retrieval task (Bottom).

	Movie Description							Fill-in-the-Blank	Multi-Choice
	B1	B2	B3	B4	M	R	Cr	Accuracy	Accuracy
CT-SAN ($K = 5$)	0.133	0.043	0.015	0.007	0.066	0.156	0.100	41.5	63.0
CT-SAN ($K = 10$)	0.135	**0.044**	**0.017**	**0.008**	**0.071**	**0.159**	0.100	**41.9**	**63.8**
CT-SAN ($K = 20$)	**0.136**	**0.044**	0.016	**0.008**	0.068	0.156	**0.106**	**41.9**	63.3

Table 4. Performance comparison of our model (CT-SAN) in three tasks, varying the number of detected concept words K.

Without Attention. We also study an effect of spatial attention in the proposed concept word detector (section 2). With a simple baseline model denoted by (no-ATT-SAN), the spatial attention component in the concept word detector is replaced by a single two-layered LSTM. Specifically, we compute the LSTM states $\{\mathbf{h}_n\}_{n=1}^{N}$ (a single LSTM instead of L ones) by feeding the average-pooled visual features $\overline{\mathbf{v}_n} \in \mathbb{R}^D$, and then the concept confidence vector $\mathbf{p}$ using the last hidden state:

$$\mathbf{h}_n = \text{LSTM}(\overline{\mathbf{v}_n}, \mathbf{h}_{n-1}) \;\; (n = 1 \dots N), \tag{27}$$

$$\mathbf{p} = \sigma(\mathbf{W}_p \mathbf{h}_N + \mathbf{b}_p) \;\; \in \mathbb{R}^V, \tag{28}$$

which replaces Eq.(2) and Eq.(5), respectively. This baseline model simply transforms the video representation into concept words, but does not involve any spatial attention.

Quantitative Result. As shown in Table 3, the performance of (no-ATT-SAN) is better than (Base-SAN), but poorer than the full model (CT-SAN), in all of the four tasks. This implies that the spatial attention helps detect concept words that are useful for video captioning. Especially, (CT-SAN) outperforms (no-ATT-SAN) in the fill-in-the-blank and the multi-choice tasks with a large margin. Nevertheless, using semantic attention turns out to be more helpful than not using it, as one can observe that (no-ATT-SAN) shows better performance than (Base-SAN).

The performance of (rand-SAN) with semantic attention but with very poor concept words, is much inferior to (Base-SAN), which even lacks semantic attention. As such, we find that the quality of concept words is crucial for performance enhancement.

C.2. On the Number of Concept Words

We also conduct another simple experiment on the number of concept words. We compute the performance of (CT-SAN), with changing the number of detected concept words, $K \in \{5, 10, 20\}$. As shown in Table 4, we observe only a marginal performance difference. However, as the number of concept words increases, the time required to train the whole model increases, and overfitting is more prone to occur.

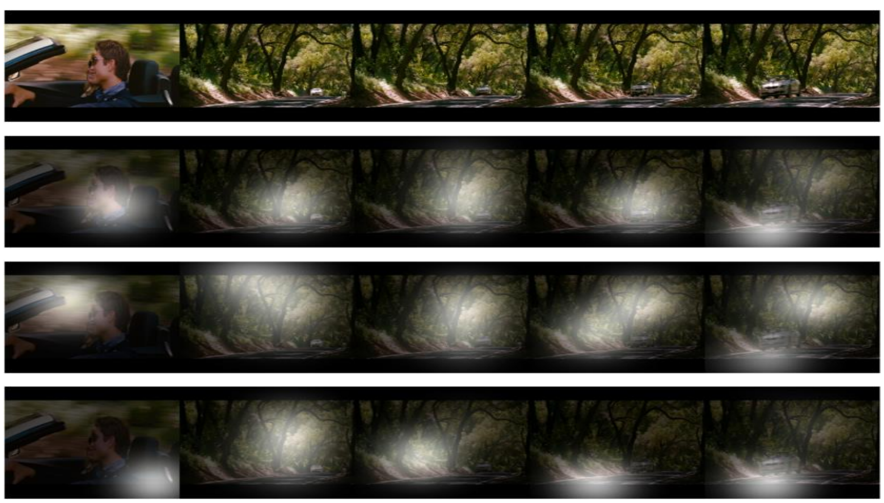
D. More Examples and Qualitative Results

We visualize some examples of the spatial attention computed in the concept word detector in Figure 6. The spatial attentions roughly captures high-level concepts in the video (*e.g.* a blue *car* moving left to right, in Fig.6(a)). Figure 7 shows some examples of generated movie description with the concept words detected by several baselines and our approach.

