Visual Story Cloze Test: Predicting the end of a visual story

Athanasios Efthymiou
University of Amsterdam
athanasios.efthymiou@student.uva.nl

Athanasios Roidis
University of Amsterdam
athanasios.roidis@student.uva.nl

Georgios Sidiropoulos
University of Amsterdam
george.sidiropoulos@student.uva.nl

Abstract

Understanding the temporal occurrence of a sequence of events is an up-and-coming task in the AI field since it requires understanding of high-level concepts. In this work we introduce a new task for evaluating story understanding, the Visual Story Cloze Test, which is visual counterpart of a regular Story Cloze Test. This test requires a system to choose the correct ending to a five-image story from a set of five candidates. Specifically, we want to examine whether deep neural networks are capable of learning the temporal dependency that lies behind a story sequence. In this work, we propose a model for this particular task which is based on Convolution and Recurrent Neural Networks. We also propose two methods for the creation of the dataset needed for this task, built upon the Visual Storytelling Database on which we train and test our model. Experimental evaluation shows that the proposed model can achieve reasonable results with respect to the difficulty of the task, meaning that the task is learnable.

1. Introduction

The worldwide and extensive usage of photo-taking devices in combination with the outburst in social networks, led to an explosion in image sharing and the dramatical increase of visual data. Users of social networks are overwhelmed by the vast amount of available images, and are struggling to keep track of various pictures taken. Since images can be thought of as sequential data, models that can reason with this information in an efficient and intelligent way are in great demand.

A system capable of reasoning on temporal visual information could give rise to a multitude of applications. For example, given an album of images that take place over a specific time period, it would be possible to summarize the most important events by picking the images in a way that it resembles a visual story, or the system could even predict the future events at real time while the users are taking pictures, so that it will provide recommendations for future plans before the users even ask for it.

In this study, we focus on learning the temporal patterns of visual stories. An example of a visual story can be seen on figure 1. Each story is comprised of 5 sequential images, paired with their narrative caption concerning the content of the image within the context of the story. On the first image we can see a man getting prepared for a trip, before he goes on a walk to the forest to reach the campsite (images 2-3-4), while the story finally ends with a campfire at night. As it can be seen from this example, there is a clear, inherent temporal common sense structure behind a story and a three act structure, meaning that each story should have a setup, a middle and a climax.

1. [male] thought gearing up for camping was a big hassle
2. and then there was all the walking
3. the packs were heavy, the trail was long
4. but when he made it to the campsite he enjoyed seeing his friends
5. and sitting around the campfire

Figure 1. A Visual Story with annotations

Therefore we would like to investigate whether some of the more sophisticated and robust deep networks can capture this regularity of the temporal relations in a given story thought the *Visual Story Cloze Test*. This task requires a system to choose the correct ending to a sequence of five images depicting a story. Important to not here is that in

this work we are going to use only visual information, since textual annotations would rarely be available in a real-life application. In more detail, for each four-image sequence the model should be able to choose the true ending of the sequence from five possible endings, from which only one is correct. Figure 2 illustrates the formulation of this task. The story starts with an image of some food ingredients, which are then prepared to be cooked and while an image of a woman putting the food on the dishes is shown, but we don't know how this sequence ends. Even though this story could end in many ways, one would reasonably think that what follows is an image of people eating the food, and that is what we would ideally like the network to learn by choosing the correct ending from the five candidates. this obviously requires from the model to be able to capture the underlining temporal pattern on the story.

Figure 2. The problem

Following this brief introduction, in section 2 we will provide a brief review of the preceding studies which are related to this work. In section 3 we describe the models that we used to learn the temporal patterns in a Visual Story, while in section 4 we present the Dataset in which this study was built on. As the set up of the network that we used for our model is non-trivial, in section 5 we provide a description of a Diagnostic Experiment on synthetic data that we conducted in order to check for the validity of our proposed model. Subsequently, in section 6, we illustrate the experiments and the results that we obtained. Finally, in section 7 we discuss about the conclusions that can be drawn from the outcomes of this work and any additions that could be implemented in a future work.

