Movie Domain specific knowledge graph embedding for recommendation model

1. Background

Demand for movie content recommendation is increased

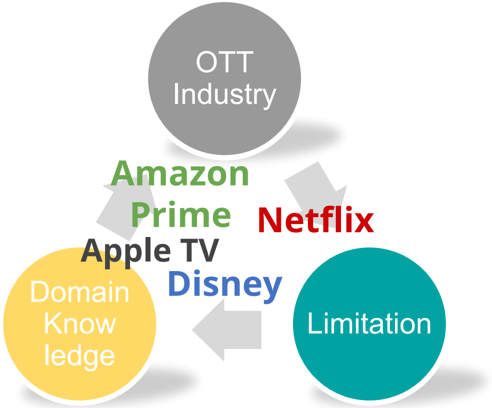
- Increase OTT service and diverse platform that user can choose content
- User can actively choose the diverse content across the country, period, genre
- Recommend right content become one of the most important competitiveness for OTT provider

Limitation of current approach

- Most of existing model usually learn from user historical behaviors on the service.
- Cold-start issue
  - Not enough user behavior during short session
  - No learning about previous user behavior history
- Cannot understand user hidden purpose and interest

Using additional information to enrich data

- Not only depending on user- content interaction relationship
- Using domain information to enrich the data
- Understand the hidden user intention for choosing content



2. Problem

Limitation of current approach

- Depend on the only interaction between user and content
- Not enough Data to understand the user real intention
- Matrix completion is extremely sparse

Knowledge Graph

Lack of domain knowledge

- Using Open source KG that cover general information
- The long-tail distribution of entities results

Not suitable KG Network

- Not enough information
- Too much detail information

Knowledge graph attribute

- Need to be screened Knowledge graph attribute for relation and entity
- Noisy and contain topic-irrelevant connections

3. Description and Goal of Project

Research Goal

- Create the knowledge graph by considering movie domain specific knowledge
- Help recommendation model by supporting additional information to capture user underlying intent and interest

Contribution

Knowledge Graph for Recommendation Model

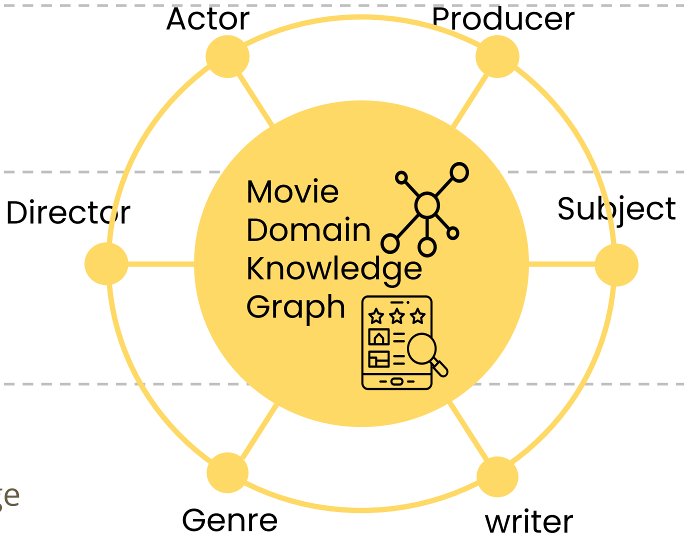
- Novel problem to capture user intents by leveraging knowledge graph

Knowledge Graph Attribute

- Novel knowledge graph attribute based movielens data analysis understanding domain specific knowledge

Structure of Knowledge Graph

- Novel formation of knowledge graph combining two knowledge graph to emphasis important domain attribute



Content

2. Understanding KG

- Knowledge graph concept and goal
- Movielens data analysis
- Knowledge graph attribute
  - Entity
  - Relation

1. Knowledge graph concept and Goal

KG concept

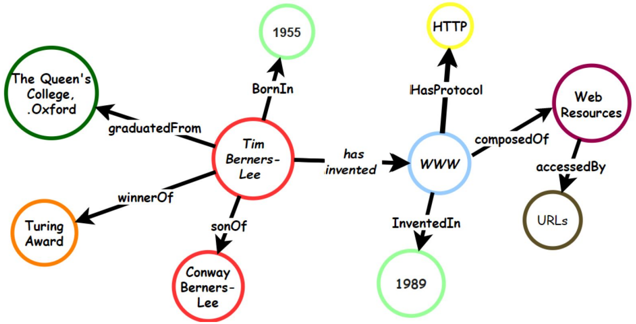
- It represent the data using graph with form of entity and relation.
- The vertices represent the entities and the edges represent the relationships between these entities.
- It consisted of predefined set of entities and each fact in the form of an RDF triple (subject, relation, object)as<h,rr,tt>.