In the following, we present more examples of movie description results in Figure 8. Additional examples of the fill-in-the-blank task follows in Figure 9. and more examples of the multi-choice test are given in Figure 10. Finally, we present examples of the movie retrieval task in Figure 11. We also show each model's output and the detected concept words correspondingly.

GT : He slows down in front of one house with a triple garage and box tree on the front lawn and pulls up onto the driveway.
Ours : A car pulls up onto the driveway.
Concepts : *drive, down, pull, car, outside, front, house, street, get, road*

(a)



GT : Smiling and chatting, they speed down a narrow sun-dappled road in the woods.
Ours : SOMEONE drives through the trees.
Concepts : *road, drive, car, tree, house, park, down, back, speed, pull*

(b)

Figure 6. Visualization of spatial attentions in the movie description model. In the first row, we show five sampled keyframes from the input movie. Below, we select three tracing-LSTMs among $L = 16$ ones and show their spatial attention maps $\boldsymbol{\alpha}_t^{(l)}$ (see section 2).

GroundTruth : Reaching underneath her dress once again, she shimmies them up to her waist.
CT-SAN : She walks over to the couch and puts on dress.
CT-SAN-concepts : *room, couch, cat, down, sits, dress, bed, back, walk, table*
no-ATT-SAN : She puts on a shoe and puts her head down on the bed
no-ATT-SAN-concepts : *smile, sits, down, step, bed, bedroom, people, room, put, turn*
S2VT : SOMEONE is wearing a white gown and a woman in a white dress
Temporal Attention : She looks at SOMEONE

(a)

GroundTruth : They head along a winding road through mountains, across a steel girder bridge, and through rolling countryside.
CT-SAN : The sun sets down on the river and the car pass through the bridge.
CT-SAN-concepts : *drive, city, road, bridge, river, sky, down, building, sun, car*
no-ATT-SAN : The hogwarts express is visible through the mist.
no-ATT-SAN-concepts : *car, walk, field, hogwarts, track, down, wall, street, dark, past*
S2VT : The sun is rising on the dark road.
Temporal Attention : The car pulls up.

(b)

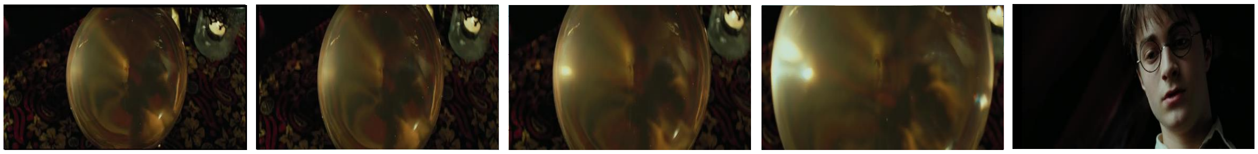
GroundTruth : Now, at night, our view glides over a highway, its lanes glittering from the lights of traffic below.
CT-SAN : The city lights are on the city
CT-SAN-concepts : *city, sky, skyscraper, light, night, crowd, view, building, sun, york*
no-ATT-SAN : The expo lights twinkle from the night
no-ATT-SAN-concepts : *crowd, couple, red, back, down, form, light, street, man, open*
S2VT : The sun is floating in the sky.
Temporal Attention : We see lights.

(c)

GroundTruth : He watches them get in their Mercedes, then spots SOMEONE in a parked car.
CT-SAN : SOMEONEs car pulls up outside the house.
CT-SAN-concepts : *car, street, get, drive, down, outside, run, front, pull, back*
no-ATT-SAN : SOMEONE and SOMEONE walk out of the car.
no-ATT-SAN-concepts : *car, street, tree, kiss, run, drive, back, down, black, house*
S2VT : SOMEONE gets out of the car and run away.
Temporal Attention : A car pulls up the street.

(d)

Figure 7. Examples of our method and baselines in movie description. We show the generated description and the detected concept words of (CT-SAN) and (no-ATT-SAN). We also compare other movie description baselines, including S2VT [29] and Temporal Attention [35] (we referenced their public code).

