

Notes taking - Build is ctrl alt b

Navey Mavey

August 27, 2023

# Chapter 1

## Knowledge Graph

### 1.1 Definition

A knowledge graph, also known as a semantic network, represents a network of real-world entities—i.e. objects, events, situations, or concepts—and illustrates their relationship. This information is usually stored in a graph database and visualized as a graph structure, prompting the term knowledge “graph.”

#### 1.1.1 Ontologies

- serves to create a formal representation of the entities in the graph - based on a taxonomy (since there are multiple, ontologies has its own definition) -
- Ontology example: Madison Square Garden - distinguish between events at the location using variables such as time

#### 1.1.2 how it works

- schemas - Provide the framework for the knowledge graph
- identities - classify the underlying nodes appropriately
- context - determines the setting in which that knowledge exists
- help distinguish words with multiple meanings
  - fueled by machine learning, NLP to construct a comprehensive view of nodes, edges and labels through a process called semantic enrichment

### 1.2 Combining Graph Neural Networks and Sentence Encoders for Knowledge-aware Recommendations

- first exploited **graph neural networks** to encode both **collaborative features**, such as the interactions between users and items, and structured properties of the items
- Next, we used a **sentence encoder** that relies on trans-

formers to learn a representation based on textual content describing the items.  
 - Finally, these **embeddings are combined** by **exploiting a deep neural network** where both *self-attention* and *cross-attention* mechanisms are used to learn the relationships between the initial embeddings and to further refine the representation

### 1.3 Interactive Recommender System via Knowledge Graph-enhanced Reinforcement Learning

- <https://arxiv.org/pdf/2006.10389.pdf>
- <https://towardsdatascience.com/introduction-to-reinforcement-learning-markov-decision-process-44c533ebf8da>
- Reinforce learning
  - a process in which an agent learns to make decision through trial and error
  - modelled mathematically as a Markov decision process
  - there is a learner and a decision maker called agent and the surrounding with which it interacts is called environment
  - environment, in return, provides rewards and a new state based on the actions of the agent
- Q learning
  - Q-learning finds an optimal policy in the sense of maximizing the expected value of the total reward over any and all successive steps, starting from the current state.
  - Reinforcement learning in IRS
    - model based techniques - utilize policy iteration to search for the optimal recommendation policy where an action is defined to be an item and a state is represented as n-gram of items
    - model free techniques - more popular lately (including policy gradient based) - pg based learn a stochastic policy as a distribution over the whole item space and sample an item according to such distribution
- KGQR (Knowledge Graph Enhanced Q-Learning Framework for Interactive Recommendation)
  - integrate graph learning and sequential decision making as a whole to facilitate knowledge in KG and pattern mining in IRS

- to alleviate data sparsity, the user feedback is modeled to propagate via structure information of KG so that the user's preference can be transited among correlated items
- This way one interactive record can affect multiple connected items, thus the sample efficiency is improved
- Aggregating the semantic correlations among items in KG, the item embedding and the user's preference are effectively represented -; leads to more accurate Q-value approximation
- 4 main components
  - \* graph convolution module
  - \* state representation module
  - \* candidate selection module
  - \* Q-learning network module
- at each timestep, the IRS sequentially recommends item to user and correspondingly updates its subsequent recommendation strategy based on user's feedback
- IRS models user's preference via graph convolution module and state representation
  - \* interaction history
  - \* knowledge graph
- state representation - Graph convolutional embedding layer - Behavior aggregation layer

## 1.4 Knowledge graph embedding based on semantic hierarchy

- The knowledge graph is mapped to a polar coordinate system, where concentric circles naturally reflect the hierarchy, and entities can be divided into modulus parts and phase parts, and then the modulus part of the polar coordinate system is mapped to the relational vector space through the relational vector, thus the modulus part takes into account the semantic information of the knowledge graph, and the phase part takes into account the hierarchical information.
- 

## 1.5 Building Knowledge Graphs and Recommender Systems for suggesting Reskilling and Upskilling Options from the Web

- the adaptation of knowledge extraction methods to the human resources and continuing education domain

