Multi-class link prediction with PyKEEN and Large Language Models

William Liaw Sriraam Appakutti Palani Eddie Groh
Mochamad Ardiansyah Nurgaha

Mehwish ALAM
Associate Professor

Language Models and Structured Data

9 January 2025

Table of Contents

1. Introduction
2. Methodology
3. Experimental Setup and Key Metrics
4. Analysis and Discussion
5. Summary

Introduction

Knowledge Graph Completion

Goal: Predict missing links to complete and enhance knowledge graphs.

Methods: Use embedding models, LLM-based techniques, and hybrid approaches.

Python **K**nowledge **E**mbedding and **E**valuation **N**etwork

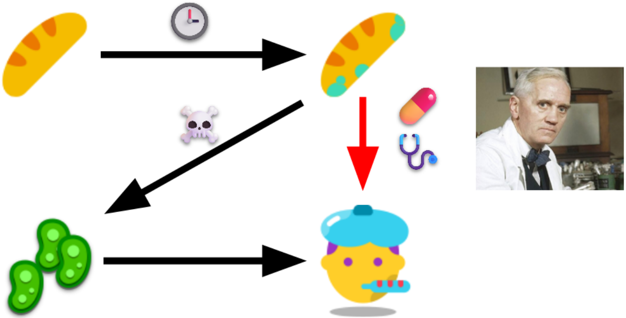
Purpose: Knowledge graph embedding and link prediction tasks.

Neo4j Desktop

Purpose: Neo4j is a graph database designed to store and manage connected data.

Goal & Motivation

How to predict missing links to complete and enhance knowledge graphs with LLMs?



Problem Statement

Motivation: Knowledge graphs are often incomplete, limiting applications like

- Recommendation systems
- Biomedical research
- Drug repurposing

Objective: Address incompleteness by

1. Employing PyKEEN for link prediction and relationship classification.
2. Exploring LLMs to complement traditional methods.

Methodology Overview

1. PyKEEN:
   - Enables KGE models over entities and relations into vector spaces.
   - Facilitates link prediction and multi-class classification.
2. Neo4j:
   - Query and manage the graph data.
   - Retrieve triples (h, r, t) for PyKEEN training.
3. LLM Integration:
   - Dual embedding architecture of RotatE with LLaMA 3.2-3B
   - Future exploration on zero-shot, few-shot, and RAG techniques.

PyKEEN Setup

Extract triples using Cypher query

MATCH (h)-[r] → (t)
RETURN id(h) AS head, type(r) AS relation, id(t) AS tail

Convert triples to PyKEEN's TriplesFactory format

For example: [("Gene_A", "causes", "Disease_X")] → [0, 0, 1]

- These ID mappings are used during model training.

Train using RotatE model with

- 100 epochs
- Embedding dimension = 128
- Family of KGE models tested.

Neo4j and LLMs Integration

Neo4j Integration

- Setup Neo4j Desktop with Hetionet Database.
- Visualize Schema using `CALL db.schema.visualization()`.

LLMs (Dual Embedding Architecture)

Integrating RotatE with LLaMA 3.2-3B for Knowledge Graph Embeddings

Key Components:

- **RotatE:** Traditional entity and relation embeddings in complex space, trained using PyKEEN.
- **LLaMA 3.2-3B (RLM-A):** LLM embeddings, initialized with Wikidata entries for entities.

This semantic-rich embeddings enhances link prediction.

Experimental Setup

Hetionet:

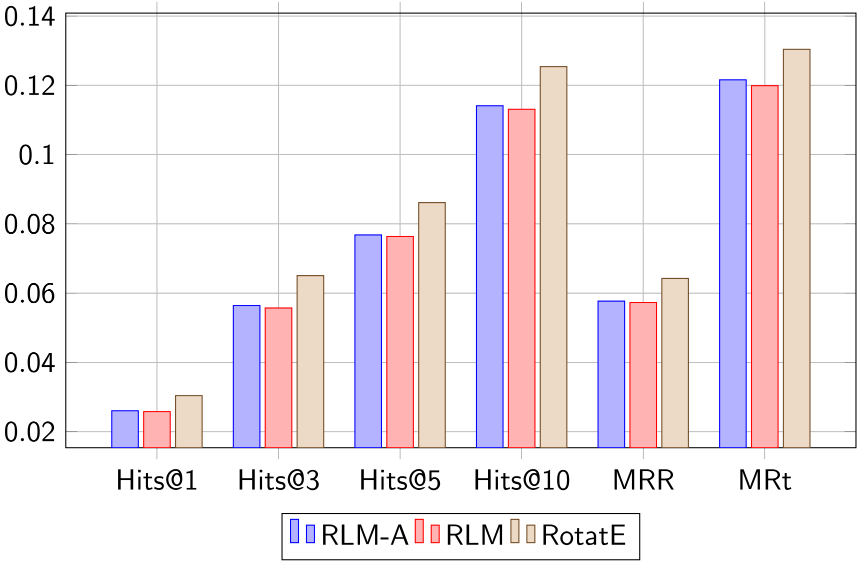
- Biomedical knowledge graph representing relationships between diseases, drugs, genes, and other biological entities.
- High edge-to-node ratio

Statistic	**Value**
Entities (Nodes)	22,634
Relationships (Edges)	561,721
Unique Relation Types	10
Unique Triples	561,721

Evaluation Metrics:

- Hits@K - Fraction of correct predictions ranked in the top K.
- Mean Reciprocal Rank (MRR) - Average inverse rank of correct predictions.
- Mean Rank (MR) - Average rank of correct predictions.

Model Comparison



Analysis and Discussion

Highlights:

- Poor performance across all KGE models tested specifically Translational and Semantic Matching models, except promising result from RotatE.
- Integrating RotatE with LlaMA 3.2-3B did not yield further benefits.

Future Improvements:

- Refine hyperparameters.
- Incorporate negative sampling strategies for better generalization.

Challenges:

- Hetionet's biomedical heterogeneity requires models to handle complex relationships.

Proposed Solutions

Explore other embedding models:

- Rule-based models: e.g., AnyBURL for patterns missed by embeddings.
- BoxE (constraints modeling).
- CNN-based models: ConvE, R-GCN for local feature capture.
- Heterogeneous models: HolE, AutoSF for complex datasets like Hetionet.

Summary

Key Takeaways:

1. PyKEEN demonstrates potential for knowledge graph completion.
2. Llama 3.2:3b integration with RotatE didn't show better results.
3. Hetionet is a complex biomedical datasets, needs more advanced models/approaches.

References:

1. [Ali et al., 2021] PyKEEN: A Python Library for Training and Evaluating Knowledge Graph Embeddings.
2. [Himmelstein et al., 2017] Hetionet: Systematic integration of biomedical knowledge.
3. [Neo4j, 2024] Neo4j Graph Database Documentation.