Designing a Knowledge Graph for fitness and exercise sequence generation

Shyam Krishnan Ondanat Veetil
ondanatv@usc.edu

Cibi Chakravarthy Senthilkumar
csenthil@usc.edu

December 8, 2022

1 Project Goals

In an era where more and more people are moving towards a healthy lifestyle, exercises and yoga constitute a central part of everyday life. Even though there is a plethora of information available online, users usually find it overwhelming to discover information that suits their fitness level, calorie requirement or available equipment. The aim of our project is to solve this problem by developing a knowledge graph that provides the user with appropriate exercises based on their preferences. In particular, this project aims to achieve the following goals.

- Construct a knowledge graph by unifying information on various physical activities like Yoga, Meditation, exercises and workouts
- Suggest similar cardiovascular exercises based on equivalent calorie consumption and intensity level.
- Generate a set of exercises based on available equipment and target body part which can be done in a single session.

2 Technical Challenge

2.1 Challenge #1

While data extraction from online sources (Table 1) and KG construction (Fig. 1) for gym and yoga exercises were straightforward, representing cardiovascular activities was challenging. Suggesting similar cardiovascular exercises requires the representation of calories and intensity ranges. For example what would be the best way to represent the fact that *Swimming spends 5 times more calories than jogging when done for the same amount of time* The solution was to *augment the KG* by splitting the calorie expenditure into ranges and making these the new entities in KG (Fig.2). Exercises were then linked to these entities based on their calorie requirement and intensity ranges.

Using this augmented Knowledge graph, *graph embeddings* were generated for each exercise and they were grouped into clusters. Exercises from the same clusters represented exercises with similar calorie expenditure. we used two metrics to evaluate the clusters - *Average intra-cluster difference between calories/MET and Average inter-cluster difference between calories/MET* and our results are mentioned in Table 3

One problem with clustering is we cannot suggest exercises more than that are in a cluster. In order to do that, we have to make a nearest neighbour search whose time complexity is O(n). Instead, we propose a *Hierarchical querying* method which works as follows. We start from the bottommost node and suggest exercises that are linked to the same node. When more exercises are needed we move up the tree one by one and this only takes a time complexity of O(the depth of the tree - log n).

2.2 Challenge #2

Another challenge that we faced was coming up with workout regimes for a single day based on the chosen equipment and muscle. Given that any exercise could be paired with any other exercise we have to *choose from a million combinations*, which is really huge. Hence, we approached this problem by creating clusters of exercises in such a way that exercises from the same cluster can be performed together in a single workout session.

Suppose we come up with such clusters, we need a way to evaluate them. We identified that the quality of clusters could be gauged using two questions:-

- Do the exercises from a single session belong to a single cluster?
- How often are the target body parts in a single cluster trained together?

To quantify the quality for the first question we used *Mean Pairwise Target Area Frequency(MPTAF)* calculated as follows.

$$\text{MPTAF} = \sum_{c}^{\text{clusters}} \sum_{i}^{\text{target areas } \epsilon \text{ c}} \sum_{j}^{\text{target areas } \epsilon \text{ c}} \frac{\text{number of workout regimes with both i and j}}{\text{number of workout regimes with either i or j}}$$

To answer the second question we developed this metric named *Mean Reciprocal Cluster Cardinality(MRCC)* calculated as follows

$$\text{MRCC} = \sum_{W} \frac{1}{\text{number of clusters that contain the exercises of w}}$$

First, we tried the conventional graph embedding approaches to create the clusters. Then, We improved the performance of our clusters by using an unsupervised GCN which is trained on the following task. Given an exercise, target area and a piece of equipment, predict whether this exercise affects this target area and uses this equipment. This improved our performance to 0.75

We further improved the score to 0.81 by taking a supervised approach where we train the network to learn to predict whether a given pair of workouts can belong in the same workout regime. (Table.2)

3 Conclusion and Learning

Hence, we have successfully constructed a knowledge graph to suggest different exercises based on user preferences. Also, one can notice that it can be expanded into areas of Physiotherapy and Gerontology related exercises. During this process we have learnt the following.

- In-depth knowledge about various stages of KG development
- Neo4j and React.js development stack
- Advanced graph embeddings using GCN
- Importance of evaluating KG quality at each stage

Appendices

Ontology

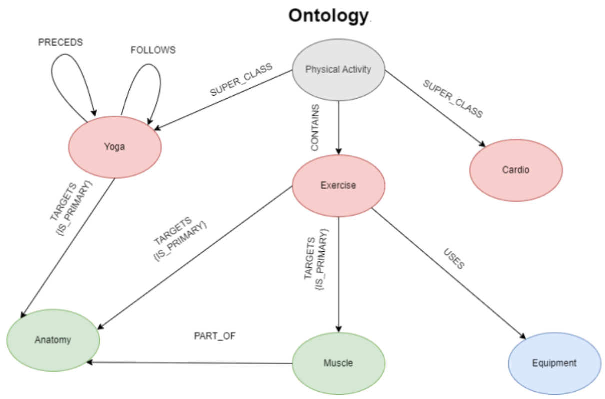

Figure 1: KG Ontology

KG Augmentation

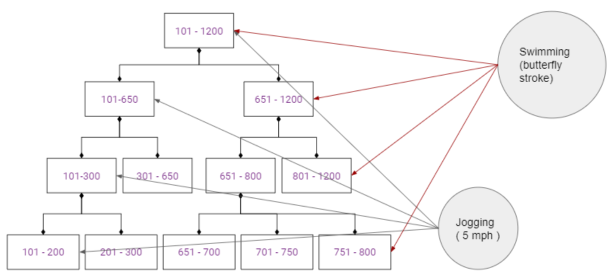

Figure 2: Augmenting knowledge graphs with quantities

Data Sources

Source	Content
Yoga Journal	Yoga and Meditation details
fatsecret.com	Cardiovascular exercises
Muscle&Strength	Workout routines and Gym exercises
exRx	Gym Exercises

Table 1: Data Sources used

Evaluation results for clustering algorithms

Method	Chosen number of Clusters	MRCC Score
TransE	11	0.42
ComplexE	13	0.39
DistMult	15	0.38
HolE	11	0.34
GCN	15	0.75
GCN Supervised	12	0.81

Table 2: MRCC score for various embeddings

Method	Chosen number of Clusters	Average intracluster difference	Average inter cluster difference
TransE	4	114.81/1.56	34.4/0.47
ComplexE	14	50.16/0.68	61.39/0.84
DistMult	14	50.16/0.68	61.39/0.84
HolE	13	57.8/0.79	67.9/0.93
ConvKB	7	76.91/1.05	61.12/0.83

Table 3: Embedding Performance