2. Related Work

Several lines of research have focused on the challenging task of learning temporal patterns through the textual Story Cloze Test. Learning temporal dependencies has a rich history especially in NLP research. A number of recent works investigate the influence of the textual information combined with the visual information in the demanding task of learning temporal relations and properties from a given story. Partly inspired by this recent research this work significantly differs from them due to the fact that we do not take advantage of the textual information of a given story, but we only focus on the visual part.

Perhaps the most crucial and closely connected work for this study is the one of Huang *et al*. [5] as their work basically builds on the sequential vision-to-language dataset they introduced and that is also used in our work. Besides explaining the construction procedure of the dataset and analyzing it, they also introduce a baseline for the generation of the textual story given the whole image sequence, as well as the isolated descriptions of those images.

Accordingly, another interesting study related to our work might be the one conducted by Agrawal. *et al*. [1] whose motivation is also to enable AI systems to better understand and predict the temporal nature of events on the same dataset the we use in our experiments. By using a combination of neural networks they propose a solution based on a learning-to-rank algorithm for their newly introduced task, "visual story sequencing", which is the task of reordering a set of 5 image-caption pairs in the right order. Even though their work resembles our study, besides the difference of the two tasks, their approach is heavily text-based with image-based features only providing complementary improvement.

Several groups of researchers in the Computer Vision area are more and more focusing on predicting the future in images and videos. Probably the most similar to our work is Vondrick *et al*.'s [8], in which they try to predict actions one second in the future and objects five seconds in the future in a large video collection. Although our task is very similar to theirs, and the approach to some extend given that they also mostly focus on the visual representation of images, it differs to ours due to the fact that their algorithm operates on video streams and consequently the duration of the sequence and the time interval between two consecutive frames is much smaller than ours. This means that their model mostly has to deal with human-actions and the movement of objects between frames, whereas we would like our model to try to comprehend on a higher level since in our study we work on stories containing in many cases visually dissimilar but semantically coherent sets of images.

Kim and Xing [6] also operate on a huge dataset, but they focus on storyline reconstruction and image recommendation tasks. In contrast, to our work they incorporate the meta-data of the images as temporal side information in order to learn these temporal dependencies.

3. Approach

Our model rather than predicting the true ending of a story given the first four images and a set of five candidates, it is treating the problem as a binary classification task; given a sequence of 5 images it tries to predict whether this is a true story or not. Therefore, at train time it learns to assign a probability of 1 to the true sequence, while for the

other 4 false stories, created from the first 4 images of the true sequence and each one of the false endings as the 5th, it assigns a probability of 0. At test time, it compares the probabilities of those 5 stories to be true and it assigns as the true ending the one with the highest probability of being true. In the following section we will describe the architecture of our model, which is based on a combination of Convolutional Neural Networks (CNN) for feature extraction and Recurrent Neural Networks (RNN) for the actual prediction.

3.1. CNN - Feature Extraction

A Convolution neural network is a type of feed-forward neural network that is based on the convolution operator and has been proven very effective in areas such as image recognition and analyzing visual imagery in general. The first part of the network is comprised by a combination of alternating convolution and pooling layers that are responsible for transforming the data in a way to capture task specific features, thus they operate as feature detectors, while the second part is a sequence of fully connected layers that perform the classification task. It has been shown that a pretrained CNN on image classification can operate as a general feature detector that can also be used for other problems by using one of its fully connected layers. This is known as transfer learning.

For our experiments we use a VGG16 network, which is a deep convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper Very Deep Convolutional Networks for Large-Scale Image Recognition [7] . The model achieves 92.7% top-5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes. From this network we use the features from the last fully connected layer, before the softmax activation function that returns the probability distribution, known as the fc8 layer. The architecture of the network can be seen on figure 3

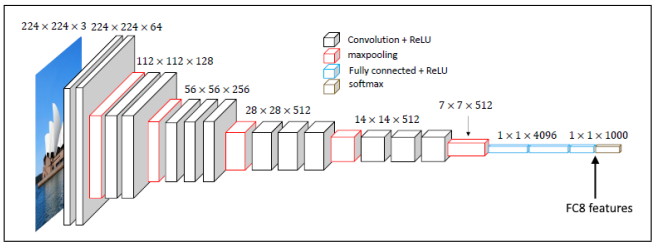


Figure 3. The architecture of a VGG16.

