Video Captioning with Multi-Faceted Attention

Xiang Long
Tsinghua University
longx13@mails.tsinghua.edu.cn

Chuang Gan
Tsinghua University
ganchuang1990@gmail.com

Gerard de Melo
Rutgers University
gdm@demelo.org

Abstract

Video captioning has attracted an increasing amount of interest, due in part to its potential for improved accessibility and information retrieval. While existing methods rely on different kinds of visual features and model architectures, they do not make full use of pertinent semantic cues. We present a unified and extensible framework to jointly leverage multiple sorts of visual features and semantic attributes. Our novel architecture builds on LSTMs with two multi-faceted attention layers. These first learn to automatically select the most salient visual features or semantic attributes, and then yield overall representations for the input and output of the sentence generation component via custom feature scaling operations. Experimental results on the challenging MSVD and MSR-VTT datasets show that our framework outperforms previous work and performs robustly even in the presence of added noise to the features and attributes.

1 Introduction

The task of automatically generating captions for videos has been receiving an increasing amount of attention. On YouTube, for example, every single minute, hundreds of hours of video content are uploaded. Obviously, there is no way a person could sit and binge-watch these overwhelming amounts of video, so new techniques to search and quickly understand them are highly sought. Generating captions, i.e., short natural language descriptions, for videos is an important technique to address this challenge, while also greatly improving their accessibility for blind and visually impaired users.

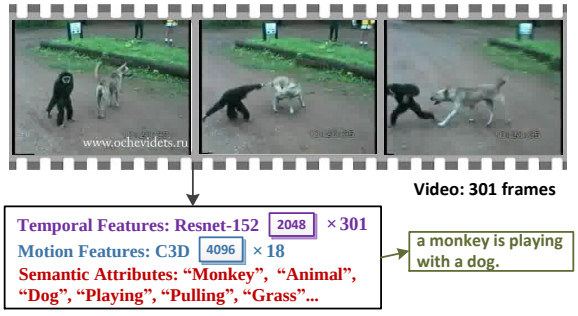

Figure 1: Example video with extracted visual features, semantic attributes, and the generated caption as output.

The study of video captioning has an extensive history but remains challenging, given the difficulties of video interpretation, natural language generation, and their interplay. Video interpretation hinges on our ability to make sense of a stream of video frames and of the relationships between consecutive frames. The generated output needs to be a grammatically correct sequence of words, and different parts of the output caption may pertain to different parts of the video. In previous work, 3D ConvNets (Du et al., 2015) have been proposed to capture motion information in short videos, while LSTMs (Hochreiter and Schmidhuber, 1997) can be used to generate natural language, and a variety of different visual attention models (Yao et al., 2015; Pan et al., 2016a; Yu et al., 2016; Long et al., 2018) have been deployed, attempting to capture the relationship between caption words and the video content.

These methods, however, only make use of visual

Transactions of the Association for Computational Linguistics, vol. 6, pp. 173–184, 2018. Action Editor: Jianfeng Gao.
Submission batch: 9/2017; Revision batch: 11/2017; Published 3/2018.

information from the video, often with unsatisfactory results. In many real-world settings, we can easily obtain additional information related to the video. Apart from sound, there may also be a title, tags, categories, and other metadata (Shutova et al., 2015), as well as user-supplied comments and cross-lingual cues. Moreover, extra labels may also be predicted automatically, as we do in the experiments in this paper. Both visual video features, as well as attributes, can be imperfect and incomplete.

However, by jointly considering all available signals, we may obtain complementary information that aids in generating better captions. Humans often benefit from additional context information when trying to understand what a video is portraying, as well.

Incorporating these additional signals is not just a matter of including additional features. While generating the sequence of words in the caption, we need to flexibly attend to 1) the relevant frames along the time axis, 2) the relevant parts within a given frame, and 3) relevant additional signals to the extent that they pertain to a particular output word. In doing so, we need to account for the heterogeneity of different features and attributes in terms of their number and scale, and we need to exploit the relationships between input words, features, and semantic attributes.

Based on these considerations, we propose a novel multi-faceted attention architecture that jointly considers multiple heterogeneous forms of input. This model flexibly attends to temporal information, motion features, and semantic attributes for every channel. The temporal and motion features are commonly used computer vision signals, while the semantic attributes, which in our paper are mainly automatically predicted labels and cross-lingual cues, provide additional information. An example of this is given in Figure 1. Each part of the attention model is an independent branch and it is straightforward incorporating additional branches for further kinds of features, making our model highly extensible. We present a series of experiments that highlight our contribution of elegantly combining features and attributes outperforming previous work on standard datasets and analyzing the stability of our model against noisy features and attributes.

2 Related Work

Machine Translation. Some of the first widely noted successes of deep sequence-to-sequence learning models were for the task of machine translation (Cho et al., 2014b; Cho et al., 2014a; Sutskever et al., 2014). In several respects, this is actually a similar task to video caption generation, just with a rather different input modality. What they share in common is that both require bridging different representations, and that often both use an encoder-decoder paradigm with a Recurrent Neural Network (RNN) decoder to generate sentences (Cao et al., 2017) in the target language. Many techniques for video captioning are inspired by neural machine translation techniques, including soft attention mechanisms focusing on different parts of the input when generating the target sentence word by word (Bahdanau et al., 2015).