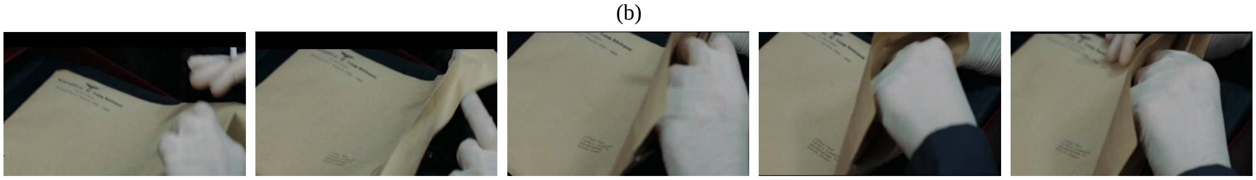


GT : He sees the face of SOMEONE.
Ours : SOMEONEs eyes widen as he stares at the glowing surface of the sphere.
Concepts : *light, glowing, sphere, screen, surface, watch, cloud, image, yellow, room*

(a)

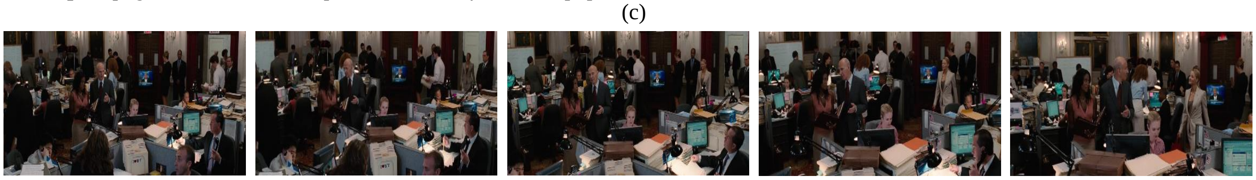
GT : A while later, SOMEONE sits alone in front of two uneaten salads on the table.
Ours : SOMEONE sets a plate on a table and sets it on the table
Concepts : *plate, table, breakfast, food, counter, kitchen, pick, egg, set, napkin*

(b)



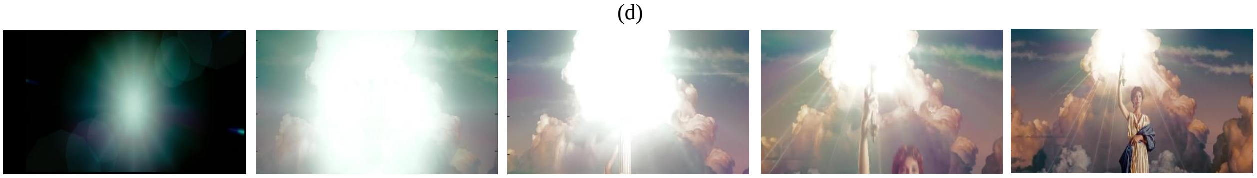
GT : We glimpse a black eagle emblem amid the return address.
Ours : SOMEONE opens the envelope and finds a note written on the page.
Concepts : *page, note, card, envelope, book, name, find, read, paper, letter*

(c)



GT : SOMEONE approaches a balding man.
Ours : SOMEONE sits at a desk in a meeting room.
Concepts : *desk, sits, table, office, down, room, man, people, phone, woman*

(d)



GT : A bright white light is held aloft by a statuesque woman wearing a long, white robe with a blue sash draped around her.
Ours : The sun shines brightly in the sky
Concepts : *light, cloud, sky, sphere, white, fiery, smoke, towards, energy, back*

(e)

GT : SOMEONE reaches out a hand and grabs SOMEONEs breast.
Ours : She is wearing a pink dress.
Concepts : *dress, woman, pink, girl, apartment, down, walk, back, black, room*

(f)

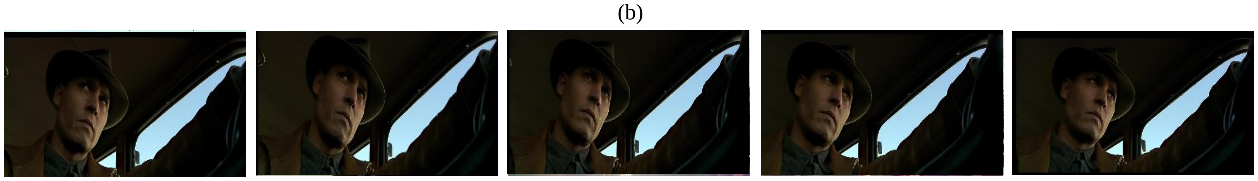
Figure 8. Examples of movie descriptions. (a)-(d) are positive examples, and (e)-(f) are near-miss or wrong examples.

