Beyond Crossmodal and Multimodal Models

Task: Video Hyperlinking

Mikail Demirdelen, Mateusz Budnik, Gabriel Sargent, Rémi Bois, Guillaume Gravier

IRISA, Université de Rennes 1, CNRS
Table of contents

1. Introduction
2. Segmentation
3. Representations
4. Runs description
5. Results
6. Conclusion
Introduction
A crossmodal system

In 2016, IRISA used a crossmodal system[1]:

• Segmentation step
  → Get segments from whole videos

• Segments/anchors embedding step:

  \[\text{VGG-19} \rightarrow \text{Avg} \rightarrow \text{BiDNN} \rightarrow \text{Anchor representation}\]

• Comparing and ranking step
  → For each anchor, compare and rank each segment
This system had the best score on P@5
→ Go further with this approach?
Segmentation
Motivation

In 2016, we had around **300,000 segments**
→ Limited number of segments
→ Problems with the overlap

Create more segments!

Some constraints:
→ The segment should not cut the speech
→ They must last between 10 and 120 seconds
The method

With a constraint programming framework:

- Keep all the segments that last between 50 and 60 seconds without cutting the speech
- When there we none, expand the duration between 10 and 120 seconds

1.1 million new segments $\rightarrow$ **1.4 million segments** in total (around 4 times more)
Representations
Our model greatly depends on the quality of the representation of each modality
→ Can we improve them?

**Development set:** each triplet (anchor, target, matching) submitted last year

We extracted/recovered:

- For each anchor, its transcript and one or more keyframes
- For each target, its transcript and one keyframe
Visual Representation

Embedding of the keyframes using different pre-trained CNNs (VGG-19[7], ResNet[2], ResNext[9] and Inception[8])

When multiples keyframes, there was an additional step of  
**keyframe representation fusion:**

- **Single**: Using a single keyframe and discarding the rest
- **Avg**: The embedding is the average of all of the keyframes embeddings
- **Max**: Each feature of the embedding is the maximum of all keyframes corresponding feature
Visual Representation

<table>
<thead>
<tr>
<th>Models</th>
<th>Single P@5</th>
<th>Single P@10</th>
<th>Average P@5</th>
<th>Average P@10</th>
<th>Max P@5</th>
<th>Max P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG19</td>
<td>41.60</td>
<td>41.27</td>
<td>43.40</td>
<td>41.60</td>
<td>42.60</td>
<td>41.03</td>
</tr>
<tr>
<td>Inception</td>
<td>40.40</td>
<td>41.83</td>
<td>41.00</td>
<td>41.39</td>
<td>42.60</td>
<td>41.73</td>
</tr>
<tr>
<td>ResNext-101</td>
<td>41.00</td>
<td>39.37</td>
<td>41.40</td>
<td>40.10</td>
<td>41.80</td>
<td>39.90</td>
</tr>
<tr>
<td>ResNet-200</td>
<td>43.80</td>
<td>41.57</td>
<td>47.20</td>
<td>44.37</td>
<td><strong>47.60</strong></td>
<td><strong>44.87</strong></td>
</tr>
<tr>
<td>ResNet-152</td>
<td>44.40</td>
<td>41.37</td>
<td>45.60</td>
<td>41.67</td>
<td>45.20</td>
<td>40.40</td>
</tr>
</tbody>
</table>

→ We chose to use a *ResNet-200* network and a *Max* keyframe representation fusion method
Same experiments with transcripts:

<table>
<thead>
<tr>
<th>Models</th>
<th>P@5</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Word2Vec[5]</td>
<td>44.2</td>
<td>45.3</td>
</tr>
<tr>
<td>Doc2Vec[4]</td>
<td>38.4</td>
<td>39.4</td>
</tr>
<tr>
<td>Skip-Thought[3]</td>
<td>40.2</td>
<td>41.6</td>
</tr>
</tbody>
</table>

→ We chose to keep *Word2Vec*. 
Runs description
A bidirectional deep neural network (BiDNN) was trained with ResNet as a visual descriptor and a Word2Vec as a textual descriptor: 

→ *BiDNNFull* is our baseline for testing other improvements to the system.
We chose to keep the list of tags as a filter to compare anchors and targets that share at least one tag in common.
However:

- 77% of videos have tags
- They have a mean number of tags of 4.71

Too restrictive?

