

문제 상황 – LightGCN 구현 차이

LightGCN_1
<div>+embedding_size +n_layers +embedding +norm_adj +to +config +name +device +n_users +n_items +trainable</div>
<div>+generate_graph(dataset) +get_rep() +bpr_forward(users, pos_items, neg_items) +predict(users)</div>

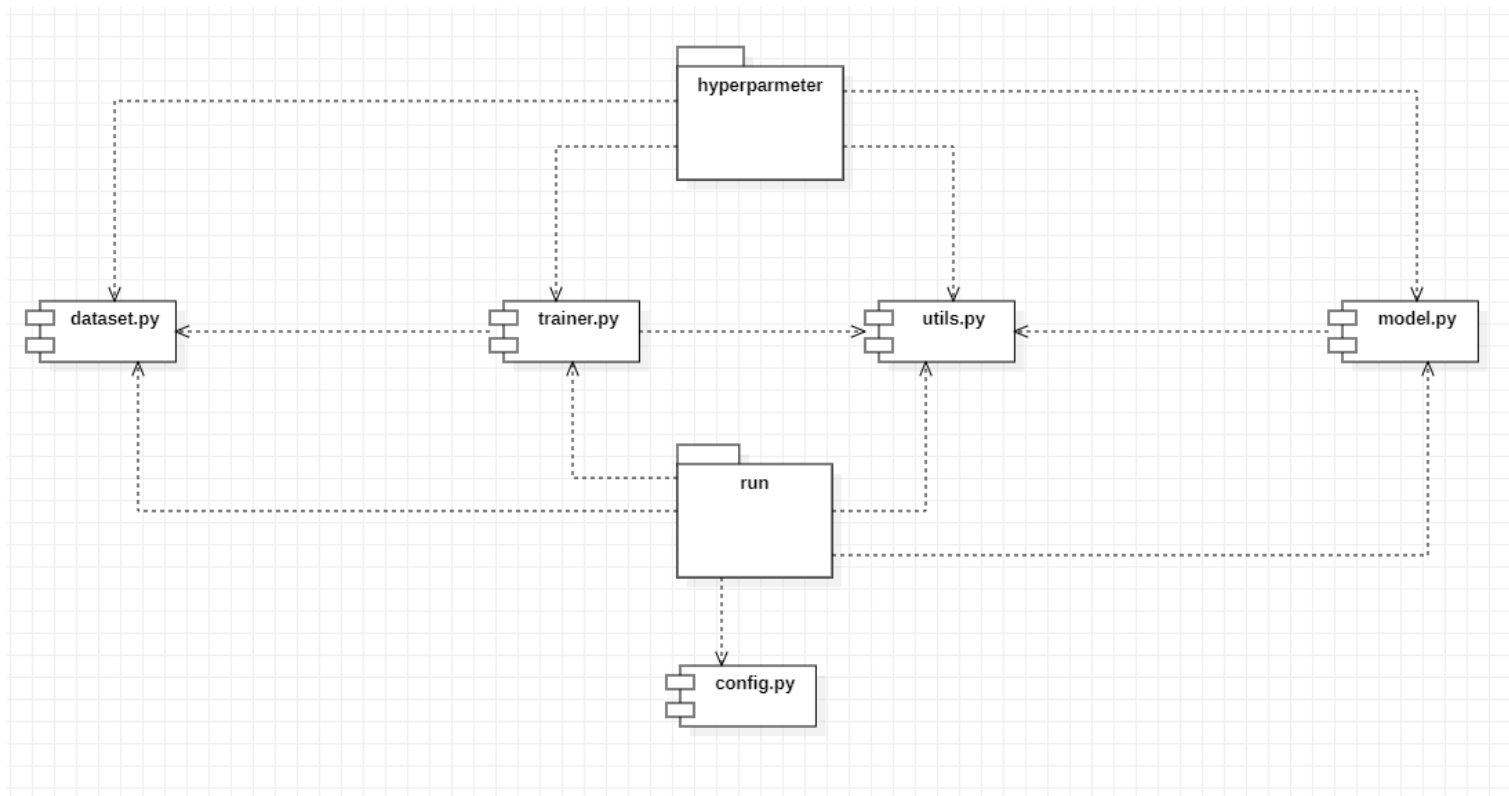
IGCN

LightGCN_2
<div>+config +n_layers +dataset +num_users +num_items +latent_dim +keep_prob +A_split +embedding_user +embedding_item +f +Graph</div>
<div>+__dropout_x(x, keep_prob) +__dropout(keep_prob) +getUsersRating(users) +getEmbedding(users, pos_items, neg_items) +bpr_loss(users, pos, neg) +forward(users, items) +computer()</div>

LightGCN(GTN)