Blank Sentence : He slows down in front of one _____ with a triple garage and box tree on the front lawn and pulls up onto the driveway.
Answer/Our result : (house / house)
Concepts : *drive, car, pull, down, front, outside, house, street, get, road*

(a)

Blank Sentence : They down their drinks and set the glasses back down on the _____.
Answer/Our result : (counter / counter)
Concepts : *back, down, table, drink, smile, arm, counter, sits, beer, bar*

(b)

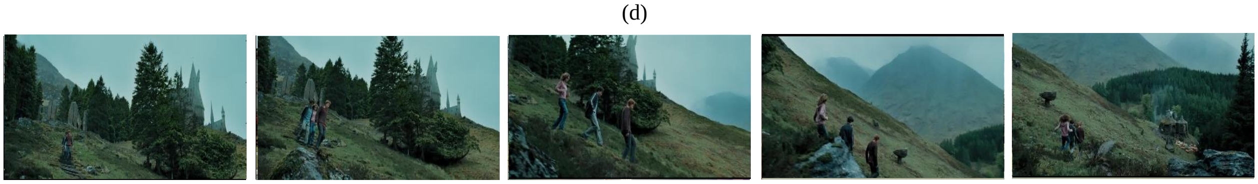


Blank Sentence : he nervously shifts his _____ back and forth under the tipped down brim of his fedora.
Answer/Our result : (eyes / eyes)
Concepts : *gaze, stare, car, glance, back, smile, window, down, watch, hat*

(c)

Blank Sentence : In the girls _____, SOMEONE sits in bed wearing glasses and writing.
Answer/Our result : (bedroom / bedroom)
Concepts : *bed, sits, back, room, down, bedroom, table, arm, open, put*

(d)



Blank Sentence : People _____ down the steep steps cut into the grassy slope towards SOMEONEs cottage.
Answer/Our result : (run / walk)
Concepts : *tree, towards, down, field, walk, people, horse, run, hill, river*

(e)

Blank Sentence : As they walk down a _____, SOMEONE takes notes.
Answer/Our result : (hall / corridor)
Concepts : *man, room, corridor, walk, step, pocket, people, down, stand, men*

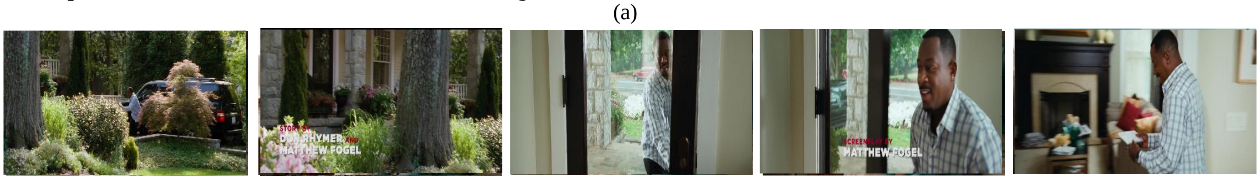
(f)

Figure 9. Examples of fill-in-the-blank task. (a)-(d) are positive examples, and (e)-(f) are near-miss or wrong examples.

① He takes the mic.
② SOMEONEs boss, SOMEONE, switches off his monitors.
③ then moves the woman down the hall.
④ He furtively puts them back in place.
⑤ The stern-faced G-Man turns gravely toward SOMEONE and nods.

Concepts : *necklace, dance, arm, dress, woman, back, hug, down, hair, back*

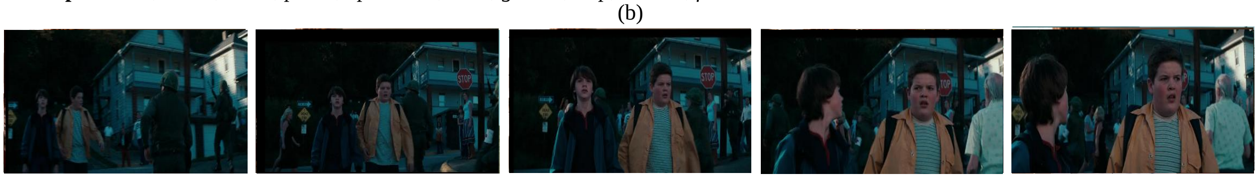
(a)



① As they drive away, SOMEONE peers through the back window seating between two agents.
② SOMEONEs at home composing.
③ SOMEONE gives SOMEONE a dubious look.
④ and climbs the stairs after the detective.
⑤ He hurries up the front walkway to his house and enters.

Concepts : *walk, house, down, porch, apartment, back, garden, step, outside, front*

(b)



① He heads over.
② His eyes catch site of a companion set by the fireplace.
③ Looking away, she shrugs again.
④ He shows her a web video of a woman dunking her breasts in cake batter.
⑤ People walk down the street.