Use the text of the descriptions:

- Selection of only verbs, nouns and adjectives
- Lemmatization
- Exclusion of stopwords and hapaxes

→ *BiDNNFilter* is the same as *BiDNNFull* but with the addition of the list of keywords—tags and description—used as a filter.
Some issues about the keyframe representation fusion method:
→ Basic treatment of information contained in multiple keyframes

We use the Moore-Penrose pseudo-inverse:

- **Captures a notion of movement** between multiple keyframes
- **Deals with different variations** found across all keyframes.
- It can improve the search quality[6].

→ *BiDNNPinv* is the same as *BiDNNFull* where the Max function is replaced by the pseudo-inverse.
Quantify the usefulness of the BiDNN in this system

We replaced the BiDNN by a L2-normalization followed by a concatenation:

→ NoBiDNNPinv’s embedding pipeline is described by the picture.
Results
### Results

<table>
<thead>
<tr>
<th>Runs</th>
<th>MAP</th>
<th>MAISP</th>
<th>P@5</th>
<th>P@10</th>
<th>P@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDNNFull</td>
<td>13.34</td>
<td>10.14</td>
<td>68.80</td>
<td>71.20</td>
<td>42.40</td>
</tr>
<tr>
<td>BiDNNFilter</td>
<td>10.81</td>
<td>8.43</td>
<td>76.00</td>
<td>74.40</td>
<td>38.00</td>
</tr>
<tr>
<td>BiDNNPinv</td>
<td>15.29</td>
<td>11.52</td>
<td>75.20</td>
<td>74.40</td>
<td>43.40</td>
</tr>
<tr>
<td>noBiDNNPinv</td>
<td>12.46</td>
<td>10.16</td>
<td>72.80</td>
<td>73.20</td>
<td>39.60</td>
</tr>
</tbody>
</table>

- *BiDNNFilter* obtained the best P@5 and P@10 showing the interest of the filter to increase precision.
- *BiDNNPinv* obtained the best MAP, MAISP and P@20 showing the pseudo-inverse gives more precision stability.
- The score difference between *BiDNNPinv* and *noBiDNNPinv* confirms the relevance of the crossmodal model.
Conclusion
Adding a filter increases the precision

The pseudo-inverse succeeds at capturing relevant information on multiple keyframes

We can think of future interesting developments:

- Combine both the filter and the pseudo-inverse
- Incorporate the metadata within the neural network, using it as a third modality
- Use the pseudo-inverse on both anchors and targets
Thank you for your attention!

*Irisa at trecvid2016: Crossmodality, multimodality and monomodality for video hyperlinking.*

In *Working Notes of the TRECVis 2016 Workshop*, 2016.

K. He, X. Zhang, S. Ren, and J. Sun.

*Deep residual learning for image recognition.*


*Skip-thought vectors.*

Q. Le and T. Mikolov.  
**Distributed representations of sentences and documents.**  

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean.  
**Distributed representations of words and phrases and their compositionality.**  

R. Sicre and H. Jégou.  
**Memory vectors for particular object retrieval with multiple queries.**  
K. Simonyan and A. Zisserman.  
Very deep convolutional networks for large-scale image recognition.  

Going deeper with convolutions.  
In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He.  
Aggregated residual transformations for deep neural networks.  
Some good/bad cases

BiDNNFilter:

**Good cases**

- anchor_131: good description + tags
- anchor_132&137: good description with no tags

**Bad cases**

- anchor_124: very general tags \(\rightarrow\) not better than BiDNNFull
- anchor_126: only three tags that do not describe the video (grit, grittv, laura_flanders)
- anchor_141: no tags and a very long description (709 words)

BiDNNPinv:

**Good cases**

- anchor_141: an anchor with a lot of keyframes?

The **bad cases** are hard to identify
Moore-Penrose pseudo-inverse

Given a set of anchor vectors represented as columns in a $d \times n$ matrix $X = [x_1, ..., x_n]$ where $x_i \in \mathbb{R}^d$:

$$m(X) = X(X^TX)^{-1}1_n$$  \hspace{1cm} (1)

where $1_n$ is a $n$ dimensional vector with all values set to 1.