Image Captioning. Image captioning can be regarded as a greatly simplified case of video captioning, with videos consisting of just a single frame. Recurrent architectures are often used here as well (Karpathy et al., 2014; Kiros et al., 2014; Chen and Zitnick, 2015; Mao et al., 2015; Vinyals et al., 2015). Spatial attention mechanisms allow for focusing on different areas of an image (Xu et al., 2015b). Recently, image captioning that incorporate semantic concepts has achieved strong results. You et al. (2016) proposed a semantic attention approach to selectively attend to semantic concept proposals and fuse them into hidden states and outputs of RNNs, but their model is difficult to extend to deal with multiple sets of features and attributes together. Overall, none of these methods for image captioning need to account for temporal and motion aspects.

Video captioning. For video captioning, many works utilize a recurrent neural architecture to generate video descriptions, conditioned on either an average-pooling (Venugopalan et al., 2015b) or recurrent encoding (Xu et al., 2015a; Donahue et al., 2015; Venugopalan et al., 2015a; Venugopalan et al., 2016) of frame-level features, or on a dynamic linear combination of annotation vectors obtained via temporal attention (Yao et al., 2015). Hierarchical recurrent neural encoders (HRNE) with attention mechanisms have been proposed to better encode videos (Pan et al., 2016a). Yu et al. (2016) exploit several

forms of visual attention and rely on a multimodal layer to combine them. In our work, we present an effective multi-faceted attention, which can achieve more stable improvements in comparison to simple multimodal layers, to jointly model multiple heterogeneous signals, and we experimentally show the benefits of this approach over previous work in Section 4.

3 The Proposed Approach

In this section, we describe our approach for combining multiple forms of attention for video captioning. Figure 3 illustrates the architecture of our model. The core of our model is a sentence generator based on Long Short Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997). Instead of a traditional sentence generator, which directly receives a previous word and selects the next word, our model relies on two multi-faceted attention layers to selectively focus on important parts of temporal, motion, and semantic features. The multi-faceted attention layers integrate information before the input reaches the sentence generator to enable better hidden representations in the LSTM, and before prediction of the next word to obtain more reasonable probability scores.

We first briefly review the basic LSTM, and then describe our model in detail, including our multi-faceted attention mechanism to consider temporal, motion, and semantic attribute perspectives.

3.1 Long Short Term Memory Networks

A Recurrent Neural Network (RNN) (Elman, 1990) is a neural network adding extra feedback connections to feed-forward networks, enabling it to work with sequences of inputs. The network is updated not only based on each input item but also based on the previous hidden state. RNNs can compute the hidden states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m)$ given an input sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ based on a recurrence of the form:

$$\mathbf{h}_t = \phi(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h), \qquad (1)$$

where the weight matrices $\mathbf{W}$, $\mathbf{U}$ and bias $\mathbf{b}_h$ are parameters to be learned, m is the length of the input sequence, and $\phi(\cdot)$ is an element-wise activation function.

RNNs trained via unfolding have proven inferior at capturing long-term temporal information. LSTM units were introduced to avoid these challenges. LSTMs not only compute the hidden states but also maintain an additional cell state to account for relevant signals that have been observed. They have the ability to remove or add information to the cell state, modulated by gates.

Given an input sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$, an LSTM unit computes the hidden state $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m)$ and cell states $(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m)$ via repeated application of the following equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \qquad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \qquad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \qquad (4)$$

$$\mathbf{g}_t = \phi(\mathbf{W}_g \mathbf{x}_t + \mathbf{U}_g \mathbf{h}_{t-1} + \mathbf{b}_g) \qquad (5)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \qquad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t, \qquad (7)$$

where $\mathbf{W}_i$, $\mathbf{W}_f$, $\mathbf{W}_o$, $\mathbf{W}_g$ are input-to-state transition matrices, $\mathbf{U}_i$, $\mathbf{U}_f$, $\mathbf{U}_o$, $\mathbf{U}_g$ are state-to-state transition matrices, $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$, $\mathbf{b}_g$ are biases to be learned, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes the element-wise multiplication of two vectors. For convenience, we denote the computations of the LSTM at each time step t as $\mathbf{h}_t$, $\mathbf{c}_t = \mathrm{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1})$.

3.2 Input Representations

When training a video captioning model, as a first step, we need to extract feature vectors and attributes that serve as inputs to the network. For visual features, we can extract one feature vector per frame, leading to a series of what we shall refer to as *temporal features*. We can also compute another form of feature vector from several consecutive frames, which we call *motion features*. Additionally, we could also extract other forms of visual features, such as features from an area of a frame, the same area of consecutive frames, etc. In this paper, we only consider temporal features, denoted by $\{\mathbf{v}_i\}$, and motion features, denoted by $\{\mathbf{u}_i\}$, which are commonly used in video captioning.