KG Advantage

- KG can depict semantically-interrelated relations that could be domain specific ontologies in a machine-readable format
- KG provide network between entities with defined relationship and this makes it possible to see how everything is related at a big picture level.

KG Embedding

- Representing knowledge by learning vector representations of nodes and edges of labeled, directed multigraphs.
- Predict missing facts involved existing entity and relation.
- Enables the real-world application to consume information to improve performance.



<Example of Knowledge Graph>

2. Movielens data analysis

Description

- GroupLens Research has collected and made available rating data sets from the MovieLens web site
- It contain the information that user select the movie with rating

Dataset

- Using Movielens 100K, 1m , 2m, 25m
- Combine IMDB information with movielens data
- User-Movie-Rating - Movie information (writer, director, actor, genre, producer, country, year..)

Reason

- Background: To choose right entity and relationship
- Goal of analysis : understand which attribute is important for movie domain

Analysis

- Analysis user viewing history
  - Which feature are the most highly affect the user viewing history
  - Calculate the probability that user will re- choose in same category
  - Higher probability is interpreted that it could affect the user decision making => Important attribute

    Formula : $\frac{\text{Number of 'Drama genre movies'}}{\text{Total number of movie}}$

- Correlation between features
  - Understanding the interaction/relationship between the entities
  - Genre- Actor, Movie- Actor, Director- Genre..

2. Movielens data analysis : Analysis user viewing history

Dataset	1st Entity	2nd	3rd
Movielens 100K	Genre 23%	Director	actress
Movielens 1m	Genre 23%	Producer 15%	Director
Movielens 10m	Director 58%	Actress 47%	Producer 42%
Movielens 25m	Director 55%	Actress 47%	Genre 35%

- There are clear pattern in user history
- Depend on the Movielens dataset, the important attribute are slightly different
- The ratio affect the number of attribute category and variance
- The most important entities have highly relevance to movie history list

Director, Actress, Genre, Producer are the key attribute to affect the user viewing history

2. Movielens data analysis : Correlation between features

- **Chi-square test**
  - Probability of H0 being True
  - Higher Chi-square refer to high relevant between the features

<Genre with other>

Genre_multiple	
movieId	72,219,000
actor	69,175,468
writers	65,186,408
actress	46,203,882
producer	38,964,062
director	31,048,176
tag	15,909,626

<Director with other>

Director_multiple	
Movie ID	288,305,850
writers	266,476,187
actor	264,019,340
actress	194,157,302
producer	179,429,289
tag	61,813,090
genres_ml	31,048,176