3.2. Recurrent Neural Networks

On the other hand there are the Recurrent Neural Networks, which are a class of artificial neural network where connections between units form a directed cycle. While in traditional neural networks all inputs are independent of each other, this cyclic connectivity of the RNNs allows it to have some sort of memory which captures information about what has been calculated so far, thus allowing it to exhibit dynamic temporal behavior. This memory is often called "hidden state" and it is updated every time a new input is fed through the network on a new timestep, but is reset before a new sequence of inputs. The architecture of a basic RNN can be seen on figure 4 in which the hidden state is given by the variable 's'

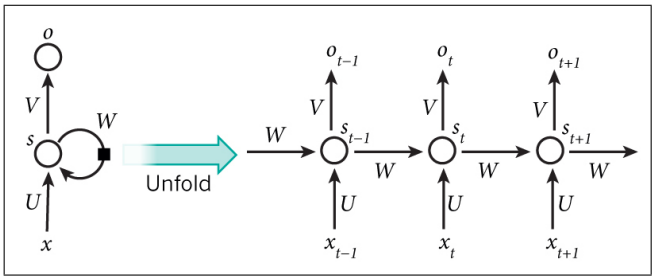


Figure 4. A simple RNN in its unfolded form.

Long short-term memory (LSTM) networks [4] are a particular subclass of RNNs which have become very common and yield state-of-the-art results in many tasks that require temporal behavior. This is because of their ability to learn long-term dependencies with the addition of 3 "gates" that control information flow. In more detail:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

where x_t is the input vector, h_t the output vector, c_t the cell state vector (hidden state) and f_t, i_t, o_t are the forget, input and output gate vectors respectively, with W, U, b being their parameters.

3.3. Network architecture

Thus, our proposed model is a combination of the two aforementioned types of neural networks, as it can be seen on figure 5 At its core, it is a regular RNN that operates on a fixed amount of 5 timesteps, one for each image in the story. But instead of accepting the raw pixels of the image itself, the image is first fed through a VGG16 in order to extract high-level features from the fc8 layer, and then these features are used instead as an input to the RNN. This way the RNN operates a higher concept level and it doesn't need to detect objects in the scene itself, since those are given by the VGG, but it only needs to find how these objects interact within the context of a story. At each timestep the hidden

state of the network contains information about what has happened so far in the story, and once it has gone through the whole story it tries to predict from the final state if the story was true or not. Even though we use the fc8 layer of the VGG and an LSTM on figure 5 , this model is quite flexible since any layer or even any other network can be used for feature extraction and any recurrent architecture can be used for the final prediction. At this point it should be noted that the VGG is not fine-tuned and its weights are kept frozen, therefore it is only used as a preprocessing step for feature extraction.

4. Task and dataset specification

4.1. Visual Storytelling dataset

We train and evaluate our model on the Visual Storytelling dataset (VIST), the first dataset for sequential vision-to-language created by Microsoft Research. It includes 10,117 Flickr albums with 210,819 unique photos which are part of the YFCC100M release, all under a Creative Commons license. Each album is also labeled with which topic it belongs to (birthday, wedding etc.), while there 69 distinct topics in total.

These individual images were then presented to crowd workers using Amazons Mechanical Turk to collect the corresponding stories and descriptions, as the one shown on figure 1. More specifically, some workers were tasked with selecting a subset of five photos from a given album to form a photo sequence and write a story about it, while other workers were tasked with writing a story based on one photo sequence generated by other workers. In total 50200 stories were created and were split 80/10/10% between the training/validation/test sets.

4.2. Candidate selection

But in order to be able to use this dataset for our ending prediction task we will also need to find four false alternative endings for each story, since the dataset only contains the true sequence of images. Ideally we would like to emulate a multiple choice question in which some alternative answers are easier to identify as being wrong and can be quickly ignored, while others are more similar to the true answer, so that the ability of the model to truly can be tested.

Therefore we propose two methods for sampling candidate endings; a simple approach in which we sample four random images from the story's album, and a more sophisticated approach, which is based on heuristics, that chooses four images based on the similarity that they have with the true ending.

The former approach is pretty straightforward; four images are randomly picked from the same album that the story was created but are also not part of the story. It is easy to see why some of the those images might be similar to the true ending, since they are from the same album and will probably take place in the same area, involving the same people. The opposite might also be true though; because of the diversity of some albums and the fact that we do not enforce any other form of similarity, they might be irrelevant.