For semantic attributes, we need to extract a set of related attributes denoted by $\{a_i\}$. These can be based on title, tags, etc., if available. In our experi-

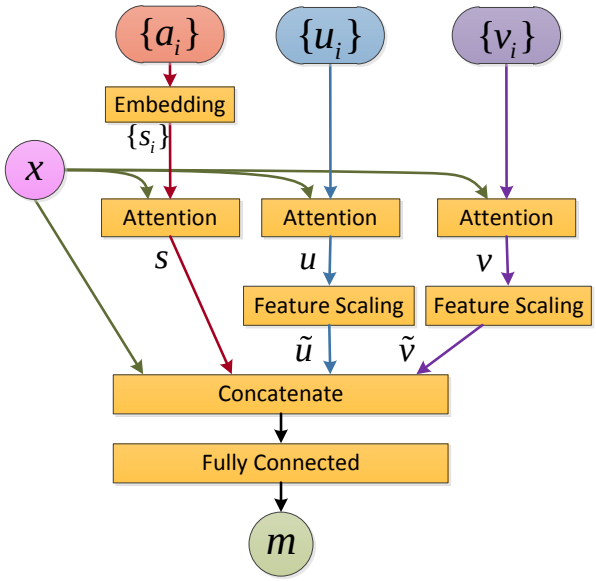

Figure 2: Multi-faceted attention for temporal features, motion features, and semantic attributes.

ments, we instead consider techniques to automatically extract or predict attributes that are not directly given. This has the advantage of being more broadly applicable. Since we have captions for the videos in the training set, we are able to train different models to predict caption-related semantic attributes for videos in the validation and test sets. We also conduct additional experiments with cross-lingual cues as external semantic attributes. We describe our specific experimental setups in Section 4.2.

The vocabulary then not only consists of the union over all words w_i in any caption of the training set, but also of the union of all attributes a_i in any video across the dataset.

An embedding matrix $\mathbf{E}$ is used to represent words and we denote $\mathbf{E}_w$ by an embedding vector of a given w. Thus, we obtain input word embedding vectors as:

$$\mathbf{x}_t = \mathbf{E}_{w_t}. \tag{8}$$

3.3 Multi-Faceted Attention

In this section, we introduce the core part of our proposed model, the multi-faceted attention mechanism. First, we introduce the components of this attention model and subsequently, we describe how it is instantiated to operate on temporal features, motion features, and semantic attributes.

3.3.1 Attribute Embedding

We have two different kinds of representations of the input video: regular features and semantic attributes. The former are feature vectors that can directly be fed to the network, while the latter are items in the vocabulary. While it is possible to transform these into vectors via one-hot encoding, the results are unsatisfactory. Considering that semantic attributes and words in captions may share a great deal of overlap, we can use the same word embedding matrix $\mathbf{E}$ to transform semantic attributes into semantic embedding vectors:

$$\mathbf{s}_i = \mathbf{E}_{a_i}. \tag{9}$$

3.3.2 Attention

Assuming that we have a series of feature vectors for a given video, we generate a caption word by word. At each step, we need to select relevant information from these feature vectors, which we from now on refer to as *annotation vectors* $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$, where n is the number of annotation vectors.

Due to the variability of the length of videos, it is challenging to directly input all these vectors to the model at every time step. A simple strategy is to compute the average of the annotation vectors and provide this average vector to the model at every time step:

$$\mathbf{y}_t = \frac{1}{m} \sum_{i=1}^{m} \mathbf{d}_i. \tag{10}$$

However, this strategy collapses all available information into a single vector, neglecting the inherent structure, which captures the temporal progression, among other things. Thus, this sort of folding leads to a significant loss of information. Instead, we wish to focus on the most salient parts of the features at every time step. Instead of naively averaging the annotation vectors $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$, a soft attention model calculates weights α_i^t for each $\mathbf{d}_i$, conditioning on the input vector $\mathbf{x}$. For this, we first compute basic attention scores e_i and then feed these through a softmax layer to obtain a set of attention weights $\{\alpha_1^t, \alpha_2^t, \ldots, \alpha_n^t\}$ that quantify the

