Sequence To Sequence - Video to Text

Venugopalan, Rohrbach, Donahue, Mooney, Darrell, Saenko

Presented By: Sanat Sharma
Motivation

Generate accurate captions for a video, something that didn’t have much work done on at the time of paper release.

Utilize Seq-2-Seq models for caption generation on videos, which was also not implemented at the time.

For the most part, short videos depicting one task (for the most part) are utilized.
Sequence To Sequence Models

Sequence To Sequence models are powerful paradigms. In their simplest form, comprised of an encoder, a decoder and an intermediate form.

Initially used extensively in language translation.
S2VT

MultiModal Approach - targets the temporal aspects of videos, in addition to visual features of video frames

Encoder Features

  Image Flow Model

  RGB Image Features Model

The Image and/or flow features are sent to an LSTM encoder which encodes them and passes the hidden state output to a decoder.

The encoding and decoding of the frame and word representations are learned jointly from a parallel corpus
S2VT Features

Handle a variable number of input frames

Learn and use the temporal structure of the video

Learn a language model to generate natural grammatical sentences
RGB Encoder Model

CNN reads in frames and generates features based on input frame’s intensity values.

A stacked LSTM uses the image features and encodes the frames one by one.

Once all frames are read, the model generates a sentence word by word.
Flow Encoder Model

Flow can help incorporate important temporal aspects of the video/set of frames.

To model the temporal aspects of activities typically shown in videos

- Compute the optical flow between pairs of consecutive frames.

- The flow images are also passed through a CNN and provided as input to the LSTM. Flow CNN models have been shown to be beneficial for activity recognition.
Combined Model

Use Shallow fusion technique to integrate flow and RGB features.

At each step of decoding, model proposes set of candidate words.

Rescore hypothesis with weighted sum of score from Flow and RGB networks.
Combined Model (contd)

Calculate score for each word in decoding step. For each word, $p(y_t = y')$ find score as

$$\alpha \cdot p_{rgb}(y_t = y') + (1 - \alpha) \cdot p_{flow}(y_t = y')$$

The Hyper-parameter $\alpha$ is tuned on the validation set.
Model Training

Encoder LSTM receives sequence of frames, encodes them and passes hidden state to the decoder.

For RGB model, a VGG 16 network, pretrained on the ImageNet dataset is used -
Activations from the second last layer (fc6) are used and passed to first LSTM layer.
Model Training (contd)

Flow Encoder uses AlexNet architecture to classify images into 101 categories. Again features from the second last layer are utilized.

1. Train CNN on Activity classes

2. Use optical flow to extract flow images.

3. Take activations from layer before classification

Slide Credit: Subhashini Venugopalan
Model Training (contd)

Decoder receives <BOS> tag as well as the encoder hidden state.

Decoder estimates conditional probability of output sequence given an input sequence using Softmax over all the words in the output vocabulary

\[
p(y_1, \ldots, y_m|x_1, \ldots, x_n) = \prod_{t=1}^{m} p(y_t|h_{n+t-1}, y_{t-1})
\]

h denotes hidden state
Model Training (contd)

The model tries to maximize the log-likelihood of the predicted output. Log-likelihood is optimized using Stochastic Gradient descent

\[ \theta^* = \arg \max_{\theta} \sum_{t=1}^{m} \log p(y_t|h_{n+t-1}, y_{t-1}; \theta) \]

Note: An important to note is that an explicit <EOS> tag is required to terminate sentence. The decoder needs to learn when to emit the <EOS> tag.
### Examples (Microsoft Video Corpus)

<table>
<thead>
<tr>
<th>Correct descriptions.</th>
<th>Relevant but incorrect descriptions.</th>
<th>Irrelevant descriptions.</th>
</tr>
</thead>
</table>
| ![Image of a man doing stunts](image1)  
*S2VT: A man is doing stunts on his bike.* | ![Image of a small bus and a building](image2)  
*S2VT: A small bus is running into a building.* | ![Image of a man pouring liquid into a pan](image3)  
*S2VT: A man is pouring liquid in a pan.* |
| ![Image of a herd of zebras](image4)  
*S2VT: A herd of zebras are walking in a field.* | ![Image of a man cutting paper](image5)  
*S2VT: A man is cutting a piece of a pair of a paper.* | ![Image of a polar bear](image6)  
*S2VT: A polar bear is walking on a hill.* |
| ![Image of a woman doing hair](image7)  
*S2VT: A young woman is doing her hair.* | ![Image of a cat trying to get a small board](image8)  
*S2VT: A cat is trying to get a small board.* | ![Image of a man doing a pencil](image9)  
*S2VT: A man is doing a pencil.* |
| ![Image of a man shooting a gun](image10)  
*S2VT: A man is shooting a gun at a target.* | ![Image of a man spreading butter on a tortilla](image11)  
*S2VT: A man is spreading butter on a tortilla.* | ![Image of a black clip walking](image12)  
*S2VT: A black clip to walking through a path.* |
Examples (Video in Motion)

