Reasoning Over Knowledge Graph Paths for Recommendation - Documentation

This documentation relates to data preparation and execution of code and work described in the AAAI 2019 paper "Explainable Reasoning over Knowledge Graphs for Recommendation." The paper describes sample use of data. For example, data from three possible sources:

1. https://grouplens.org/datasets/movielens/1m/
2. https://www.imdb.com/
3. https://wsdm-cup-2018.kkbox.events/

Environment Configuration

The following software needs to be configured and pre-installed:

1. Path RNN

torch environment with nn, rnn, optim and cunn installed

2. FMG

python3 environment with numpy, scipy, ctypes and argparse installed

3.ItemPop and NFM

python2/3 environment with sklearn and TensorFlow installed

A. Preparation of Song Data

A.1 : Preparation of Raw Input Data Files

Samples of song data may be prepared for machine learning as below.
See e.g., https://www.kaggle.com/c/kkbox-music-recommendation-challenge for examples of sample song data.

- song_person.dict: key: song id, value: a list of all the related song person id
- person_song.dict: reverse of the song_person.dict
- song_type.dict: key: song id, value: and a list of all the related song type id
- type_song.dict: key: reverse of the song_type.dict

- song_user.dict: key: song id, value: and a list of all the related user id
- user_song.dict: key: reverse of the song_user.dict
- **song_id.txt**: store all the song ids
- **user_id.txt**: store all the user ids
- **user_song_tuple.txt**: all the direct relations between user and song in the Knowledge Graph.

Attention: The three blue files need to have data sorted, and the key use_id should be used to sort user_song_tuple.txt in order to save the memory.

A.2 : Configure the Raw Data Files

The raw data files are then configured by executing the shell scripts below:

path_config.sh: config file

run_path_find.sh: run run_path.sh

A.3: Generation of Training Data

Data used for PathRNN:

negative_matrix.tsv.translated, positive_matrix.tsv.translated:

path_rnn_test_samples_0.0.txt: sampling data used for evaluation

Data used for FMG (FMG data):

fmg_test_samples_0.0.txt: sampling data used for evaluation

user_neg_song.txt: all the negative sampling data

user_pos_song.txt: all the positive sampling data

user_song_train.txt: training data

Attention: All the id mentioned here are the raw id. You need to use PathRnn and Fmg's id-mapping list to transform before using.

A.4. Script to Configure Training Data

bash run_path_find.sh ./path_config.sh

B. Preparation of Movie Data

Samples of movie data may be prepared for machine learning as below.
See e.g., https://grouplens.org/datasets/movielens/1m/ for examples of movie data.

B.1.Input Data

data/input: the three files are generated from training data to generate the path dataset.
data/vocab: embedding dictionary

all_entity_id.txt: mapping from entity to id. we add #UNK_ENTITY and #PAD_TOKEN as padding.
all_relation_id.txt: mapping from relation to id. we add #UNK_RELATION ,#PAD_TOKEN and #END_RELATION as padding
domain-label: mapping from label to id.

entity_to_type.txt: mapping from entity to entity type.

entity_type_id.txt: mapping from entity type to id. we add #UNK_TYPE and #PAD_TOKEN as padding.

B.2. Pre-processing of Movie Data

run bash movie_data_format.sh

B.3. Training of movie data (run_scripts/)

config.sh: config parameters for model training

train.sh: model training script

Running: bash train.sh ./config.sh

B.4. Evaluation of movie data (eval/)

Test dataset: test_samples/test_samples_0.0.txt

Runing: bash eval.sh <model path> <result save path> <topk> model

Running Sample: bash eval.sh ../run_scripts/results/lse/2018-09-01-13-08-29/listen/model- latest ./config8

C. FMG (song_fmg)

C.1.Input

data/song/entity_ids:

data/song/tuples:

Attention: In order not to exceed memory limit, please split user_song_test_fmg.txt into several small files and put those files into test_samples folder.

C.2. Data Preprocessing

Run: bash data.sh

C.3. Training

Run: python3 movie_run_exp.py config/song.config –reg 0.5

C.4. Test

Run: bash run_test.sh you need to modify config/song.config set "test" as 1 before testing. You also need to modify the model path.

D. ItemPop and NFM (songBaseModel)

D.1. Input Data

baseModel_song_id.txt

baseModel_user_id.txt

baseModel_train.txt

baseModel_test.txt

kkbox_kg.csv: (sample code:/notebooks/kkbox_path_new/get_kg_csv.py)

data format: (item id, relation id, positive entity id, negative entity i

D.2. ItemPop Training and Testing (ItemPop)

Run: python ItemPop.py

D.3. NFM Training and Testing (MF)

Run: bash train_nfm.