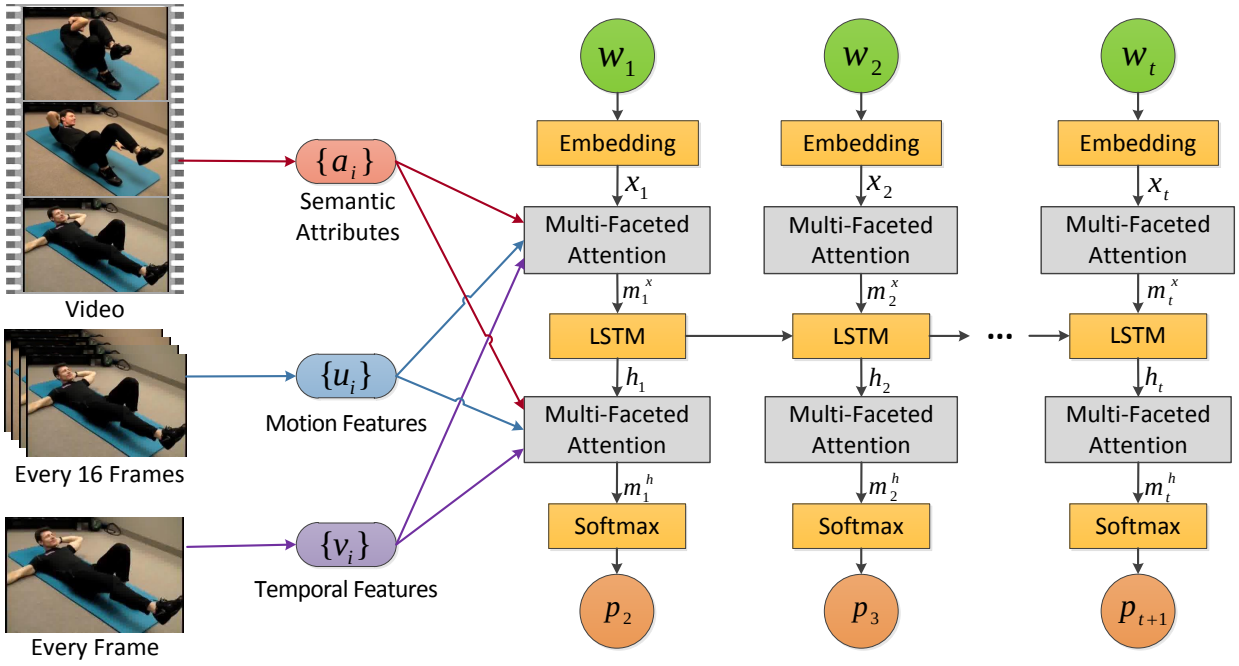

Figure 3: Overall model architecture.

relevance of $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$ for $\mathbf{x}$:

$$e_i = \mathbf{x}^\intercal \mathbf{U} \mathbf{d}_i \tag{11}$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \tag{12}$$

$$\mathbf{y} = \sum_{i=1}^{n} \alpha_i \mathbf{d}_i, \tag{13}$$

where $\mathbf{U}$ is the parameter matrix to be learned. We obtain the corresponding output vectors $\mathbf{y}$ as weighted averages. For convenience, we denote the attention model output as $\mathbf{y} = \text{Attention}(\mathbf{x}, \{\mathbf{d}_i\})$.

3.3.3 Feature Scaling

Through attention, we obtain one output vector for one feature set. Before combining them together, we need to overcome another problem. These features do not have the same scale and distribution, making it difficult to learn relationships between them. We introduce a feature scaling operation to convert features to the same scale. We also tried L2 normalization and batch normalization (Ioffe and Szegedy, 2015), but neither of these proved effective. We speculate that the former is due to the fact that the transformation for each feature is different, which breaks the value relationship along each dimension, whereas in the latter, batch normalization does not work well when the batch size is small. Our feature scaling instead is both simple and effective. For each feature, we apply an element-wise multiplication with a parameter vector of the same dimension:

$$\widetilde{\mathbf{y}} = \mathbf{w}_y \odot \mathbf{y}. \tag{14}$$

For convenience, we denote the feature scaling operation as $\widetilde{\mathbf{y}} = \text{FS}(\mathbf{y})$. We apply feature scaling only to features such as the temporal and motion features, but not to input words and semantic attributes, because the attribute embedding shares the same embedding matrix with the caption words. Keeping it unchanged can help the model discover relationships between attributes and words. In fact, we found that applying feature scaling to input words and semantic attributes leads to poorer results.

3.3.4 Multi-Faceted Attention for Temporal, Motion and Semantic

We next introduce our multi-faceted attention for temporal features, motion features, and semantic attributes, as illustrated in Figure 2. First, we transfer semantic attributes $\{a_i\}$ to semantic features $\{\mathbf{s}_i\}$

following Equation 9. For input x, we then apply the attention model to the temporal features $\{\mathbf{v}_i\}$, motion features $\{\mathbf{u}_i\}$, and semantic features $\{\mathbf{s}_i\}$:

$$\mathbf{v} = \text{Attention}(\mathbf{x}, \{\mathbf{v}_i\}) \quad (15)$$

$$\mathbf{u} = \text{Attention}(\mathbf{x}, \{\mathbf{u}_i\}) \quad (16)$$

$$\mathbf{s} = \text{Attention}(\mathbf{x}, \{\mathbf{s}_i\}). \quad (17)$$

Subsequently, we apply feature scaling to the temporal output $\widetilde{\mathbf{v}} = \text{FS}(\mathbf{v})$ and motion output $\widetilde{\mathbf{u}} = \text{FS}(\mathbf{u})$.

