東京大学
THE UNIVERSITY OF TOKYO

PR Reviews

Media Computing in Practice (2022)

M1 Boming YANG

Dept. ICE, IST

2022.7.6

[update] 🐬 build from dockerfile

- Issue:
  1. Make a more convenient method to build and run.
  2. Users do not need to set up the environment.

- Solutions:
- Use Docker to have it run the same way on all environments.
  - Build Docker Image with Dockerfile
    - **Cmake Version** (no apt-get)
    - **nointeractive** (or stunk at time-zone setting)
  - Install Xquartz on MacOS
    - Docker start with DISPLAY
- Ref: https://github.com/mviereck/x11docker
  - https://qiita.com/hoto17296/items/bdb2ab24bc32b6b7f360
  - https://medium.com/@benjamin.botto/opengl-and-cuda-applications-in-docker-af0eece000f1

```
+ FROM ubuntu:20.04
+ ENV PWD=/2022-FunPalettes
+ WORKDIR ${PWD}
+ ENV DEBIAN_FRONTEND noninteractive
+
+ ADD . / ./
+ RUN apt-get update -y \
+     && apt-get install -y wget tar libx11-dev libeigen3-dev libsdl2-dev mesa-utils build-essential libgl-dev
+ RUN wget https://cmake.org/files/v3.20/cmake-3.20.6-linux-x86_64.tar.gz \
+     tar -zxvf cmake-3.20.6-linux-x86_64.tar.gz \
+     && mv cmake-3.20.6-linux-x86_64 cmake-3.20.6 \
+     && ln -sf $PWD/cmake-3.20.6/bin/* /usr/bin
+ RUN mkdir build && cd ./build \
+     && cmake .. -DCMAKE_BUILD_TYPE=Release \
+     && cmake --build . --config Release
+ CMD $PWD/build/FunPalttes
```

```
+
+ ## Build FunPalette Docker (Only MacOS)
+ Install xquartz to receive the GUI
+ ```
+ brew cask install xquartz
+ open -a XQuartz
+ ```
+ Open the Preferences Menu of XQuartz
+ - Turn on **Allow connections from network clients**
+
+ And Open X server on Mac
+ ```
+ defaults read org.xquartz.X11 enable_iglx -bool true
+ xhost +
+ ```
+ ### Docker Part
+ Build Image with dockerfile
+ $IP from ```ipconfig en0```
+
+ ```
+ docker build -t funpalettes:latest .
+ docker run --rm -it -e DISPLAY=$ip:0  -v -v /tmp/.X11-unix:/tmp/.X11-unix --name myfun funpalettes:latest bash
```

[update] create action & update README

- Issue:
  1. Lack of Github action
  2. Code is not highlighted in README

- Solutions:
  1. Make a github action.
  2. Update the code blocks in README.md

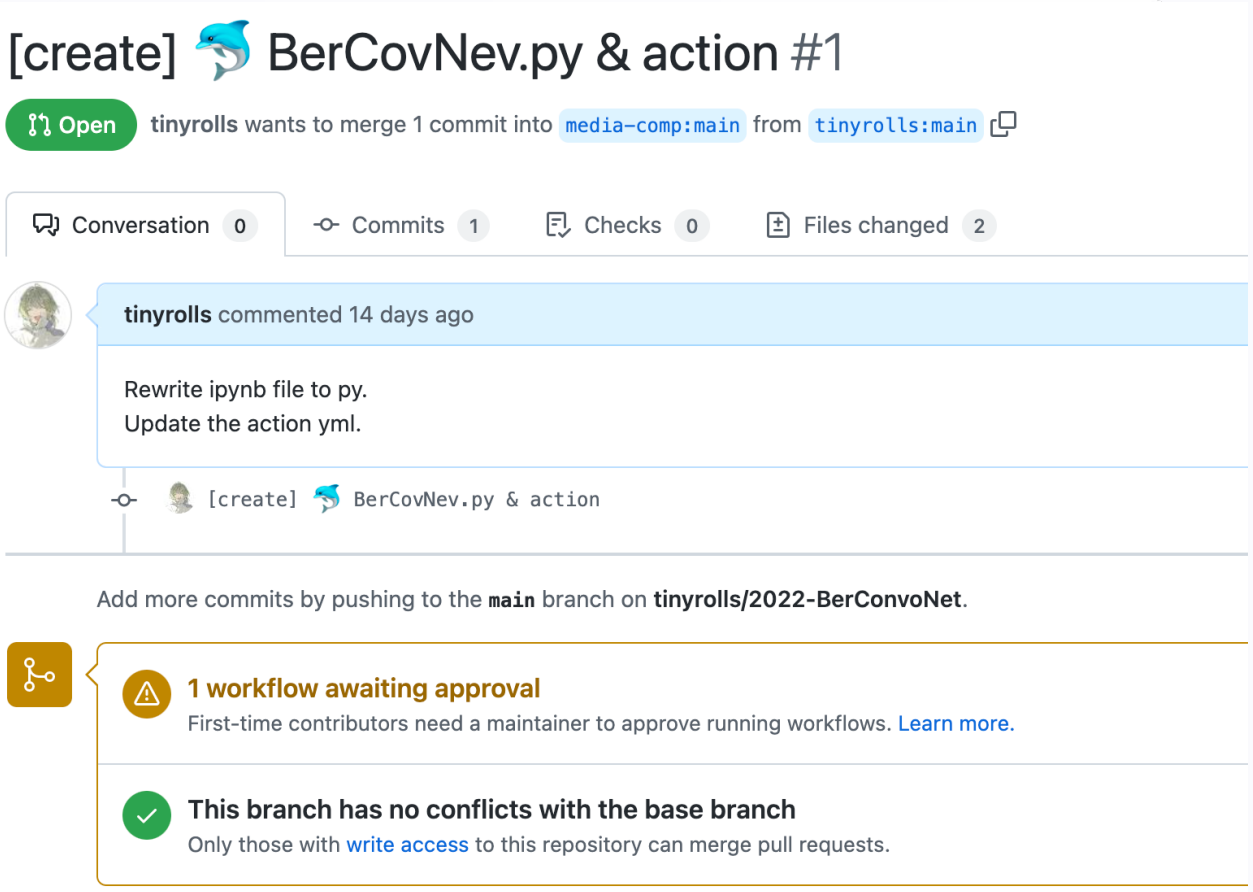
✓ 34 .github/workflows/main.yml

