Dense Encoding for Video-to-Text Matching

Jianfeng Dong¹, Xirong Li², Chaoxi Xu², Jing Cao², Xun Wang¹, Gang Yang²

¹Zhejiang Gongshang University
²AI & Media Computing Lab, Renmin University of China

Video to Text (VTT) Task @ TRECVID 2018
**Matching and Ranking Task**

**Task:** given a query video, participants are asked to rank a list of pre-defined sentences.

Given video → Candidate sentences

- a man speaks to audiences indoors
- a boy jumps on a trampoline
- a person skates indoors

Ranked sentences

- a boy jumps on a trampoline
- a person skates indoors
- a man speaks to audiences indoors
Cross-modal Similarity

Key question: how to compute cross-modal similarity?

Video    | Similarity | Sentence
----------|------------|-----------
Athletics make a choreography in gym.

Common space based cross-modal retrieval
Cross-modal Retrieval

Common space based cross-modal retrieval models can be typically decomposed into two modules:

• Data encoding
• Common space learning

video as a sequence of frames
sentence as a sequence of words

A boy jumps on a trampoline
Our Model

Dual Dense Encoding

Common Space Learning

- Image CNN
- GRU
- k=2
- k=5
- GRU
- GRU
- Word2Vec
- Concatenation
- Level-3 encoding
Dual Dense Encoding

By jointly exploiting multi-level encodings, dual dense encoding is designed to explicitly model global, local and temporal patterns in videos and sentences.

Level 1. Global Encoding by Mean Pooling
Level 2. Temporal-Aware Encoding by biGRU
Level 3. Local-Enhanced Encoding by biGRU-CNN

Video Encoding

Dense encoding generates new, higher-level features progressively.

Level 1: Global  Level 2: Temporal  Level 3: Local
Sentence Encoding

Dense encoding for sentences is very similar to the dense encoding for videos.

Level 1: Global  Level 2: Temporal  Level 3: Local

Sentence
A boy jumps on a trampoline

One-hot encoding  word2vec  GRU  GRU  GRU  GRU

$\overline{w}_s$  $\overline{h}_s$  $c_2$  $c_4$
Common Space Learning

We choose VSE++ as the common space learning model. Note the dual dense encoding can be flexibly applied to other common space learning models.

Loss Function

Triplet Ranking Loss:

\[
\mathcal{L}(v, s; \theta) = \max(0, \alpha + S_\theta(v, s^-) - S_\theta(v, s)) + \max(0, \alpha + S_\theta(v^-, s) - S_\theta(v, s)),
\]

How to select negative samples \(s^-\) and \(v^-\):

- Randomly selected samples
- Select the most similar yet negative samples
Word2VisualVec++

- Represent sentences into a visual feature space
- Use the improved triplet ranking loss instead of MSE

## Datasets

<table>
<thead>
<tr>
<th></th>
<th>Dataset</th>
<th>#Videos</th>
<th>#Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train</strong></td>
<td>MSVD</td>
<td>1,970</td>
<td>80,863</td>
</tr>
<tr>
<td></td>
<td>MSR-VTT</td>
<td>10,000</td>
<td>200,000</td>
</tr>
<tr>
<td></td>
<td>TGIF</td>
<td>100,855</td>
<td>124,534</td>
</tr>
<tr>
<td><strong>Validation</strong></td>
<td>tv2016train</td>
<td>200</td>
<td>200</td>
</tr>
</tbody>
</table>
Visual Features

Video frames are extracted uniformly with an interval of 0.5 second.

CNN features:
- ResNext-101: 2,048 dim
- ResNet-152: 2,048 dim

The extracted features are available at: https://github.com/li-xirong/avs
### Ablation Study

Dense encoding exploiting all the three levels is the best.

On MSR-VTT dataset

<table>
<thead>
<tr>
<th>Encoding strategy</th>
<th>Text-to-Video Retrieval</th>
<th>Video-to-Text Retrieval</th>
<th>Sum of Recalls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>Level 1 (Mean pooling)</td>
<td>6.4</td>
<td>18.8</td>
<td>27.3</td>
</tr>
<tr>
<td>Level 2 (biGRU)</td>
<td>6.3</td>
<td>19.4</td>
<td>28.5</td>
</tr>
<tr>
<td>Level 3 (biGRU-CNN)</td>
<td>7.3</td>
<td>21.5</td>
<td>31.2</td>
</tr>
<tr>
<td>Level 1 + 2</td>
<td>6.9</td>
<td>20.4</td>
<td>29.1</td>
</tr>
<tr>
<td>Level 1 + 3</td>
<td>7.5</td>
<td>21.6</td>
<td>31.2</td>
</tr>
<tr>
<td>Level 2 + 3</td>
<td>7.6</td>
<td><strong>22.4</strong></td>
<td><strong>32.2</strong></td>
</tr>
<tr>
<td>Level 1 + 2 + 3</td>
<td><strong>7.7</strong></td>
<td>22.0</td>
<td>31.8</td>
</tr>
</tbody>
</table>
Our Runs

Run 0: dual dense encoding model (single)

Run 1: equally combines eight dual dense encoding models with their last FC layer and visual feature varies

Run 2: equally combines eight Word2VisaulVec++ models with sentence encoding and visual feature varies

Run 3: combines run 1, run 2 and eight VSE++ models with sentence encoding and visual feature varies
## Evaluation Results

<table>
<thead>
<tr>
<th>Run</th>
<th>Model</th>
<th>Fusion</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
<th>Set D</th>
<th>Set E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 0</td>
<td>Dense</td>
<td>×</td>
<td>0.450</td>
<td>0.448</td>
<td>0.430</td>
<td>0.433</td>
<td>0.448</td>
</tr>
<tr>
<td>Run 1</td>
<td>Dense</td>
<td>√</td>
<td>0.505</td>
<td>0.502</td>
<td>0.495</td>
<td>0.494</td>
<td>0.500</td>
</tr>
<tr>
<td>Run 2</td>
<td>W2VV++</td>
<td>√</td>
<td>0.458</td>
<td>0.453</td>
<td>0.448</td>
<td>0.436</td>
<td>0.455</td>
</tr>
<tr>
<td>Run 3</td>
<td>Dense W2VV++ VSE++</td>
<td>√</td>
<td>0.516</td>
<td>0.505</td>
<td>0.492</td>
<td>0.491</td>
<td>0.509</td>
</tr>
</tbody>
</table>
Leaderboard

Our runs lead the evaluation on five test sets.

Set A
Take-home Messages

- Dual dense encoding explicitly modeling global, local and temporal patterns is effective to encode videos and sentence
- Late fusion of multiple models is an important trick

The extracted features are available at: https://github.com/li-xirong/avs

Thanks!