# Big Files You Need to Download:

This is the contents of the .gitignore file for the repository. The highlighted lines were not included in the Git upload because of file size limits, so you will need to redownload them and put them in the proper locations for all the code to work.
.gitignore:

**TER_MovieGraph/Annotations/py3loader/2017-11-02-51-7637_py3.pkl**

- Download from the Toronto MovieGraph dataset

ActionGenome/dataset/ag/frames/*
ActionGenome/dataset/ag/videos/*
ActionGenome/dataset/ag/annotations/*
**STTran_clean/dataloader/object_bbox_and_relationship_filtersmall.pkl**
**STTran_clean/fasterRCNN/models/faster_rcnn_ag.pth**
STTran_clean/data/*
**STTran_clean/pretrained_models/***

- Links to these files, particularly ‘sgdet’ can be found lower in this document under the ‘evaluation’ section

TER_MovieGraph/scene_library/videos/*
TER_MovieGraph/scene_library/frames/*
**TER_MovieGraph/Annotations/py3loader/***

- Download from the Toronto MovieGraph dataset

**STTran_clean/moviegraph/2017-11-02-51-7637_py3.pkl**

- Download from the Toronto MovieGraph dataset

# New Files I Created:

**Core Files:**

TER_STTran_MovieGraph/STTran_clean/Eval_Predictions_with_MGAnnotations.ipynb

This file ties the whole project together. It loads the frames for each scene, predicts the triplets for each scene, loads the ground truth annotations for the scene, and then compares the predictions to the annotations.

TER_STTran_MovieGraph/STTran_clean/moviegraph/MGAnnotations.py

Class to load MovieGraph annotations, extract ‘interactions’ from the location of time nodes, and then process these interactions as triplets in words. I also try and map MovieGraph predicates to predicates found in ActionGenome

TER_STTran_MovieGraph/STTran_clean/lib/evaluation_recall_mg.py

Class used to process MovieGraph annotations created by STTran. It follows a lot of the same logic as the class used to evaluate Action Genome predictions.

TER_STTran_MovieGraph/STTran_clean/dataloader/movie_graph.py

Class to load movie graph annotations, so they can later be easily compared with the scene predictions

TER_STTran_MovieGraph/TER_MovieGraph/scene_extractor_script_time_based.py

A script to create the scene cuts for a movie using the scene graph time windows. Time rather than frame number is used to spit the scenes. The script takes one parameter, the number of frames that should be extracted per second.

**Exploration/Supporting Code:**

TER_STTran_MovieGraph/STTran_clean/STTran_MovieGraph_MakePredictions.ipynb

Environment where I prototyped how to use STTran to generate predictions for movie graph scenes

TER_STTran_MovieGraph/STTran_clean/dataloader/movie_graph_test.ipynb

Environment to test out how to build a MovieGraph class that handles loading individual clips annotations

TER_STTran_MovieGraph/TER_MovieGraph/scene_extraxctor.ipynb

Notebook to test out how one combines the MovieGraph scene cut annotations to construct the bounds for each scene.

TER_STTran_MovieGraph/STTran_clean/moviegraph/Explore%20Annotations.ipynb

Environment to explore how one accesses, parses, and visualizes MovieGraph scene annotations.

TER_STTran_MovieGraph/TER_MovieGraph/scene_extraxctor.ipynb

Environment where I tested how to create the scene boundaries specified for the MovieGraph films. The work here was used to build scene_extractor_script_time_based.py

**Main Repository:** https://github.com/yrcong/STTran

*Some parts copied from readme files*

We use python=3.6, pytorch=1.1 and torchvision=0.3 in our code. First, clone the repository:

```
git clone https://github.com/yrcong/STTran.git
```

We borrow some compiled code for bbox operations.

```
cd lib/draw_rectangles
python setup.py build_ext --inplace
cd ..
cd fpn/box_intersections_cpu
python setup.py build_ext --inplace
```

*For the object detector part, please follow the compilation from https://github.com/jwyang/faster-rcnn.pytorch We provide a pretrained FasterRCNN model for Action Genome.*

Download the model from here (380 MB) and place it at the path below.

```
fasterRCNN/models/faster_rcnn_ag.pth
```

Compile the faster_rcnn model for the local environment.