```
@@ -0,0 +1,34 @@
+ name: Test Actions
+
+ on:
+   push:
+     branches: [ main ]
+
+ jobs:
+   build:
+     runs-on: ubuntu-latest
+
+     steps:
+       - uses: actions/checkout@v3
+       - uses: actions/setup-python@v3
+         with:
+           python-version: '3.7'
+
+       - name: Install dependencies
+         run: |
+         pip install opencv-python chardet
+         pip install -r requirements.txt
+
+       - name: Dataset
+         run: |
+           python downloadBasaltDataset.py
+           python dataProcessing.py
+           python compileLabels.py
+
+       - name: Training
+         run: |
+           python train.py
+
+       - name: Test
+         run: |
+           python test.py
```

[update] 🐬BerCovNev.py & action

- Issue:
  1. Lack of Github action
  2. Jupyter could not run conveniently in github action

- Solutions:
  1. Make a single py file from ipynb
  2. Update the github action



2022-KGAT(Received)

Update PaperReview.md #2

Merged **tinyrolls** merged 8 commits into `media-comp:main` from `joshua-shuhan:main` 16 days ago

Conversation 0 | Commits 8 | Checks 1 | Files changed 2

joshua-shuhan commented 28 days ago — Contributor

Hi Boming! I read the paper and updated some contents in `PaperReview.md`. Also, some typos were fixed.

joshua-shuhan and others added 8 commits 28 days ago

- fix typos
- add explanations of embedding layer
- update contents
- update, typo fix
- add block formula
- Revert "add block formula" ...
- Update PaperReview.md
- Update docker-check.yml