Concepts : *walk, step, people, back, down, boy, stand, woman, run, man*

(c)

① Now her hair in pigtails, SOMEONE sings on a sound stage.
② Scientists and military men look at the wreckage of something big.
③ Now, in bathing suits, the couple jumps into a pool.
④ She sets down her belongings and sits heavily on a love seat.'
⑤ Now, SOMEONE lies on the bed using his tablet as SOMEONE enters.

Concepts : *bed, sits, bedside, room, lie, back, bedroom, gaze, stare, down*

(d)

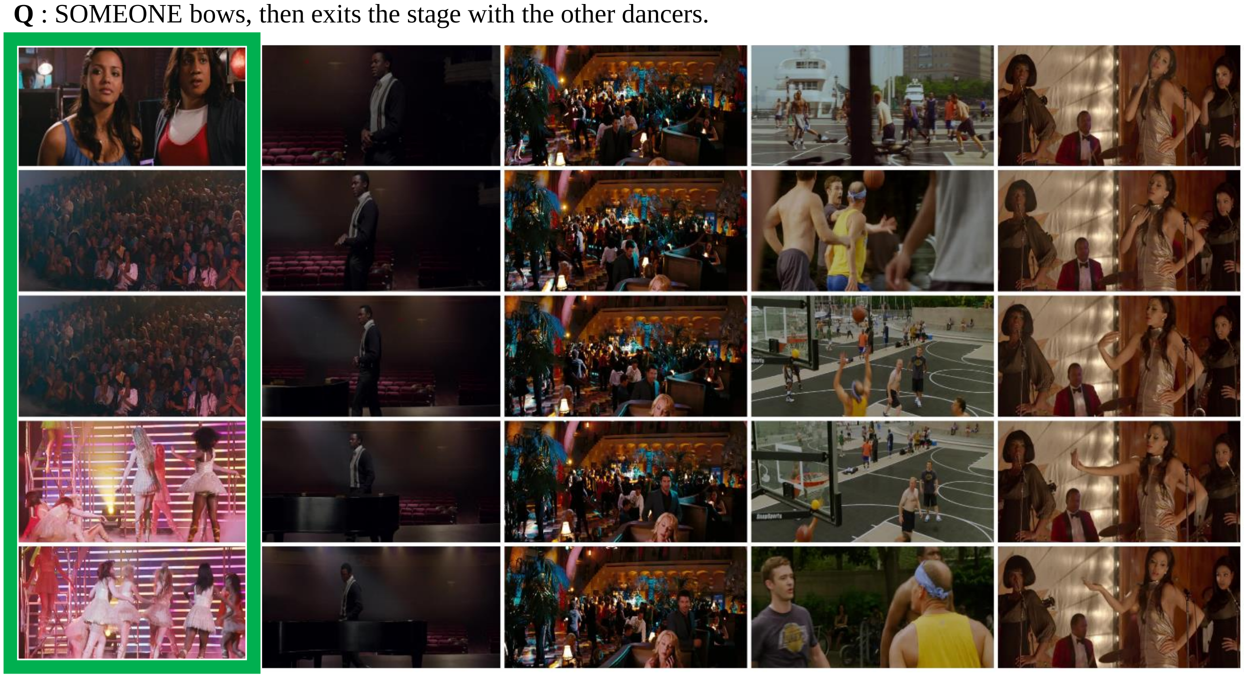
Figure 10. Examples of multiple-choice test. The groundtruth answer is in bold, and the output of our model is marked with a red checkbox. (a)-(c) are positive examples, and (d) is a near-miss or wrong example.

Q : SOMEONE meets her daughters gaze.

Concepts: *smile, down, back, gaze, stare, woman, close, room, kiss, lip*

(a)

Q : SOMEONE bows, then exits the stage with the other dancers.



Concepts: *crowd, stage, run, dancer, people, down, audience, dance, back, leap*

(b)

Figure 11. Positive examples of movie retrieval. From left to right, we show the 1st-5th retrieved movie clips from natural language sentence. The groundtruth movie clip is shown in the green box.

Q : SOMEONEs face contorts in anguish as she gazes at the comatose woman.

Concepts: *smile, gaze, nod, back, stare, give, tear, glance, woman, down* **43 th**

(a)

Q : Setting her own drink down, she faces the stage and takes his hand.

Concepts: *woman, smile, drink, bar, people, table, back, sits, watch, glass* **11 th**

(b)

Figure 12. Negative examples of movie retrieval. The first 4 columns represent the 1st-4th retrieved movie clips, and the last one is the groundtruth movie clip (in the red box). We also show the retrieved rank of the groundtruth.