- Install all python dependencies with pip
  - `pip install -r requirements.txt`
- Compile dependencies using the following commands
  - `cd lib`
  - `python setup.py build develop`

**Data Set**
Download the Action Genome Dataset using this ToolKit

Clone the folder `git clone https://github.com/JingweiJ/ActionGenome`

Download Dataset from here. Select "Data (scaled to 480p, 13 GB)" and extract (or softlink) them under `dataset/ag/videos`

Download Action Genome annotations (~287 MB) tgm and place them under `dataset/ag/annotations.`

After preparing all 480p videos into your `dataset/ag/videos`, dump the frames into `dataset/ag/frames`:

```
python tools/dump_frames.py
```

The dumped frames are ~74GB. The dumping may take half a day to finish. Note that we have only annotated sampled frames (see the sampling strategy in our paper) rather than all frames. If you prefer to dump all frames, run:

```
python tools/dump_frames.py --all_frames
```

After Processing Data WIth the ToolKit the directory structure should look like
he directories of the dataset should look like:

```
|-- action_genome
    |-- annotations   #gt annotations
    |-- frames        #sampled frames
    |-- videos        #original videos
```

In the experiments for SGCLS/SGDET, we only keep bounding boxes with short edges larger than 16 pixels. Please download the file object_bbox_and_relationship_filtersmall.pkl (136 MB) and put it in the `dataloader` folder

## Code Edits

In the file /STTran/dataloader/action_genome.py we must update the package supplying the imread method. Scipy.misc method is discontinued

```
 from scipy.misc import imread
from imageio import imread
```

## Train

You can train the STTran with train.py.

For PredCLS:

```
python train.py -mode predcls -datasize large -data_path $DATAPATH
```

- For SGCLS:

```
python train.py -mode sgcls -datasize large -data_path $DATAPATH
```

- For SGDET:

```
python train.py -mode sgdet -datasize large -data_path $DATAPATH
```

Command for Azeez Local: python test.py -mode sgdet -datasize large -data_path /Users/AzeezA/Google\ Drive/Life/2022/UCA/TER_Project/ActionGenome/dataset/ag/ -model_path sgdet.tar

# Evaluation

We follow three standard tasks from image-based scene graph generation [42] for evaluation : (1) predicate classification (PREDCLS): given ground truth labels and bounding boxes of objects, predict predicate labels of object pairs. (2) scene graph classification (SGCLS): classify the ground truth bounding boxes and predict relationship labels. (3) Scene graph detection (SGDET): detect the objects and predict relationship labels of object pairs. The object detection is regarded as successful if the predicted box overlaps with the ground-truth box at least 0.5 IoU. All tasks are evaluated with the widely used Recall@K metrics (K = [10, 20, 50]) following With Constraint, Semi Constraint and No Constraint. The threshold of confidence in the relationship is set to 0.9 in Semi Constraint for all experiments if no special instruction.

You can evaluate the STTran with test.py.

- For PredCLS (trained Model 902MB):

```
python test.py -m predcls -datasize large -data_path $DATAPATH -model_path
$MODELPATH
```

- For SGCLS (trained Model 426MB): :

```
python test.py -m sgcls -datasize large -data_path $DATAPATH -model_path
$MODELPATH
```

- For SGDET (trained Model 417MB): :

```
python test.py -m sgdet -datasize large -data_path $DATAPATH -model_path
$MODELPATH
```

Azeez local command:

```
 python test.py -mode sgdet -datasize large -data_path
../ActionGenome/dataset/ag/ -model_path pretrained_models/sgdet.tar
```

# MovieGraph Annotations

Hi,

Thanks for your interest in the dataset.

Here are the subtitles (for full movies as well as the clipped scenes):
http://www.cs.toronto.edu/~makarand/downloads/moviegraphs/mg_srt.tar.gz

and the annotations (it's a pickle file, we have a python loader that heavily builds on some crucial classes defined in GraphClasses.py, look at startup.py to get a feel for this)
http://www.cs.toronto.edu/~makarand/downloads/moviegraphs/mg.tar.gz
I tried to make it compatible with py3 (and it works for me), but if it doesn't, please check it with py2.7. You'll also need networkx1.10 since 2.xx changes some important defintions.

The face tracks are here
www.cs.toronto.edu/~makarand/downloads/moviegraphs/mg_ftracks.tar.gz and finally, if you want to replicate by buying DVDs, then some shot and scene information is here
www.cs.toronto.edu/~makarand/downloads/moviegraphs/mg_videoinfo.tar.gz

Finally, some questions asked by previous users are answered here
http://www.cs.toronto.edu/~makarand/downloads/moviegraphs/faq.html Feel free to ask me other ones!

Regards
Makarand