<Movie with other>

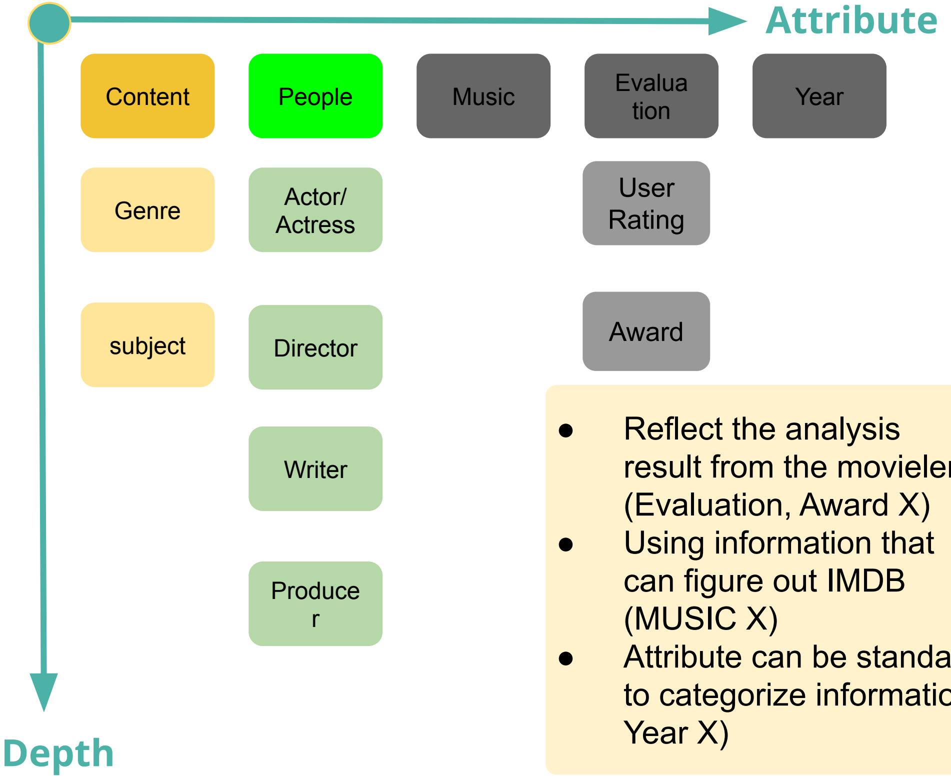
Movie ID_multiple	
actor	671,826,750
writers	618,042,600
actress	470,088,675
producer	397,204,500
director	288,305,850
tag	138,164,574
genres_ml	72,219,000

- Between the variable, All case shows the lower than 0.05 p-value
  => It can conclude that the features are related to each other
  - Genre is highly related to movie, actor, writer in order
  - Director is highly related to Movie writer, actor in order
  - Movie is highly related to actor, writer, actress

Movie, Actor(Actress), Genre, Director, Writer shows the high relevant

3.Knowledge graph attribute

- Based on the Movielens data analysis, select the entity attribute, relationship



- Reflect the analysis result from the movielen (Evaluation, Award X)
- Using information that can figure out IMDB (MUSIC X)
- Attribute can be standard to categorize information(Year X)

Entity	Relation
Director	Is directed by, is edited by
Actor/ Actress	Is filmed by, Is character in
writer	is written by
producer	Is produced by
Genre	Is categorized subclass of, opposite of, based
Subject	Is series of Is topic of

Content

3. Create the Knowledge Graph

- Embedding method
- People Base Knowledge Graph
- Content Base Knowledge Graph

1. Create Knowledge graph

Step1. Existing KG

IMDB30 KG

KB4Rec KG

- Base model for movie domain specific knowledge graph

Step2. Domain Knowledge

IMDB30 KG

KB4Rec KG

Analysis Movielens

- Understand movie domain knowledge
- Analysis result describe the important feature affecting user preference

Step3. Construct KG

IMDB30 KG

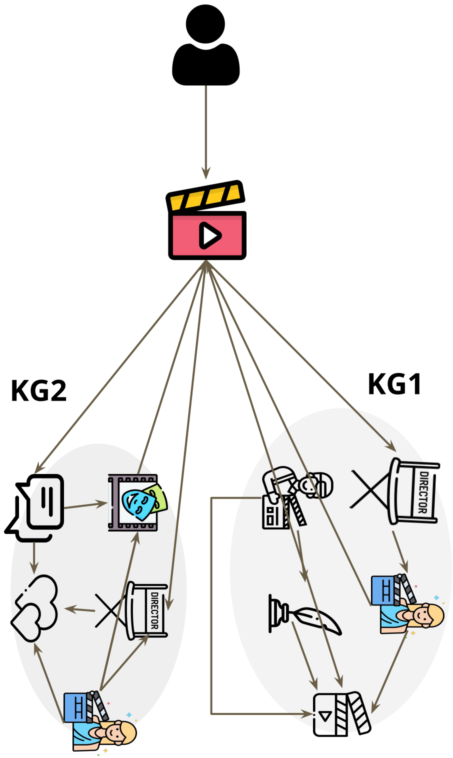
KB4Rec KG

Analysis Movielens

Reconstruct Network KG

- Reflect the analysis result in Knowledge graph Network to focus on the important features