S2VT: Someone sits on his bed, his head on his bed, his eyes open and he takes his hand.
GT: Hiking up his pants, his father sits on the bed's edge and leans an arm over someone's legs.
Datasets

- Microsoft Video Description corpus (MSVD)
  - Set of annotated Youtube clips (done via Amazon MT)
- MPII Movie Description corpus (MPII-MD)
  - Short video clips from Hollywood
  - 68000 video clips from 94 movies
  - Semi-automated and crowd-sourced
  - ~1 sentence per clip
- Montreal Video Annotation dataset (M-VAD)
  - 49000 short clips from 92 movies.
  - ~1-2 sentences per clip

<table>
<thead>
<tr>
<th></th>
<th>MSVD</th>
<th>MPII-MD</th>
<th>MVAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>#-sentences</td>
<td>80,827</td>
<td>68,375</td>
<td>56,634</td>
</tr>
<tr>
<td>#-tokens</td>
<td>567,874</td>
<td>679,157</td>
<td>568,408</td>
</tr>
<tr>
<td>vocab</td>
<td>12,594</td>
<td>21,700</td>
<td>18,092</td>
</tr>
<tr>
<td>#-videos</td>
<td>1,970</td>
<td>68,337</td>
<td>46,009</td>
</tr>
<tr>
<td>avg. length</td>
<td>10.2s</td>
<td>3.9s</td>
<td>6.2s</td>
</tr>
<tr>
<td>#-sents per video</td>
<td>≈41</td>
<td>1</td>
<td>1-2</td>
</tr>
</tbody>
</table>
Evaluation

Utilize METEOR - compare exact tokens, stemmed tokens, paraphrase matches, as well as semantically similar matches using WordNet synonyms

- Authors claim to use METEOR since it outperforms other metrics like BLEU, ROUGE-L when references are low.
### Evaluation (Contd)

<table>
<thead>
<tr>
<th>Model</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGM [36]</td>
<td>23.9</td>
</tr>
<tr>
<td>Mean pool</td>
<td></td>
</tr>
<tr>
<td>- AlexNet [39]</td>
<td>26.9</td>
</tr>
<tr>
<td>- VGG</td>
<td>27.7</td>
</tr>
<tr>
<td>- AlexNet COCO pre-trained [39]</td>
<td>29.1</td>
</tr>
<tr>
<td>- GoogleNet [43]</td>
<td>28.7</td>
</tr>
<tr>
<td>Temporal attention</td>
<td></td>
</tr>
<tr>
<td>- GoogleNet [43]</td>
<td>29.0</td>
</tr>
<tr>
<td>- GoogleNet + 3D-CNN [43]</td>
<td>29.6</td>
</tr>
<tr>
<td>S2VT (ours)</td>
<td></td>
</tr>
<tr>
<td>- Flow (AlexNet)</td>
<td>24.3</td>
</tr>
<tr>
<td>- RGB (AlexNet)</td>
<td>27.9</td>
</tr>
<tr>
<td>- RGB (VGG) random frame order</td>
<td>28.2</td>
</tr>
<tr>
<td>- RGB (VGG)</td>
<td>29.2</td>
</tr>
<tr>
<td>- RGB (VGG) + Flow (AlexNet)</td>
<td>29.8</td>
</tr>
</tbody>
</table>

MSVD dataset (METEOR in %, higher is better)
Critique

- Parts of the paper not explained
  - No specification about learning rate for Neural models, activation functions while training model
  - Loss function for backpropagation not specified
  - Calculation of Flow for flow encoder model not specified
  - Mechanism used to convert video into frames appropriately. What type of methodology is used to find similar frames (background subtraction?)

- Does not utilize attention, which has proven to be very effective in other Seq2Seq models
  - The paper mentions GoogleNet, which uses Temporal Attention with HoG, HoF features, and got very similar results to S2VT.
Critique

- "You can’t cram the meaning of a single $&!#* sentence into a single $!#&* vector!" - Ray Mooney
  - However the paper does try to capture features from a sequence of images into a vector
- Teacher forcing has been seen as very effective technique for faster training of decoder. Paper does not try to use teacher forcing (giving gold label to decoder regardless of what the encoder generates)
- METEOR scores while useful, do not provide much context against human capabilities
  - Could have used human participants to generate captions and compare the model performance against humans
Future Work

Better training methods

- Utilizing better attention mechanisms to improve performance
- Teacher forcing, utilizing new word embedding methods such as ELBO, BERT

Utilizing well known corpora (Wikipedia etc) to generate captions for Named Entities. Eg Tom Cruise is eating an apple.

- Furthermore, if a further application is auto-generation of captions for movies, background on the movie and the characters would help to identify them by name.

Investigate using semantic meaning to find relationships between objects while generating captions. An example could be <object, activity, attribute> pairs. This might lead to more detailed captions.
Future Work

Going the other way: creating videos/clips using natural language captions from the same corpus

Hand-labelling videos is time and labor intensive, can we do better?
Questions???

A fluffy corgi being cute