Abstract

Sorting images in an album is an interesting task in computer vision and machine learning. It allows us to build systems that can generalize and exploit the temporal relationship between images. We thus, explore this task by implementing various models and reporting the performance on the Visual Storytelling (VIST) dataset, which also includes associated captions of the images. Previous implementation of pairwise comparison model fails to capture the overall context of the story. We hence, propose a more intuitive approach to this problem taking into account and using a latent representation of the input (shuffled) sequence, which uses Pointer Network (ptrNet) to utilize the whole contextual information to perform story ordering/sorting. We display the effectiveness of this model through various experiments. We make the source code available on Github\(^1\) and a live demo of the best model here\(^2\).

1. Introduction

The ability of sequencing objects and tasks is innate in humans and comes much naturally with minimal supervision. A child learns to stack blocks in decreasing order of size to form a tower. Sequencing also has applications to sorting words (Linearization\(^10\), as it is termed in NLP), according to some grammatical rules (for instance, in a sentence *We are coming home*, grammatical rules dictate the ordering of individual words). Another popular example of sequence sorting where humans excel is at forming coherent story from a jumbled set of sentences or images, which is widely used in activity or event recognition. While sorting for absolute values has seen huge leaps in research, sorting for complex and abstract sequences like images/audio is yet to take a formidable shape. The advent of Deep Learning, which is a complex system of artificial neural networks has seen a massive improvement in the performance of various machine learning related tasks like Image and Speech Recognition. In our project, we aim to solve the problem of sorting abstract sequences using a neural model. We hope to build a system that can look at images and arrange them in a way that makes sense, like how humans arrange photo albums and to achieve this without relying on metadata (such as date uploaded) and taking help from complementary modalities like associated captions. A system that can sort images temporally/sequentially can also be used to sequence images in children’s book to generate a proper story, or can be used in a crowdsourced album for major events like New Years Eve or even the Oscars. Snapchat has a similar functionality, but this is done manually; they have a team that curates snaps from different users to create a story. Our goal is to learn the temporal structure of the entities through artificial neural networks. This leads to temporal common sense\(^1\) such that the neural network can generalize the sequencing tasks. We explore sequence-to-sequence (seq2seq) models and a recently proposed neural architecture called Pointer Network (ptrNet) for the task of ordering. From this implementation of image sorting with the help of associated captions, we hope demonstrate the model’s ability to learn a temporal sequence.

2. Related Work

Chen et al. (2009)\(^3\) use a generalized Mallows model for modeling sequences for coherence within single documents. Recently, Mostafazadeh et al. (2016)\(^8\) presented...
the ROCStories dataset of 5-sentence stories with stereotypical causal and temporal relations between events. In our work though, we make use of a multi-modal story-dataset that contains both images and associated story-like captions. Some works in vision (Pickup et al., 2014[9]; Basha et al., 2012[2]) also temporally order images; typically by finding correspondences between multiple images of the same scene using geometry-based approaches. Similarly, Choi et al. (2016)[4] compose a story out of multiple short video clips. They define metrics based on scene dynamics and coherence, and use dense optical flow and patch-matching. In contrast, our work deals with stories containing potentially visually dissimilar but semantically coherent set of images and captions.

We draw inspiration from Agrawal et al., [1] to learn temporal common sense from multi-modal stories consisting of a sequence of aligned image-caption pairs, and thus start by implementing the pairwise-model mentioned in [1] obtain baselines. We also introduce an implementation of Pointer Networks [11, 12, 5] on this task and compare with the baseline implementation.

3. Methodology

Here, we describe our approach toward gathering the sequence of stories and also explain the training procedure.