Finally, we obtain the output of the multi-facted attention via a fully connected layer after concatenating the input with the previous outputs.

$$\text{MFATT}(\mathbf{x}) = \phi\left(\mathbf{W}\left[\mathbf{x}, \mathbf{s}, \widetilde{\mathbf{v}}, \widetilde{\mathbf{u}}\right] + \mathbf{b}\right) \quad (18)$$

where $[\cdot]$ denotes concatenation. This model is highly extensible, since we can easily add extra branches for additional features or attributes.

3.4 Overall Architecture

The overall architecture is shown in Figure 3. The core of our model is the LSTM sentence generator. Unlike traditional generators, we do not directly feed the word embedding of the previous word $\mathbf{x}_t$ to the LSTM. Instead, we first apply our multi-faceted attention to $\mathbf{x}_t$:

$$\mathbf{m}_t^x = \text{MFATT}(\mathbf{x}_t). \quad (19)$$

Subsequently, we can obtain $\mathbf{h}_t$ via the LSTM. At the first time step, the mean values of the features are used to initialize the LSTM states to yield a general overview of the video:

$$\mathbf{m}_0^x = \mathbf{W}^i[\text{AP}(\{\mathbf{s}_i\}), \text{AP}(\{\mathbf{v}_i\}), \text{AP}(\{\mathbf{u}_i\})] \quad (20)$$

$$\mathbf{h}_0, \mathbf{c}_0 = \text{LSTM}(\mathbf{m}_0^x, 0, 0) \quad (21)$$

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{m}_t^x, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \quad (22)$$

where $\text{AP}(\cdot)$ denotes average pooling of the given feature set.

We also apply the multi-faceted attention to hidden states $\mathbf{h}_t$. This multi-faceted attention is followed by a softmax layer with a dimensionality equal to the size of the vocabulary. The projection matrix from the multi-faceted attention layer to the softmax layer is set to be the transpose of the word embedding matrix:

$$\mathbf{m}_t^h = \text{MFATT}(\mathbf{h}_t) \quad (23)$$

$$\mathbf{p}_{t+1} = \text{Softmax}(\mathbf{E}^\intercal \mathbf{m}_t^h), \quad (24)$$

where $\text{Softmax}(\cdot)$ denotes the softmax operation. We apply dropout (Srivastava et al., 2014) to each multi-faceted attention layer to reduce overfitting.

3.5 Training and Generation

We can interpret the output of the softmax layer $\mathbf{p}_{t+1}$ as a probability distribution over words:

$$\text{P}(w_{t+1} \mid w_{1:t}, V, S, \Theta), \quad (25)$$

where V denotes the corresponding video, S denotes the semantic attributes, and Θ denotes the model parameters. The overall loss function is defined as the negative logarithm of the likelihood and our goal is to learn all parameters Θ in our model by minimizing the loss function over the entire training set:

$$\min_{\Theta} -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i} \log \text{P}(w_{t+1}^i \mid w_{1:t}^i, V^i, S^i, \Theta), \quad (26)$$

where N is the total number of captions in the training set, and T_i is the number of words in caption i. During the training phase, we add a begin-of-sentence tag $\langle\text{BOS}\rangle$ to the start of the sentence and an end-of-sentence tag $\langle\text{EOS}\rangle$ to the end of sentence. We use Stochastic Gradient Descent to find the optimum Θ with the gradient computed via Backpropagation Through Time (BPTT) (Werbos, 1990). Training continues until the METEOR evaluation score on the validation set stops increasing, following previous studies that found that METEOR is more consistent with human judgments than BLEU or ROUGE (Vedantam et al., 2015). We optimize the hyperparameters, the number of hidden units in the first multi-facted attention layer as well as in the LSTM, starting from 64 and doubling the number until 2048 to maximize METEOR on the validation set.

After the parameters are learned, during the testing phase, we also have temporal and motion features extracted from the video as well as semantic attributes, which were either already given or are

predicted using a model trained on the training set. Given a previous word, we can calculate the probability distribution of the next word $\mathbf{p}_{t+1}$ using the model described above. Thus, we can generate captions starting from the special symbol $\langle \text{BOS} \rangle$ with Beam Search.

4 Experimental Results

4.1 Datasets

MSVD: We evaluate our video captioning models on the Microsoft Research Video Description Corpus (Chen and Dolan, 2011). MSVD consists of 1,970 video clips downloaded from YouTube that typically depict a single activity. Each video clip is annotated with multiple human-written descriptions in several languages. We only use the English descriptions, about 41 descriptions per video. In total, the dataset consists of 80,839 video-description pairs. Each description on average contains about 8 words. We use 1,200 videos for training, 100 videos for validation, and 670 videos for testing, as provided by previous work (Guadarrama et al., 2013).

MSR-VTT: We also evaluate on the MSR Video-to-Text (MSR-VTT) dataset (Xu et al., 2016), a recent large-scale video benchmark for video captioning. MSR-VTT provides 10,000 web video clips. Each video is annotated with about 20 sentences. Thus, we have 200,000 video-caption pairs in total. Our video captioning models are trained and hyperparameters are selected using the official training and validation set, which consists of 6,513 and 497 video clips, respectively. The models are evaluated using the test set of 2,990 video clips.