```
131 141    # Methodology
132 142    ## Attentive Embedding Propagation Layer  2/2
133 143    3. **Information Aggregation**: The final phase is to aggregate the entity representation $e_h$ and its ego-network rep
           new representation of entity $h$ :
134 144
135     -      $e_{h}^{(1)} = LeakyReLU(W_1(e_h + e_\mathcal{N_h})) + LeakyReLU(W_2(e_h \odot  e_\mathcal{N_h}))$
    145 +      $e_{h}^{(1)} = LeakyReLU(W_1(e_h + e_\mathcal{N_h})) + LeakyReLU(W_2(e_h \odot  e_\mathcal{N_h}))$.
136 146
137 147    4. **High-order Propagation**: Further stack more propagation layers to explore the high-order connectivity information
           from the higher-hop neighbors:
138 148
139     -      $e_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)e_t^{(l-1)}$
    149 +      $e_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)e_t^{(l-1)}$.
140 150
141 151
142 152    ---
         @@ -148,22 +158,23 @@ $e^*_u = e^{(0)}_u \parallel \cdots \parallel e^{(L)}_u, e^*_i = e^{(0)}_i \para
148 158
149 159    2. Finally, conduct inner product of user and item representations, so as to predict their matching score:
150 160
151     -      $\hat{y}(u,i) = e^*_u \top e^*_i$
152     -      $\mathcal{L}_{CF} = \sum_{(u,i,j) \in \mathcal{O}} - \ln \sigma(\hat{y}(u,i) - \hat{y}(u,j))$
    161 +      $\hat{y}(u,i) = (e^*_u)^{\top} e^*_i$
    162 +      $\mathcal{L}_{CF} = \sum_{(u,i,j) \in \mathcal{O}} - \ln \sigma(\hat{y}(u,i) - \hat{y}(u,j))$.
153 163
154 164    ---
```

2022-KGAT(Received)

implemented ablation study #3

Merged **tinyrolls** merged 1 commit into `media-comp:main` from `shuhuh:main` 3 minutes ago

Conversation 0 | Commits 1 | Checks 1 | Files changed 3

shuhuh commented 2 days ago

Hi Boming! I implemented the ablation study on TransR embedding and on attention original paper.

Added two options `ablation_kge` and `ablation_att` to the parser indicating whetl components or not. The default behavior is no ablation, which is the same as the ori

implemented ablation study ···

5 Main.py

```
@@ -40,7 +40,10 @@ def train(args):
40 40
41 41
42 42      # model
43    -   model = KGAT(args, data.n_users, data.n_entities, data.n_relations, None, None)
   43 +   if not args.ablation_kge and not args.ablation_att:
   44 +       model = KGAT(args, data.n_users, data.n_entities, data.n_relations, None, None)
   45 +   else:
   46 +       model = KGAT_ablation(args, data.n_users, data.n_entities, data.n_relations, args.ablation_kge, args.ablation_att)
44 47      model.to(device)
45 48      optimizer = optim.Adam(model.parameters(), lr=args.lr)
46 49
```

5 utility/parser.py

```
@@ -66,6 +66,11 @@ def parser():
66 66      parser.add_argument('--K', type=int, default=20,
67 67                          help='Calculate metric@K when evaluating.')
68 68
   69 +   parser.add_argument('--ablation_kge', type=bool, default=False,
   70 +                       help='Whether to do ablation study on TransR embedding component.')
   71 +   parser.add_argument('--ablation_att', type=bool, default=False,
   72 +                       help='Whether to do ablation study on attention.')
   73 +
69 74      args = parser.parse_args()
70 75
71 76      save_dir = 'result/KGAT/{}/entitydim{}_relationdim{}_{}_{}_lr{}_pretrain{}/'.format(
```

東京大学
THE UNIVERSITY OF TOKYO

Thanks

Media Computing in Practice (2022)