- developing a knowledge-driven recommender system that draws upon this background knowledge to support users in identifying useful reskilling and upskilling options
- slot filling - combines multiple pieces of information (eg. skills, learning outcomes) into a single knowledge base entries
- When applied to open-world scenarios, slot filling is very challenging, as demonstrated by competitions such as the TAC 2017 Cold Start Slot Filling Task in which even the winning systems only obtained F-measures below 20 %

## Chapter 2

# Natural Language Processing

- Rule based modelling of human language

### 2.1 Tasks

- Speech Recognising - Part of speech tagging (whether its a noun or a verb etc. as some words can be both - word sense disambiguation (semantic analysis of word) - Named entity recognition - eg. Fred is a name, Germany is a location, etc - Co-reference resolution - 2 words refer to the same entity. eg. Lea = she but could also be a metaphor or idiom - Sentiments analysis: attempt to extract subjective quality (sarcasm, confusion) - Natural language generation

### 2.2 BERT

- train deep bidirectional representation of unlabeled text by joint conditioning on both left and right context in all layers

## Chapter 3

# Learning as a Network (LaaN)

builds upon connectivism, complexity theory, and double-loop learning. It promotes a theory of change, movement, dynamism, self-organization, emergence, and effectiveness, which puts the learner at the center and represents a knowledge ecological approach to learning.

# Chapter 4

## GCN

### 4.1 Neural Graph Collaborative Filtering

- 2 components in learnable CF model
  - embedding - transforms users and items to vectorized representations
  - interaction modelling - reconstructs historical interactions based on the embedding

### 4.2 LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation

- 2 most common designs in GCNs, feature transformation and nonlinear activation, contributes little to the performance of collaborative filtering, instead they add difficulty
- simplify the design of GCN to make it more concise and appropriate for recommendation
- Basic idea: Collaborative filtering (matrix factorization, SVD++, Neural attentive item similarity (NAIS))
- NGCF (Neural graph collaborative filtering) - deepens the use of subgraph structure with huge-hop neighbours.
  - takes inspiration from GCN
  - follows propagation rule to refine embeddings
  - heavy and burdensome - not useful for CF
  - semi supervised node classification
    - \* each node has rich semantic features as input, such as the title and abstract words of a paper



- it works by propagating L layers, nGCF obtains L+1 embedding to describe user and an item
- then it concatenates these L+1 embeddings to obtain the final user embedding and item embedding, using inner product to generate the prediction score
- $\sigma$  (not linear activation function) and feature transformation matrix W1 and W2 are not as useful in collaborative filtering
- it might be useful in semi-supervised learning for features learning (eg. of title and abstract words of a paper)
- in collaborative filtering each node of user item interaction graph only has an ID
- LightGCN includes the most essential GCN component - Neighbourhood aggregation - for collaborative filtering
- BASIC IDEA : learn representation for nodes by something features over the graph

$$e_u^{k+1} = AGG(e_u^{(k)}, \{e_i^{(k)} : i \in \mathcal{N}_u\})$$

- AGG : aggregation function - considers the kth layer's representation
- we aggregate only the connected neighbours and not the target node itself (self connection) since the layer combination operation essentially does the same thing
- Layer combination and model prediction -  $\alpha_k$  is importance of the kth layer embedding -  $\lambda$  treated as hyperparameter
- Performance comparison with NGCF
  - in all cases Light GCN outperforms NGCF by a large margin.
  - increasing the number of layers can improve the performance but the benefit diminish -  $\lambda$  using 3 layers leads to satisfactory performance in most cases
  - LightGCN fits training data better than NGCF (lower training loss)
  - even comparing with state of the art (see table) its still better
  - even using 4 layers, the performance does not degrade

### 4.3 Interest-aware Message-Passing GCN for Recommendation

- to tackle over-smoothing problem in LightGCN in the domain of recommendation

- existing GCN-based recommendation models have not distinguished the high-order neighbors, and just simply aggregate the messages from all those neighbors to update user embeddings.
- As a result, the embeddings of dissimilar users are also involved in the embedding learning of a target user, negatively affecting the performance.
- result graph: LightGCN gets worse as the number of layers increases whereas IMP-GCN does not