Introduction

For reproducibility, we provide code of our LSCM model based on multi-level visual features. To remain anonymous, we provide only references instead of external links for setup.

Setup

We recommend the following dependencies.

- Python 2.7 (migration to Python 3 is in progress)
- Tensorflow 1.5
- Numpy
- Spacy
- PyDenseCRF

This code derives from RRN [1]. Please refer to its GitHub repo for more details about environment setup and data preparation.

Data Preparation

- **Dataset Preprocessing.** We conduct experiments on 4 benchmarks of referring image segmentation, including `UNC`, `UNC+`, `Gref` and `ReferIt`. After downloading these datasets (please refer to RRN [1]), run the following commands for data preprocessing:

```
python build_batches.py -d Gref -t train
python build_batches.py -d Gref -t val
python build_batches.py -d unc -t train
python build_batches.py -d unc -t val
python build_batches.py -d unc -t testA
python build_batches.py -d unc -t testB
python build_batches.py -d unc+ -t train
python build_batches.py -d unc+ -t val
python build_batches.py -d unc+ -t testA
python build_batches.py -d unc+ -t testB
python build_batches.py -d referit -t trainval
python build_batches.py -d referit -t test
```

- **Embedding Generation.** Please first download GloVe [2] Embedding (`glove.840B.300d.zip`) from its officail website and save it to `data/`. Then run the following commands to generate embeddings:

```
python scripts/embedding_generate.py -d referit
python scripts/embedding_generate.py -d Gref
```

Training and Evaluation

For convenience, we further provide an example bash script to train on UNC train set and evaluate on val set by running the following command:

```
./run.sh
```

- **Training.** Specify several options/flags and then run `python trainval_model.py`
  - `m` : `train`, training mode
  - `d` : Specify dataset, `unc`, `unc+`, `Gref` or `referit`
  - `t` : Split of the dataset, `train` or `trainval`
  - `f` : Path to save the checkpoints
- **Evaluation.** Specify several options/flags and then run `python trainval_model.py`
  - `m` : `test`, evaluation mode
  - `d` : Specify dataset, `unc`, `unc+`, `Gref` or `referit`
  - `t` : Split of the dataset, `val`, `testA`, `testB` or `test`
  - `i` : `700000`, checkpoint of which iteration to be evaluated with
  - `f` : Path where the checkpoints are saved

References

[1] Li, R., Li, K., Kuo, Y.C., Shu, M., Qi, X., Shen, X., Jia, J.: Referring Image Segmentation via Recurrent Refinement Networks. In: CVPR (2018)

[2] Pennington, J., Socher, R., Manning, C.: GloVe: Global Vectors for Word Representation. In: EMNLP (2014)