lancopku / **SGM**

Sequence Generation Model for Multi-label Classification (COLING 2018)

27 commits	1 branch	0 packages	0 releases	1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

ypengc7512 Update README.md — Latest commit 75b2df6 on 5 Jan

models	fix some bugs and update to Pytorch 1.1.0	5 months ago
utils	fix some bugs and update to Pytorch 1.1.0	5 months ago
README.md	Update README.md	5 months ago
config.yaml	fix some bugs and update to Pytorch 1.1.0	5 months ago

lr_scheduler.py	fix some bugs and update to Pytorch 1.1.0	5 months ago
opts.py	Update opts.py	5 months ago
predict.py	fix some bugs and update to Pytorch 1.1.0	5 months ago
preprocess.py	fix some bugs and update to Pytorch 1.1.0	5 months ago
train.py	fix some bugs and update to Pytorch 1.1.0	5 months ago

README.md

Sequence Generation Model for Multi-label Classification

This is the code for our paper *SGM: Sequence Generation Model for Multi-label Classification* [pdf]

Note

In general, this code is more suitable for the following application scenarios:

- **The dataset is relatively large:**
  - The performance of the seq2seq model depends on the size of the dataset.
- **There exist some orders or dependencies between labels:**
  - A reasonable prior order of labels tends to be helpful.

Requirements

- Ubuntu 16.0.4
- Python version >= 3.5
- PyTorch version >= 1.0.0

Dataset

Our used RCV1-V2 dataset can be downloaded from google drive with this link. The structure of the folders on drive is:

```
Google Drive Root                  # The compressed zip file
 |-- data                          # The unprocessed raw data files
 |    |-- train.src
 |    |-- train.tgt
 |    |-- valid.src
 |    |-- valid.tgt
 |    |-- test.src
 |    |-- test.tgt
 |    |-- topic_sorted.json        # The json file of label set for evaluation
 |-- checkpoints                   # The pre-trained model checkpoints
 |    |-- sgm.pt
 |    |-- sgmge.pt
```

We found that the valid-set in the previous version is so small that the model tends to overfit the valid-set, resulting in unstable performance. Therefore, we have expanded the valid-set. In addition, we also filtered out samples that contain more than 500 words in the original RCV1-V2 dataset.

Reproducibility

We provide the pretrained checkpoints of the SGM model and the SGM+GE model on the RCV1-V2 dataset to help you to reproduce our reported experimental results. The detailed reproduction steps are as follows:

- Please download the RCV1-V2 dataset and checkpoints first by clicking on the link, then put them in the same directory as these codes. The correct structure of the folders should be:

```
Root
 |-- data
 |    |-- ...
 |-- checkpoints
 |    |-- ...
 |-- models
 |    |-- ...
 |-- utils
 |    |-- ...
 |-- preprocess.py
 |-- train.py
 |-- ...
```

- Preprocess the downloaded data:

```
python3 preprocess.py -load_data ./data/ -save_data ./data/save_data/ -src_vocab_size 50000
```

All the preprocessed data will be stored in the folder `./data/save_data/`

- Perform prediction and evaluation:

```
python3 predict.py -gpus gpu_id -data ./data/save_data/ -batch_size 64 -restore ./checkpoints/sgm.pt -log resu
```

The predicted labels and evaluation scores will be stored in the folder `results`

Training from scratch

Preprocessing

You can preprocess the dataset with the following command:

```
python3 preprocess.py \
        -load_data load_data_path \       # input file dir for the data
        -save_data save_data_path \       # output file dir for the processed data
        -src_vocab_size 50000             # size of the source vocabulary
```

Note that all data path must end with `/`. Other parameter descriptions can be found in `preprocess.py`

Training

You can perform model training with the following command:

```
python3 train.py -gpus gpu_id -config model_config -log save_path
```

All log files and checkpoints during training will be saved in `save_path`. The detailed parameter descriptions can be found in `train.py`

Testing

You can perform testing with the following command:

```
python3 predict.py -gpus gpu_id -data save_data_path -batch_size batch_size -log log_path
```

The predicted labels and evaluation scores will be stored in the folder `log_path` . The detailed parameter descriptions can be found in `predict.py`

Citation

If you use the above code for your research, please cite the paper:

```
@inproceedings{YangCOLING2018,
  author    = {Pengcheng Yang and
               Xu Sun and
               Wei Li and
               Shuming Ma and
               Wei Wu and
               Houfeng Wang},
  title     = {{SGM:} Sequence Generation Model for Multi-label Classification},
  booktitle = {Proceedings of the 27th International Conference on Computational
               Linguistics, {COLING} 2018, Santa Fe, New Mexico, USA, August 20-26,
               2018},
  pages     = {3915--3926},
  year      = {2018}
}
```