4.2 Preprocessing

Visual Features: We extract two kinds of visual features, temporal features and motion features. We use a pretrained ResNet-152 model (He et al., 2016) to extract temporal features, obtaining one fixed-length feature vector for each frame. We use a pretrained C3D (Du et al., 2015) to extract motion features. The C3D net emits a fixed-length feature vector for 16 consecutive frames. For each video, we extract 20 features for both the temporal aspect and motion with equal time interval.

Ground Truth Semantic Attributes: While MSVD and MSR-VTT are standard video caption datasets, they do not come with tags, titles, or other semantic information about the videos. Nevertheless, we can reproduce a setting with semantic attributes by extracting attributes from captions. First, we tokenize captions and remove meaningless stopwords such as "is", "at", "that", etc. We then select the most frequent 10 words across captions of each video as the ground truth semantic attributes.

We can use the ground truth semantic attributes of the training set to train a model to predict semantic attributes for the test set. Furthermore, such attributes can also be used to evaluate the robustness of our models.

Predicted Semantic Attributes: The advantage of using predicted attributes is that it is far easier to train a classification model than a captioning model. Moreover, it is simple to transfer attributes from images to videos. The problem of predicting attributes can hence be treated as a multi-label classification task. Here, we propose an attribute classifier for both images and video based on ResNet-152 features. For an image, we extract one feature vector $\mathbf{v}$ and for a video, we can extract a set of feature vectors $\mathbf{v}_i$ for its frames. We treat the ground truth semantic attributes of an image or video as target classification labels, which can be represented as a one hot vector $\hat{\mathbf{y}}$

We apply a multilayer perceptron (MLP) to predict labels for an image $\mathbf{y} = \sigma(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{v} + \mathbf{b}_1) + \mathbf{b}_2)$. For video, we first apply a simple attention layer and then invoke the same MLP layer as for images:

$$e_i = \mathbf{w}^\intercal \mathbf{v}_i \tag{27}$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \tag{28}$$

$$\mathbf{v}_{\text{att}} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \tag{29}$$

$$\mathbf{y} = \text{MLP}(\mathbf{v}_{\text{att}}), \tag{30}$$

where $\mathbf{w}$ is a parameter vector that has the same dimensionality as $\mathbf{v}_i$. Then, we can train the model by minimizing:

$$\frac{1}{N} \sum_{j=1}^{N} (\mathbf{y}_j^\intercal \log(\hat{\mathbf{y}_j}) + (1 - \mathbf{y}_j)^\intercal \log(1 - \hat{\mathbf{y}_j})) \tag{31}$$

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
LSTM-YT (Venugopalan et al., 2015b)	-	-	-	0.333	0.291	-
S2VT (Venugopalan et al., 2015a)	-	-	-	-	0.298	-
TA (Yao et al., 2015)	0.800	0.647	0.526	0.419	0.296	0.517
TA*	0.811	0.655	0.541	0.422	0.304	0.524
LSTM-E (Pan et al., 2016b)	0.788	0.660	0.554	0.453	0.310	-
HRNE-A (Pan et al., 2016a)	0.792	0.663	0.551	0.438	0.331	-
h-RNN (Yu et al., 2016)	0.815	0.704	0.604	0.499	0.326	0.658
h-RNN*	0.824	0.711	0.610	0.504	0.329	0.675
MFATT-TM (Ours)	0.826	0.717	0.619	0.508	0.332	0.694
MFATT-TM-SP (Ours)	**0.830**	**0.719**	**0.630**	**0.520**	**0.335**	**0.721**

Table 1: Comparison with existing results on MSVD, where (-) indicates unknown scores.

where the summation is over both images and videos.

We train this attribute classifier on both MSCOCO (Chen et al., 2015) and the training set of the video captioning dataset, predicting attributes on the validation and test set of the video captioning dataset. We refer to these attributes as predicted semantic attributes (SP).

External Semantic Attributes: We also consider using external information such as cross-lingual cues (de Melo and Weikum, 2009) to generate video captions. This is related to multimodal machine translation and cross-lingual image description (Specia et al., 2016; Elliott et al., 2017), which aims to generate captions in a target language based on both source language captions and images. Our model can produce captions in the target language either with or without source language cues. For the MSVD dataset, a small number of captions in other languages are available. We consider German (DE) and Chinese (CN). The latter is tokenized using the Stanford Word Segmenter (Chang et al., 2008). We consider the words in these other languages as external semantic attributes.

4.3 Evaluation Metrics

We rely on three standard metrics, BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and CIDEr (Vedantam et al., 2015) to evaluate our methods. These are commonly used in image and video captioning tasks, and allow us to compare our results against previous work. We use the Microsoft COCO evaluation toolkit (Chen et al., 2015), which is widely used in previous work, to compute the metric scores. Across all three metrics, higher scores indicate that the generated captions are assessed as being closer to captions authored by humans.