3.1. Data Collection

We use the Visual Storytelling (VIST) dataset for our project. VIST has 81,743 unique photos in 20,211 sequences, aligned to both descriptive (caption) and story language. The VIST dataset has three tiers of language for the same image. (1) Descriptions of images in isolation (DII); (2) Descriptions of images in sequence (DIS); and (3) Stories for images in sequence (SIS). This tiered approach reveals the effect of temporal context and the effect of narrative language. We are more interested in stories from images-in-sequence as the captions are able to maintain relations with previous frames thereby conveying a story. The VIST dataset is split into albums and each album has 5 images each arranged in a way to convey a cohesive story. We collect 40,000 unique stories, and use 30,000 for training and 10,000 for testing.

3.2. Input Features

We use VGG16 pre-trained model trained on ImageNet. We extract features of 4096 dimensions from its penultimate layer to encode the images. To encode sentences, we pass them to skip-thought vectors [6]. That is, instead of using a word to predict its surrounding context, we encode each sentence with respect to its neighboring sentence. Thus, any composition operator can be substituted as a sentence encoder and only the objective function becomes modified. The vectors are a numpy array with as many rows as the length of sentences, and each row is 4800 dimensional (combine-skip model, from the paper) with the uni-skip and bi-skip models contributing to 2,400 dimensions each.

We explore three models: Visual-only, Text-only and Visual and Text features combined, and report performances for all these models in the later section. In our multi-modal approach we concatenate the image-caption features represented by a 8896(4096 + 4800) dimensional vector.

3.3. Pairwise Model

For a story, the images along with its captions are in sequence. We shuffle them and take all possible combinations of image and/or caption pairs. We develop pairwise scoring models that given a pair of elements (i, j), learn to assign a score: S(σi < σj) — i, j indicating whether element i should be placed before element j in the permutation σ. Here, [ ] indicates the Iverson bracket (which is 1 if the input argument is true and 0 otherwise). Thus, this is treated as a binary classification problem. Figure 2 illustrates the model when we take both the image and caption features. There can be 20 combinations of pairs given a story with five elements. Among these 20 vectors, we randomly sample 16 of them (every epoch, for every sample) and then shuffle these vectors so as to minimize any correlation between these input vectors, and train the model accordingly.
3.4. Pointer Networks

Our proposed methodology uses pointer networks (proposed by Vinyals et al. (2015) \[12\]) to solve the sorting problem. Pointer Networks are useful when the output is taken from the input sequence. In our approach we will be using the following networks: An encoder LSTM and a decoder Pointer Network inspired from the architecture proposed in \[5\].

1. The features are passed through an encoder LSTM, which is fed \(x_i\) at each time step, \(i\), until the end of the input sequence is reached.

2. At each step, the network produces a vector that modulates a content-based attention mechanism over inputs. The output of the attention mechanism is a softmax distribution with dictionary size equal to the length of the input. This output corresponds to the index at the output sequence of each entity in the input sequence.

For each story, we store the original sequence, and additionally shuffle and store the sequences to ensure the model eventually sees all possible permutations of all the stories in the training set over many epochs. In this case, the output will be five one-hot vectors corresponding to the desired position of each element in the input sequence.

We use ADAM optimizer at a learning rate of 0.01 decayed by a factor of 0.0001 every epoch for both the above-mentioned models.

4. Experiments and Results

We report the Spearman’s Correlation and Kendall’s Tau metrics to evaluate our results. Kendall’s tau \(\tau\), computed as

\[
\tau = 1 - 2 \times \frac{n_{inv}}{N(N-1)}
\]

where the \(n_{inv}\) is the number of inversions of pairs in the predicted sequence with incorrect relative order and \(N\) is the length of the sequence, which is 5 in our case. A \(\tau\) score of 0.5 means that half of the values matched within a given story and its predicted sequence.

If the the ranks are represented as distinct integers, which is the case, Spearman’s rank correlation \(r_s\) is given by the following:

\[
r_s = 1 - \frac{6 \sum d_i^2}{N(N^2 - 1)}
\]

where, \(d_i\) is the difference between the input and predicted rank sequence, and \(N\) is the number of observations, which is 5 in our case.