For the second approach we directly try to choose one hard, two moderate and one easy alternative ending based on the similarity of the images with the true ending, as well as the album that they belong in. Again, in order to measure the similarity of two images, we use the features extracted from the fc8 layer of the VGG. More specifically, we choose the four images using the following rules:

- **Hard candidate:** Closest image to the true ending and from the same album, but distance $>$ threshold θ_1
- **Two moderate candidates:** Closest images to the true ending but from a different album of the same topic and distance $>$ threshold θ_2
- **Easy candidate:** Sample one image from the 100 most distant images

The reason that the distance between the true ending and the alternatives should be over a manually defined threshold is so that too similar or even duplicate images are discarded. Without this additional constraint it would be incredibly difficult for the model to disambiguate between the similar candidate options, but would also be even harder to train it because it would learn that one input maps to two different outputs, since the candidates might be close to identical.

On figure 6 an example of the four candidate endings chosen by the two different sampling methods is shown for the story given in figure 2. As it can be seen, the random sampling produces two visually very similar images with the true ending, but also two unrelated images. On the other hand, the heuristic method produces candidates that are getting gradually easier, from the hard candidate that shows the preparation of the food, which is very similar to the true ending that shows people eating, to the two moderate candidates that show people next to a table with foods and two people with drinks, to the easy candidate that simply shows a firework and is completely irrelevant.

5. Diagnostic Experiment on Controlled Dataset

Recurrent neural nets are known to be sensitive to their hyper-parameters, namely the number of hidden units, the number of hidden layers, the batch size etc. as well as to the dataset (scale of data). Therefore, a diagnostic experiment on simpler, synthetic data took place in order to check the validity of the proposed model.

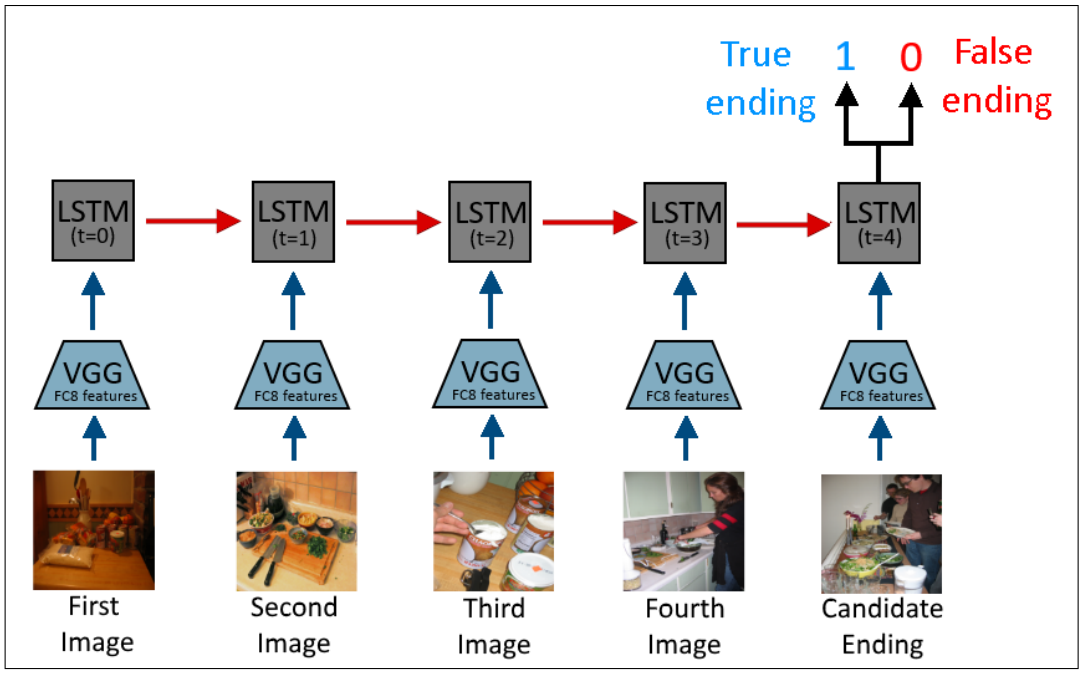


Figure 5. Our proposed model which uses a combination of a CNN and an RNN.