4.4 Experimental Settings

Based on our hyperparameter optimization on the validation set, the number of hidden units in the first multi-faceted attention layer and in the LSTM are both set to 512. The activation function of the LSTM is $\tanh$ and the activation functions of both multi-faceted attention layers are linear. The dropout rates of both of the multi-facted attention layers are set to 0.5. We use pretrained 300-dimensional GloVe (Pennington et al., 2014) vectors as our word embedding matrix. We rely on the RMSPROP algorithm (Tieleman and Hinton, 2012) to update parameters for better convergence, with a learning rate of 10^{-4}. The beam size during sentence generation is set to 5. Our system is implemented using the Theano (Bastien et al., 2012; Bergstra et al., 2010) framework.

4.5 Results

Comparison with Existing Baselines: In Table 1, we compare our methods with six state-of-the-art methods: LSTM-YT (Venugopalan et al., 2015b), S2VT (Venugopalan et al., 2015a), TA (Yao et al., 2015), LSTM-E (Pan et al., 2016b), HRNE-A (Pan et al., 2016a), and h-RNN (Yu et al., 2016). Since some of the previous work uses different features, we also run experiments for some of them whose source code was provided by the authors, or we re-implement the models described in their papers, and then evaluate them using our features. The corre-

Model	BLEU-4	METEOR
MFATT-T	0.497	0.318
MFATT-M	0.422	0.304
Cat-TM	0.493	0.317
NFS-TM	0.501	0.322
Fuse-TM	0.502	0.324
MFATT-TM	0.508	0.332
MFATT-TM-SP	**0.520**	**0.335**

Table 2: Results on MSVD.

Model	BLEU-4	METEOR
TA* (Yao et al., 2015)	0.365	0.257
h-RNN* (Yu et al., 2016)	0.368	0.259
MFATT-T	0.367	0.257
MFATT-M	0.361	0.253
Cat-TM	0.366	0.256
NFS-TM	0.370	0.259
Fuse-TM	0.375	0.259
MFATT-TM	0.386	0.265
MFATT-TM-SP	**0.391**	**0.267**

Table 3: Results on MSR-VTT.

sponding additional results are marked with '$*$'.

We observe that our approach is competitive even when just relying on visual attention. Specifically, we consider using only the temporal features and motion features (MFATT-TM), by removing the semantic branch with other components of our model unchanged. To evaluate the effectiveness of different sorts of visual cues, we also report the results of using only temporal features (MFATT-T) and using only motion features (MFATT-M). We find that even just with temporal features alone, we obtain fairly good results, which implies that the attention model in our approach is useful. Combining temporal and motion features, we see that our method can outperform previous work, confirming that our multi-faceted attention layers can extract useful information from temporal and motion features effectively. In fact, the TA, LSTM-E approaches also employ both temporal and motion features, but do not have a separate motion attention mechanism.

As for the h-RNN approach, an important difference is that h-RNNs only consider attention after the sentence generator. Instead, our attention mechanism operates both before and after the sentence generator, enabling it to attend to different aspects during the analysis and synthesis processes for a single sentence.

Effect of Multi-Faceted Attention: We compare multi-faceted attention with other alternatives: 1. Concatenating temporal and motion features at each time step (Cat-TM). 2. Simple multimodal layers, which do not apply feature scaling (NFS-TM) in multi-faceted attention. 3. Applying the model for temporal and motion features separately, averaging the final probabilities of the two models (Fuse-TM). The results are given in Tables 2 and 3. We observe that these variants prove ineffective in comparison with our multi-faceted attention (MFATT-TM). We conjecture that there are several reasons why multi-faceted attention outperforms the alternatives: First, multi-faceted attention can attend to different time periods in a video for temporal and motion features, while Cat-TM cannot. Second, multi-faceted attention smartly converts features to the same scale, while NFS-TM cannot. Third, multi-faceted attention can explore the relationships between temporal and motion features, while Fuse-TM cannot.

Another advantage of multi-faceted attention is that it can deal with semantic attributes. We observe that combining predicted semantic attributes with temporal and motion features (MFATT-TM-SP) obtains better results, which means that our multi-faceted attention can effectively combine features and attributes.

Robustness of Multi-Faceted Attention To further investigate the stability of multi-faceted attention, we first analyze the influence of noise in the features. For this, we randomly select values in the temporal features and replace them with random ones for both the training and test sets, while keeping the motion features unchanged. Figure 4 (top) provides the results with noise on MSVD (MFATT-M-Tn). These results show that even when we provide complete noise as temporal features, the results do not change significantly compared to using only motion features, i.e. omitting the temporal features. Then, we analyze the influence of noise for semantic attributes. For this, we randomly select ground truth semantic attributes and replace them with random ones and keep the temporal and motion features unchanged. Figure 4 (bottom) provides the results