[Construct KG Figure]



2. Knowledge graph concept and Goal - People Based KG

IMDB30 Model

- **General -** IMDB30 based on the public relational data released on IMDb website. The complete IMDb30 contains more than 6 million triplets, and IMDB30 subset contain 115080 triplets formed by 31343 entities and 30 relations.
- **Feature**
  - Head is fixed entity : movies
  - Tail entities are "persons" or "production companies"
- **Entity & Relation**
  - Entity (3) Movie, Actor, Production Company
  - Relation (30)

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Train	Valid	Test
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134
FB15k	14951	1345	483142	50000	59071
FB15k-237	14541	237	272115	17535	20466
IMDb30 subset	**31343**	**30**	**91909**	**11585**	**11586**

Construct Model

- Entity & Relation
  - Using All entity(3) & Relation (30)
  - Select the entity & relation(15) based on movielens analysis
- Dimension : 50 Dimension, 100 Dimension

Selected Attribute
distributors
writer
cast
producers
directors
74.5%

3. Knowledge graph concept and Goal - Content Based KG

KB4REC Model

- **General -** The Knowledge graph based on KB4rec paper and it is subgraph from freebase knowledge graph with linkage of movielense 20m data id and freebase dump
- **Feature**
  - Make linkage between movielens20 ID and freebase dump
  - Set see entity as only contain entities in the linkage
  - Called 1 step subgraph
  - Update the seed entity set to all entities in 1 step subgraph
- **Entity & Relation**
  - Entity (28): actor, casting director, character, cinematography, Collection, costume designer, set decorator…
  - Relation (47)

Datasets	#Items	#Linked-Items	#Users	#Interactions
Movie	27,279	25,982	138,493	20,000,263

Construct Model

- Entity & Relation
  - Using All entity & Relation (30)
  - Select the entity & relation(6) based on movielens analysis
- Dimension : 50 Dimension, 100 Dimension

Selected Attribute
subject
Genre
Character
directors
editor
Producer
actor
Writer
distributor
Network : 19.1%

3. Knowledge graph Embedding - TransE

Challenge of KGE Embedding

1) Nodes in knowledge graphs are entities with different types and attributes

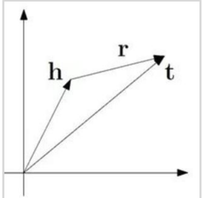
2) Edges in knowledge graphs are relations of different types

TransE Embedding

- It is a representative translational **distance model** that represents entities and relations as vectors in the same semantic space. It can capture relation information in embedding space
- Given a training set S of triplets (h, r, t) the model train vector embeddings of the entities and the relationships.
- Target to close to H + r = t
- Formula : $f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$
- In this project, there are not many relationship

  => TransE can capture appropriate meaning
- Limitation: Hard to consider diverse attribute from

  different relationship => TransR, TranH will applied



[TransE Embedding figure]

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

$\quad \boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$

$\quad \mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

$\quad \mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

$\quad S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

$\quad T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

$\quad$ **for** $(h, \ell, t) \in S_{batch}$ **do**

$\quad\quad (h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

$\quad\quad T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

$\quad$ **end for**

$\quad$ Update embeddings w.r.t. $\displaystyle\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla\left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$

end loop

[TransE Embedding step]

Content

4. Evaluation model

- KG Evaluation factors
- Intrinsic word Embedding
- Finding

1. KG Evaluation

Evaluation Aspect

- The Goal of the model : help to improve the recommendation accuracy.
  - **Aspect 1**: How **node and edge represent** the domain knowledge (word embedding)
    => Intrinsic Evaluation
  - Aspect 2: How network well capture the domain knowledge which is needed for Recommendation model
    => Extrinsic Evaluation