Figure 6. Alternative endings using random sampling (top) and the heuristics method(bottom) for the story of figure 1

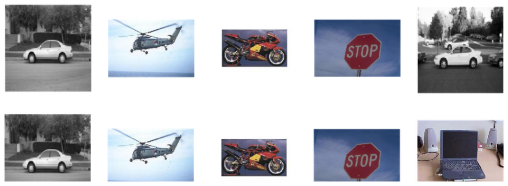


Figure 7. Example true (top) and false (bottom) story.

For this purpose we used *Caltech 101* to create a dataset consisted of pseudo-stories; pseudo-stories follow the already mentioned five image sequence principal. These pseudo-stories follow various different temporal patterns, mainly focusing on two different categories, animals and vehicles. The main underlying pattern that should always hold is that the first and the last image of the sequence must be of the same class. Moreover, the intermediate images of the sequence should be of different classes both among themselves and with respect to the first and last image. For each generated pseudo-story we create four false endings. To do that we sample three images from the same category of each corresponding pattern (animals or vehicles, hard endings) and one object from the other available categories (easy ending). By sampling three hard negatives and an easy one, we can better approximate the real problem. In Figure 7, an example of a true and a false story can be found; in this story the first image of the sequence belongs to *car* class, followed by a *helicopter* a *motorbike* and a *stop sign*, therefore the sequence should end with a car in order to be a true story.

That said, the temporal pattern that we want our model to capture is the relation between the class of the first and the last image in the sequence. In other words, we want our model to distinguish as true those stories that start and end with an image from the same class. To validate our approach we used an LSTM with 256 hidden units and two hidden layer. The accuracy of this particular model was 99.5%.

At this point it should be noted that this toy task doesn't necessarily prove that the real task we are trying to achieve is learnable, but rather it gives us a rough estimation of the values that the model's hyperparameters should take.

Parameters	Values
Hidden units	256
Hidden layers	2
Batch size	100
Learning rate	0.001
Optimizer	Adam Optimizer on cross-entropy

Table 1. Setup

Model	Sampling-Random	Sampling-Heuristics
RNN	27.6%	31.6%
GRU	27.7%	37.2%
LSTM	28.1%	39.6%

Table 2. Results w.r.t model and sampling technique.

6. Experiment

In this section the results for the different experiments are presented. For the experiments, three different models where tested; LSTMs, RNNs and Gated Recurrent Unit (GRU) [2] networks, which are based on the same basic idea as the LSTMs, but only use two gates. The main setup that was used for the three models can be found in Table 1. The models were trained for 10 epochs and from those 10 epochs only we kept the weights after the epoch that gave the best results on the validation set. This generally ranges from 4 to 8 epochs, therefore the training can take from several minutes up to an hour, based on the computational capabilities of the machine. Before presenting the results, we should underline that the accuracy for randomly choosing the true ending of a story is 20% (five possible ending candidates).

Table 2 presents the final results on the test set for the three different models and the two different sampling techniques. It is clear from the results presented that there is a noticeable difference in the performance with respect to the sampling technique that was followed in each experiment. When sampling with heuristics, the accuracy scores are much higher (approximately 10% increase for the case of LSTM and GRU). However, it should be pointed out that the aforementioned increase was expected since with random sampling we are mostly sampling hard negatives. Again, this is due to the fact that for random sampling we sample from the same album and there is a bigger chance of sampling visually similar images, making it thus harder to train and test.

Another trend that can be observed from Table 2 is that the more complex a model is, the higher the accuracy is. Particularly, the highest accuracy is achieved by LSTM followed by GRU and finally RNN. This stems from the fact that models with more complex architectures such as LSTMs are more capable in capturing more complicated patterns in the data, like the underline temporal pattern that exist in the story sequence. In detail, RNN suffers from the vanishing gradient point problem, this problem causes the RNN to have trouble in remembering values of past inputs. On the other end of the spectrum, GRU and LSTM prevent vanishing gradient problem; that and their more complex nature are the main reasons why they outperform regular RNNs.