It is observed that the pairwise model is not able to capture the contextual information of the input sequence and is thus, unable to achieve a high score. The pointer network was able to overcome this shortcoming and was able to perform significantly better at qualitative experiments by capturing the intrinsic temporal intent of the stories and leverage that to sort the shuffled input.

Out of the six sets of experiments, it was found that concatenating visual and textual modalities performs better than using modalities individually. Of the two models trained, pointer networks performs significantly better than pairwise comparison model. The pairwise gives a \(\tau\) score of 42\% whereas the pointer network gives a score of 60\%; which is a considerable improvement from the pairwise model, however, not as great as expected. A score of 60\% means that approximately 3 out of 5 entities are almost correctly placed in the predicted output sequence.

Table 1 shows the values of the metrics for the models and modalities experimented for this task.

<table>
<thead>
<tr>
<th>Model</th>
<th>(\tau)</th>
<th>(r_s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.24</td>
<td>0.20</td>
</tr>
<tr>
<td>Pairwise-V</td>
<td>0.28</td>
<td>0.24</td>
</tr>
<tr>
<td>Pairwise-T</td>
<td>0.40</td>
<td>0.38</td>
</tr>
<tr>
<td>Pairwise-VT</td>
<td>0.42</td>
<td>0.40</td>
</tr>
<tr>
<td>PrNet-V</td>
<td>0.46</td>
<td>0.36</td>
</tr>
<tr>
<td>PrNet-T</td>
<td>0.54</td>
<td>0.48</td>
</tr>
<tr>
<td>PrNet-VT</td>
<td>0.60</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Table 1. Spearman Correlation\(r_s\) and Kendall’s Tau \(\tau\) values on Test set

Figure 4 shows the Average Spearman correlation for the predicted sequences across the validation set for every 100 epochs. Figure 5 illustrates the output attention weights for an input that is already sorted. The intent was to visualize the
output values for a sorted input to see if the model is able to understand that it need not change the input at all. Although not reported, the attention map starts with uniformly distributed values for each of the 25 spots on the map (value of 0.2), and converges to identity matrix (which is also the expected ground truth for an already sorted input). The model learns to place the first entity in the sequence with very high confidence. It would be interesting to see the performance of the model using Bi-LSTMs or Stacked LSTMs to assess if it can learn to place entities into other positions as high a confidence.

![Attention Map for Pointer Network](image)

Figure 5. ptrNet’s output weights for an already sorted input

We see that the model learns to predict the entities that appear toward the beginning and the end with a high confidence, but kind of softens the values for the entities that appear in the middle. Since the model is more influenced with the language component, we believe that certain words could contribute to teaching the model to place entities in certain positions. This is also discussed and well illustrated in [1], where, words like “overall” and “lastly” prove to be discriminatory in assigning positions to entities in a story.

Acknowledgement

We would like to thank Professor Vicente Ordonez for his constant motivation and guidance and extend acknowledgment to Abhimanyu Banerjee for his useful insights.

5. Conclusion and Future Work

In this report, we address the task of sequencing shuffled stories from the corresponding image and caption pairs. To solve this, we deploy two models: a pairwise comparator model, much like the classic numeric comparison “less than” operator but now works in the temporal sense, and extended this model to sort the story; and a pointer network that works like a seq2seq model but directly outputs the expected index of the input elements in the output as an attention map. We observe that visual features are not very robust for this task. It seems so that even pointer networks are not able to fully excel in this task, but it does show promise toward furthering this model. We certainly believe that the Pointer Network is the appropriate choice of neural architecture for this task. Also, there is immense scope to improve the performance of the visual modality alone. As we proposed earlier, it will certainly help to represent the visual modality in an activity context embedding. This is something we will surely try to see if it alleviates the current model. We could also train the Skip thought embeddings instead of using pre-trained weights. With a clever design, it will be possible to perfect this task using models inspired by Pointer Networks.

References

Figure 6. Correct Predictions by the Model
Figure 7. Incorrect Predictions by the Model