OpenKE

* Maintains modularity and extensibility to easily incorporate new models into the framework
* Some existing large-scale knowledge graphs pre-trained by OpenKE
* C++ for data preprocessing and negative sampling
* 4 repositories:
  + OpenKE-PyTorch: the project based on PyTorch, which provides the optimized and stable framework for knowledge graph embedding models.
  + [OpenKE-Tensorflow1.0](https://github.com/thunlp/OpenKE/tree/OpenKE-Tensorflow1.0): OpenKE implemented with TensorFlow, also providing the optimized and stable framework for knowledge graph embedding models.
  + [TensorFlow-TransX](https://github.com/thunlp/TensorFlow-TransX): light and simple version of OpenKE based on TensorFlow, including TransE, TransH, TransR and TransD.
  + [Fast-TransX](https://github.com/thunlp/Fast-TransX): efficient lightweight C++ inferences for TransE and its extended models utilizing the framework of OpenKE, including TransH, TransR, TransD, TranSparse and PTransE.
* For those large-scale entity sets, to corrupt all entities with the whole entity set is time-costing. Hence, we also provide the experimental setting named "[type constraint](https://www.dbs.ifi.lmu.de/~krompass/papers/TypeConstrainedRepresentationLearningInKnowledgeGraphs.pdf)" to corrupt entities with some limited entity sets determining by their relations.
* KE models define a scoring function S(h,r,t) for each triple (h,r,t). scoring function returns higher score for true triples.
* KE models formalize margin-based loss as the training objective to learn embeddings
* The main difference between several models is scoring functions
* All specific models are implemented by inheriting the base class with designing their own scoring function and loss function
* FB15K is the relatively dense subgraph of Free- base; WN18 is the subset of WordNet
* TransE based on OpenKE only spends about 18 hours training the whole Wiki- data for 10000 rounds and gets stable embeddings
* Code Description:
  + Openke/ Base
    - Corrupt.h- corrupts the head, tail and relation
    - Random.h- generated random integer,contains rand\_max function
    - Triple.h- contains the triple structure with h,t and r
    - Reader.h- initializes training, test triples, lefhead, righead, lefTail, rigTail
  + Openke/ data