Another important thing to note is that even though the LSTM achieves an accuracy of roughly 40% on the prediction of the true ending, 99% of the times it learns to ignore the easy, almost irrelevant candidate by choosing one of the other more closely related candidates. Additionally, the fact that 75% of the times the prediction comes down either the true ending or the hard candidate further proves two things; that the model was able to learn the underlying patterns that exist behind the images of a visual story and that with the heuristical sampling produces hard candidates indeed.

In Figure 8 the accuracy per topic is presented. As it can be clearly seen there are several topics in which the model achieves high accuracy and at the same time there are other topics where the model performs relatively poorly. This difference in accuracy is due to the nature of the dataset. Specifically, in those topics where the corresponding stories have low temporal coherence, the model performs the poorest.

On figure 8 some qualitative results are presented from the LSTM on the heuristic method for alternative endings.

On the first row one can see an example of a right prediction by the LSTM model. As it can be clearly seen the system is able to predict the right ending even if the images in the middle of the story have no semantic relationship with the ending image. Moreover, one can argue that the system is able to learn that a story is a coherent sequence of event and that the system can correlate the first image with the last one, i.e. the boat with the sea.

On the second example the system fails to predict the right ending and it assigns the ending to the hard candidate. In more detail it picks the image with the barrels of wine instead of the bottles of wines. This is an extremely hard example in which without any further context given by the textual information it is quite hard to predict, even for a human. But still, it is quite clear that the network has learned to preserve the semantic relationship in the context of the images of the story.

On the third and fourth examples we illustrate two cases where the model fails to predict the right ending and it picks the two moderately hard alternatives. From those two examples, one can argue that the model is able to figure out the close semantic relation, since the images that the system picks as the most probable endings are quite similar to the right ending, i.e. in the first example the model picks the image of a horse instead of a dog, while in the second exam-

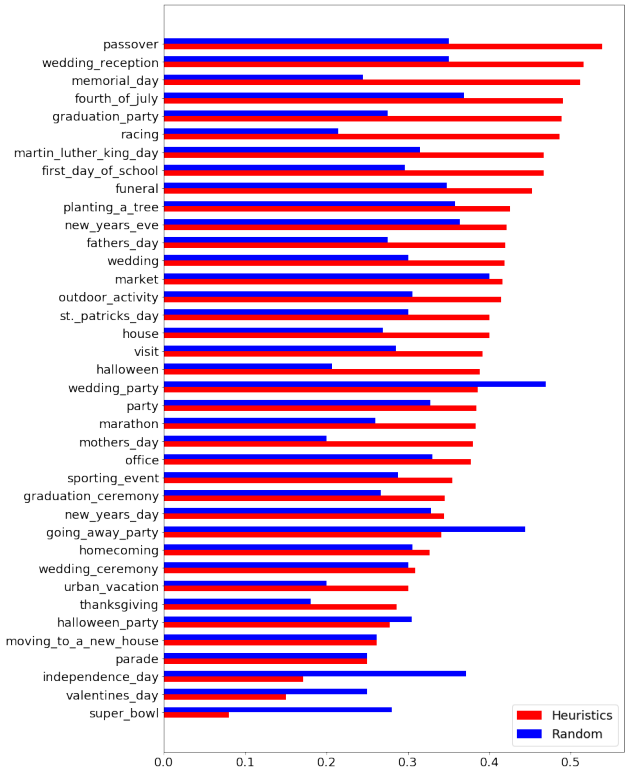


Figure 8. Accuracy per topic (includes only topics with more than 15 stories).

ple the system successfully predicts that the ending should contain humans, but it fails to predict the true ending.

Lastly, in the final example we present a case where the model completely fails to predict the true ending, as it assigns the most irrelevant, easy alternative. As expected, the system seems to perform poorly in stories which do not have a solid structure of a beginning-middle-end or a temporal common sense in general.

7. Conclusion and Future Work

In this work we introduced our approach for the *Visual Story Cloze Test* task. Our approach treats the problem as a binary problem and makes use of visual representations while it is indifferent to textual information. Undoubtedly, incorporating textual information to the current approach would lead to even higher accuracy results; however this was not the goal of this work. Since adding textual information is not our priority, as we progress on this task we could augment the visual representation of the images in order to increase the model's accuracy. In more detail, we notice that many stories take place in different scenes over time, therefore we could benefit by using features extracted from a network that was trained specifically on scene recognition.