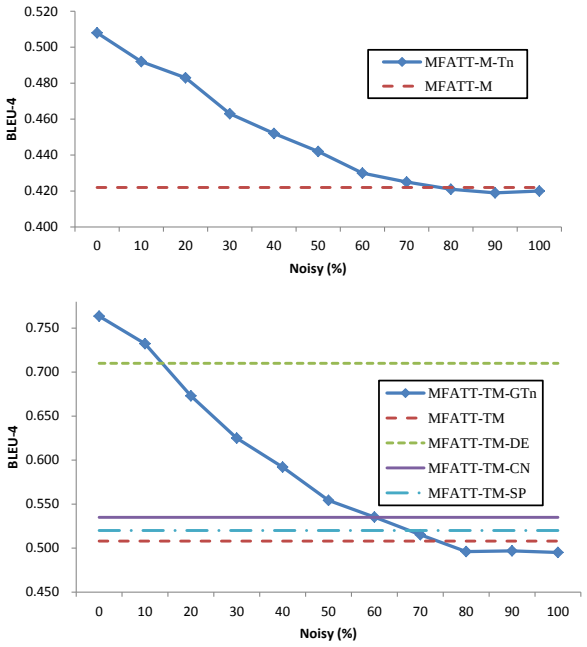

Figure 4: Results of adding noise to temporal features (top) and ground truth semantic attributes (bottom) of MSVD. The blue solids are results of adding noise.

Figure 5: Examples of generated captions on MSVD. GT1 and GT2 are ground truth captions.

with noise on MSVD (MFATT-TM-GTn). These result show that even when we provide completely noisy attributes, the results do not change significantly compared to using only temporal and motion features. We conclude that our multi-faceted attention model is robust with respect to noise in both features and attributes. This shows that we are likely to benefit from further semantic attributes such as tags, titles, comments, and so on, which are often available for online videos, even if they are noisy.

We also consider External Semantic Attributes. The results for German (MFATT-TM-DE) and Chinese (MFATT-TM-CN) are shown in Fig. 4 (bottom). We find that cross-lingual cues lead to significantly improved results, although only 83% of videos have German captions and merely 22% of videos have Chinese captions. We conclude that even just small amounts of tags that may be available in other languages can successfully be exploited by our multi-faceted attention model for captioning.

5 Conclusion

We have proposed a novel method for video captioning based on an extensible multi-faceted attention mechanism, outperforming previous work by large margins. Even without semantic attributes, our method outperforms previous work using visual features. With automatically predicted semantic attributes, our method can obtain better results. We also examined the robustness of our multi-faceted attention and find that its effectiveness remains stable in light of noise in the features and attributes. Moreover, we find that with weaker external signals such as cross-lingual cues, the scores can improve significantly. This opens up important new avenues for future work on exploring the large space of potential additional forms of multi-modal features, including visual and audio features, as well as semantic attributes, including tags, titles, and comments.

Acknowledgments

Gerard de Melo's research is supported by the DARPA SocialSim program.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop*, pages 65–72.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Wardefarley, and Yoshua Bengio. 2012. Theano: New features and speed improvements. In *NIPS Workshop*.

James Bergstra, Olivier Breuleux, Frederic Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: A CPU and GPU math expression compiler. In *Proceedings of the 9th Python in Science Conference*.

Zhu Cao, Linlin Wang, and Gerard de Melo. 2017. Multiple-Weight recurrent neural networks. In *Proceedings of IJCAI 2017*.

Pi Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *WMT*.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

Xinlei Chen and C. Lawrence Zitnick. 2015. Learning a recurrent visual representation for image caption generation. In *CVPR*.

Xinlei Chen, Hao Fang, Tsung Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325v2*.

Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST-8*.

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Gerard de Melo and Gerhard Weikum. 2009. Extracting sense-disambiguated example sentences from parallel corpora. In Gerardo Sierra, María Pozzi, and Juan-Manual Torres-Moreno, editors, *Proceedings of the First Workshop on Definition Extraction in Conjunction with RANLP 2009*, pages 40–46.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Tran Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. C3D: Generic features for video analysis. In *ICCV*.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, pages 179–211.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, pages 1735–1780.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. In *NIPS Deep Learning Workshop*.

Xiang Long, Chuang Gan, Gerard de Melo, Xiao Liu, Yandong Li, Fu Li, and Shilei Wen. 2018. Multimodal keyless attention fusion for video classification. In *AAAI*.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (M-RNN). In *ICLR*.

Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2016a. Hierarchical recurrent neural encoder for video representation with application to captioning. *CVPR*.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016b. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

Ekaterina Shutova, Niket Tandon, and Gerard de Melo. 2015. Perceptually grounded selectional preferences. In *ACL*.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, and Raymond Mooney. 2015a. Sequence to sequence – video to text. In *ICCV*.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL*.

Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. 2016. Improving LSTM-based video description with linguistic knowledge mined from text. In *EMNLP*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Paul. J. Werbos. 1990. Backpropagation through time: What it does and how to do it. In *Proceedings of the IEEE*.

Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. 2015a. A multi-scale multiple instance video description network. *arXiv preprint arXiv:1505.05914v3*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015b. Show, attend and tell: Neural image caption generation with visual attention. In *CVPR*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

Li Yao, Atousa Torabi, Kyunghyun Cho, and Nicolas Ballas. 2015. Describing videos by exploiting temporal structure. In *ICCV*.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *CVPR*.

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*.