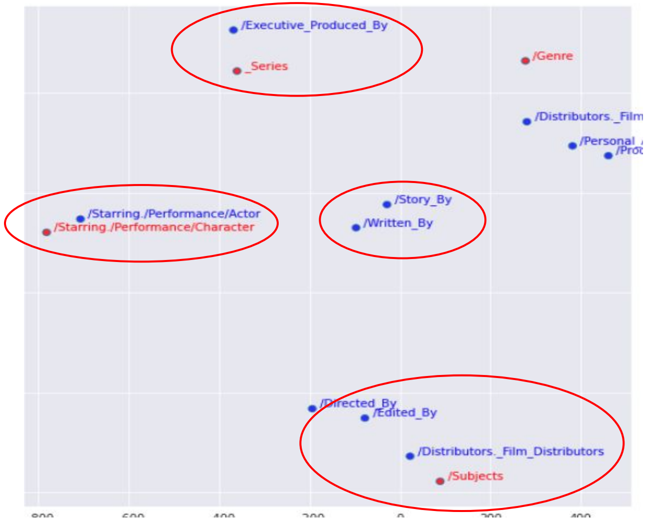
Evaluation Method

- Intrinsic Evaluation by plotting word embedding in 2D and 3-Dimension
- Evaluation
  - Model
    - Content based Knowledge graph
    - People based Knowledge graph
  - Dimension
    - 50 Dimension , 100 Dimension
  - Network
    - Full network and customize network
  - Dimension Reduction
    - PCA, T-SNE

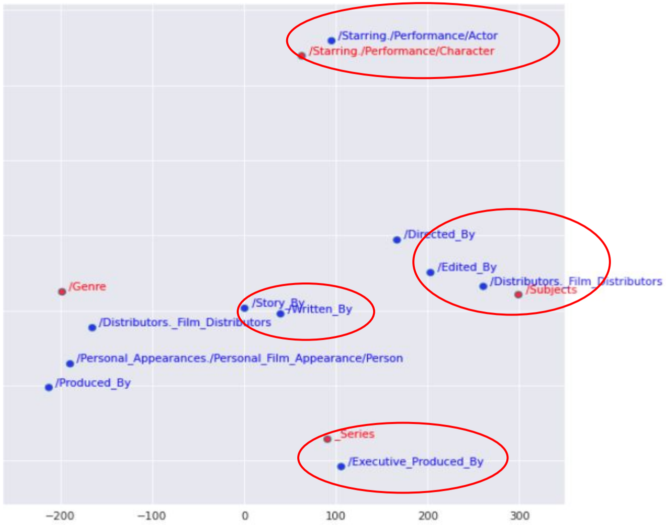
2. Intrinsic word Embedding - Content based Knowledge Graph

- Relation Embedding

  - **Fully connected Network 50- Dimension**



  - **Selected Network 100- Dimension**

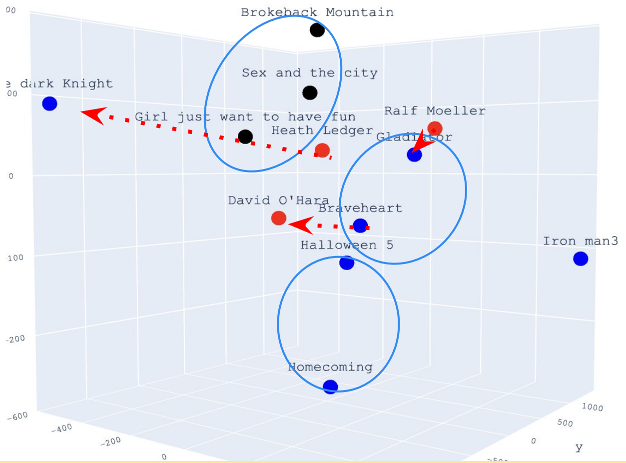


- Relation shows the quite clear result regardless of word Dimension and Network
- Relation has limited number of attribute and more opportunity to be training
- 3D plot is more hard to capture the instinct relationship between relation attribute