Although we have established a qualitative method for sampling the four alternative endings, the overall framework could further be enhanced by having a more unbiased and automatic procedure. Manually choosing the thresholds defined on section 4 is not only a strenuous and time consuming task, but the quality of the results is also varied, with some candidates either being too easy or too hard. Recent work has shown that it is possible to automate this procedure so that the model picks negative samples during train time in such a way that it helps it to learn the true ending easier.

8. Application issues

Concerning the legal issues we might have with the data we are using, as it has been explained in section 4, each image comes under a Creative Commons license. There are 7 types of licenses, ranging from more flexible to more strict. Even thought this doesn't restrict us from using them as training data, we do not have the right to produce copies or display images if the license doesn't allow it. Therefore, if this paper was to be published, we would first need to check if we can show the images presented as examples in this paper, or if we need to refer their original creator.

Concerning the effectiveness of our approach, we did a so called *sanity check* (experiment on a controlled dataset). A sanity check is a test that is done in order to evaluate whether a claim or the result of a calculation can possibly be true. In our case we did it in order to determine that our approach of dealing with the real problem is valid. More information about the sanity check that took place in the current work can be found on section 5.

References

[1] H. Agrawal, A. Chandrasekaran, D. Batra, D. Parikh, and M. Bansal. Sort story: Sorting jumbled images and captions into stories. *CoRR*, abs/1606.07493, 2016.

[2] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[3] D. Frossard. Vgg in tensorflow. `https://www.cs.toronto.edu/~frossard/post/vgg16/`, 2016.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

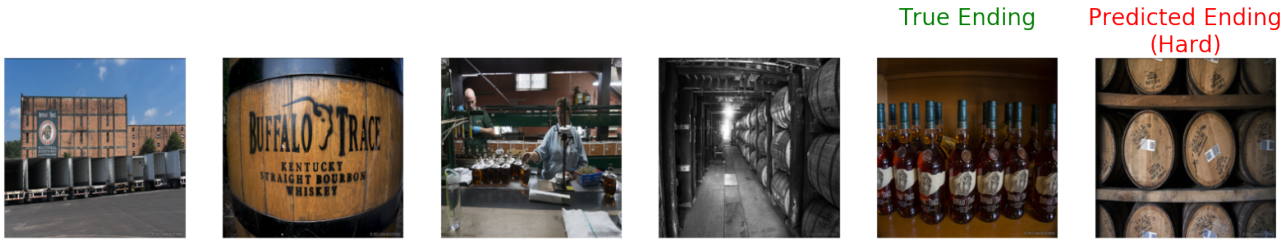
[5] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, C. L. Zitnick, D. Parikh, L. Vanderwende, M. Galley, and M. Mitchell. Visual storytelling. June 2016.

[6] G. Kim and E. P. Xing. Reconstructing storyline graphs for image recommendation from web community photos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

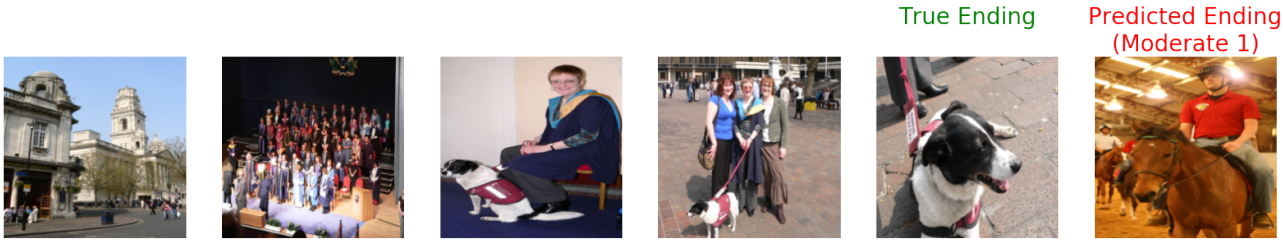
[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

True Ending prediction

Hard Ending misprediction



First Moderate misprediction



Second Moderate misprediction

Easy Ending misprediction

[8] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, abs/1504.08023,

2015.

[9] L. Yu. Visual storytelling api. https://github.com/lichengunc/vist_api, 2017.