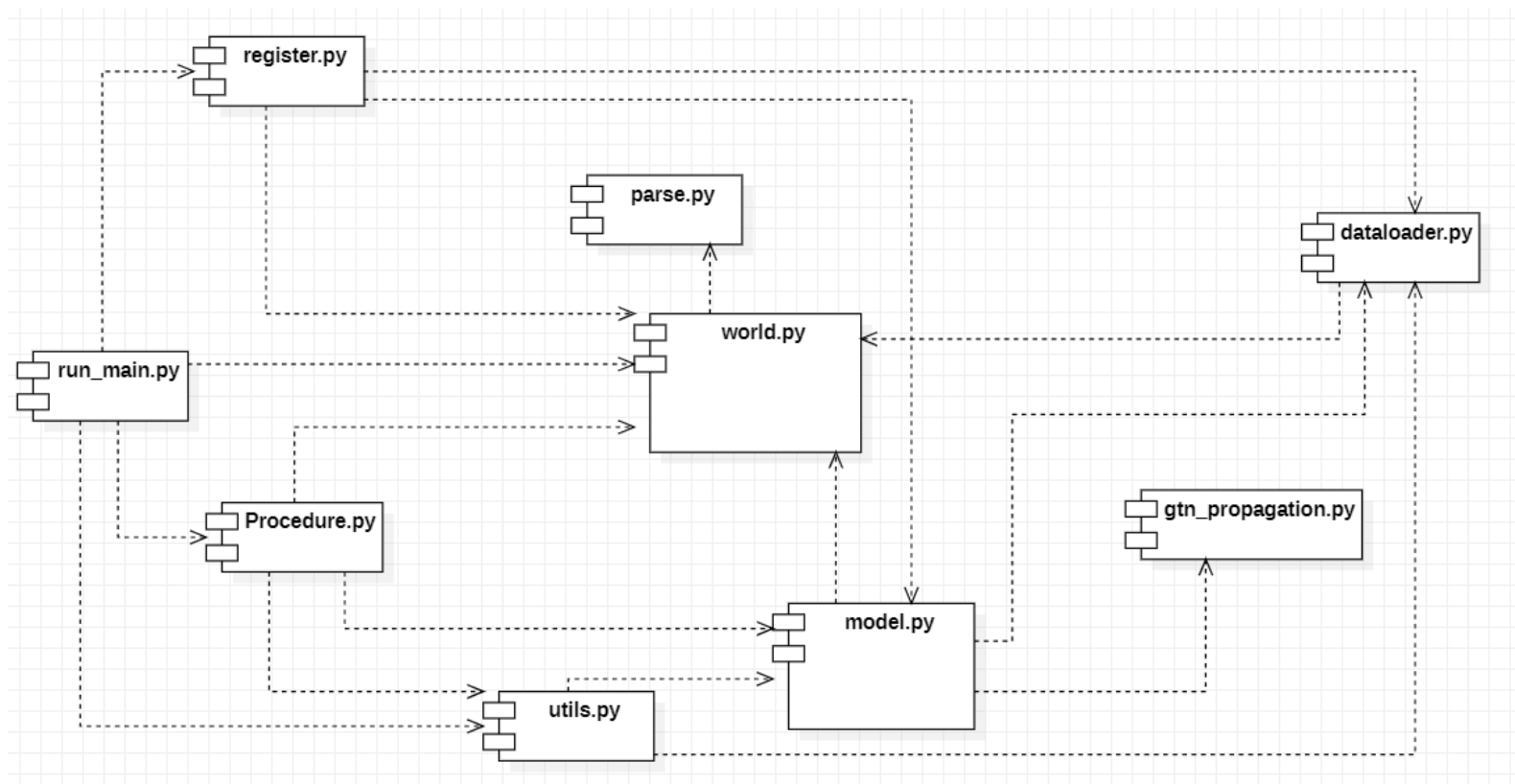
IGCN 아키텍처

- 하이퍼파라미터 패키지에 igcn_tunning.py, lgcn_tunning.py와 같은 train 모듈 존재
- model.py 모듈 내에 lgcn, igcn, 기타 baseline 모델들이 모두 정의돼 있음
 - dgl** 라이브러리를 이용해 간단하게 구현함
- 하이퍼파라미터 패키지 내의 모듈들을 실행해서 checkpoint 폴더에 save해둔 뒤 run 패키지를 통해 transductive, inductive한 시나리오에서 테스트

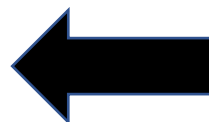
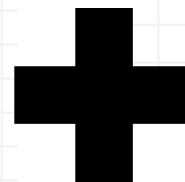
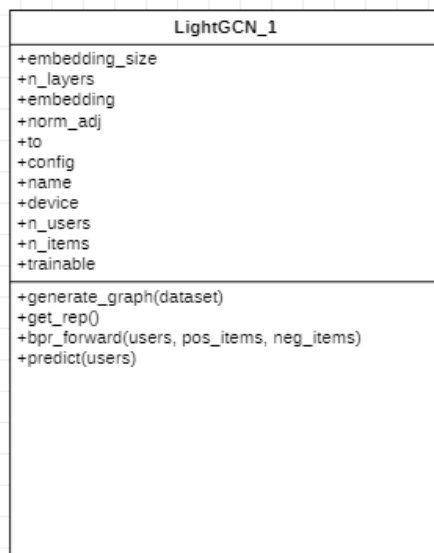


GTN 아키텍처

- SIGIR2020에 기재된 LightGCN 논문의 코드를 기반으로 함
- 기존 코드와의 차이점
 - model.py의 특정 메소드 로직의 일부 수정
 - model.py에서 gtn_propagation.py를 import한 점



과제



IGTN