CSCI 544: Group 46 Project Status Report
ImagiNarrate: Building a Narrative with Images and Generated Captions

Asmita Chotani , Apurva Gupta , Chetan Chaku , Rutuja Oza and **Priya Nayak**
University of Southern California
[achotani, apurvagu, chaku, rgoza, psnayak] @usc.edu

1 Performed Tasks

1.1 Model Execution

- Study of the research paper (Wang et al., 2018) & AREL model architecture
- Understanding the structure of input and output data points for the AREL model.
- Implemented AREL Model on train and test data points and acquired base accuracies of metrics - bleu, cider, meteor

```
setting up scorers...
computing  score ... Bleu
{'testlen': 44945, 'reflen': 45727, 'guess': [
ratio: 0.9828985063528992
Bleu_1: 0.640
Bleu_2: 0.390
Bleu_3: 0.230
Bleu_4: 0.138
computing  score ... METEOR
METEOR: 0.349
computing  score ... Rouge
ROUGE_L: 0.294
computing  score ... CIDEr
CIDEr: 0.093
Test finished. Time used: 783.6018800735474
```

Figure 1: Test Scores

- Due to large size of data (30GB) and extensive computation required for training and testing the model, we tried various platforms to setup and run the code: Cloudapps VDI (VM by USC), Google Drive + Colab, Google Cloud Platform
- Github repository setup and ensured system environment setup for all group members for end to end model execution.
- Since the AREL model is a bit dated and completely in Python 2, few of the libraries are deprecated and not available for use. For this reason, we changed the entire code base to run the model on Python 3.

1.2 Overview of AREL model training and evaluation process

We begin the model training by providing input to the model in the form of the sequence of image embeddings created using the resnet features, as well as the associated stories. We start training from a simple pre-trained model provided in the repo and train an AREL model (see Figure 2) on top of it.

1.2.1 Input

Inputs stream of 5 ordered images I = (I1, I2, ... , I5) as embeddings from resnet features, Story_line.json- JSON contains the index and text of each story based on the story id Story.h5- the storyline is saved as numpy array.

1.2.2 Model

The AREL model works on the basis of the following:

- Policy model: takes an image sequence to form a narrative story W- finds words from vocabulary to build the narrative
- Reward Model: aims at deriving a human-like reward from both human-annotated stories and sampled predictions.

1.2.3 Output

Outputs word sequence W = (w1, w2, ... , wT), wt in V where V is the vocabulary of all output token. The output is then evaluated on standard metrics and calculates the scores of the model based on BLEU, METEOR, ROUGE and CIDEr.

1.3 Timeline Creation

Divided the project into smaller components (see Figure 3) and generated an estimated timeline for the task completion and alloted task to each member of the group.

2 Risks and challenges addressed

2.1 Storage Issues

For training & testing of AREL model, preprocessed ResNet-152 features are used, which requires over 30 GB of storage space. Downloading and storing this humongous data along with code proved to be a challenge. The problem with using external storage resources was that with CloudApps VDI the data is not persistent across sessions and Google Drive does not have enough free memory.

2.2 Deprecated Python libraries

AREL model was introduced in 2018, owing to which the official model code uses deprecated python libraries from Python 2. These libraries currently are not available for download and hence the code cant be run as is.

2.3 Computational Resources

Testing the architecture was possible on local system, however training the model was computationally intensive and not possible without a powerful system. The training process failed on Cloudapps VDI and Colab.

3 Plan to mitigate risk & address the challenge

3.1 Using Google Cloud Platform

To store the huge resnet and VIST image-story data points as well as to execute the training of the AREL model, we plan to utilized Virtual Machines with customized configuration on Google Cloud Platform using Google Compute Engine API.

3.2 Implementing the model in Python 3

Since the libraries required to execute training & testing of the AREL model are available in Python 3, we essentially re-wrote the entire code for AREL to comply with Python 3 and tested it. We also faced issues in tensorflow data processing, tensorboard visualization and cuda related issues, that were fixed to finally establish baseline scores for story generation.

Considering the challenges identified till now, we have mitigated the issues as mentioned above. There wont be a need for us to change our goals owing to the challenges.

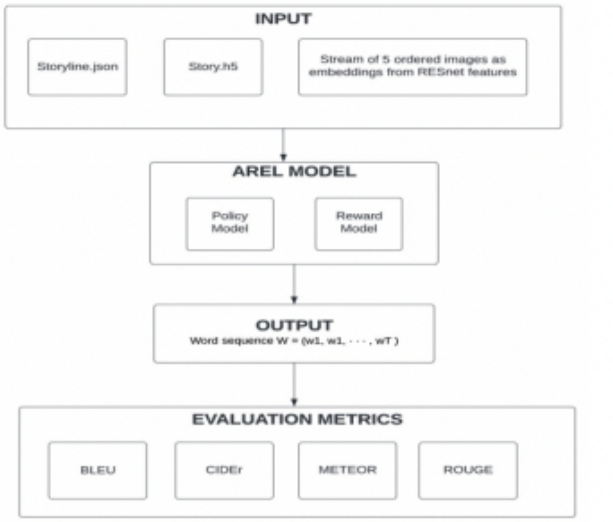

Figure 2: Model Flow

Model: Task	Due Date
AREL Model : Training the model (Image => Story)	Before Status Report
Check the formatting for the inputs required for the model	Before Status Report
Image Caption Model (Image => Caption) Research different models and choose the best model for image captioning	28th March 2023
Model Testing	30th March 2023
Utilizing AREL test data on the image captioning model and evaluate the results	3rd April 2023
Formatting the output to add to the AREL modified model	6th April 2023
Combining Both Models: Combining output from Image Caption Model with the image and feeding it to AREL Model. (Data Processing needed) (Image + Caption => Story)	13th April 2023
Understanding architecture and input-output structure of both the models	15th April 2023
Evaluate the final model and prepare final report	20th April 2023

Figure 3: Project Tasks & Timeline

	Apurva	Asmita	Chetan	Priya	Rutuja
Understand architecture of research paper model and reference research papers	✅	✅	✅	✅	✅
Platform Explored: (CV = CloudApps VDI)	CV	Local	Local	GCP	Colab
AREL Model:					
• Understand the model structure	✅	✅	✅	✅	✅
• Understand inputs and output format			✅		✅
• Data preprocessing	✅		✅		✅
• Training baseline model		✅		✅	
• Tuning, testing and evaluating baselines	✅	✅		✅	

Figure 4: Individual Contributions

3.3 References

References

Project github repository.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. arXiv.