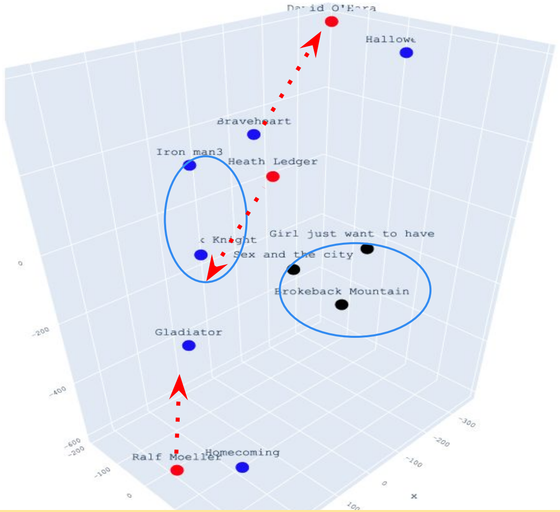
2. Intrinsic word Embedding-Content based Knowledge Graph

- Selected network Entity

[50- Dimension]



[100- Dimension]



Drama Category movie

Action and horror movie

Actor, director

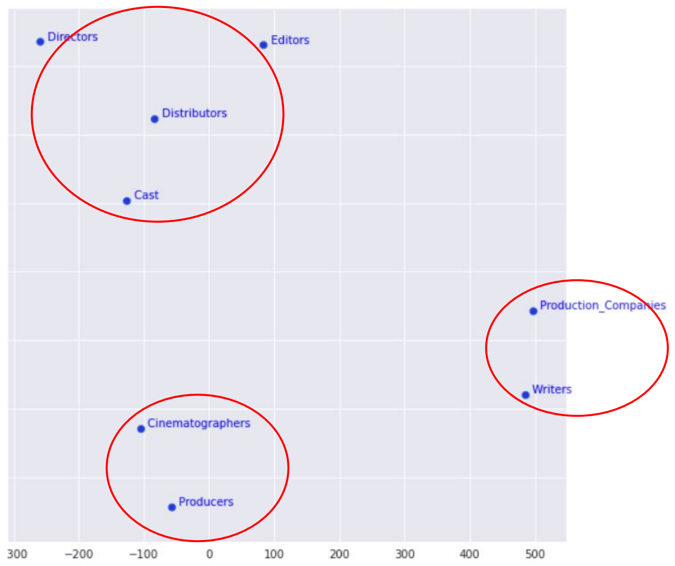
Content Cluster

Movie - People (actor, director) relationship

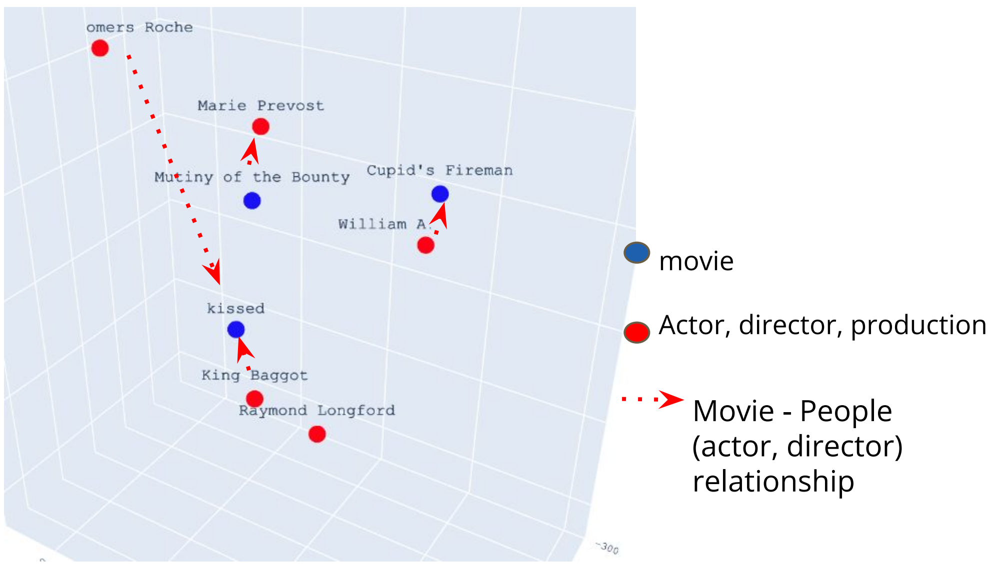
- In entity case, selected network show the better result with 3D
- **Good to capture the Genre or Similar content** (relationship between movie) such as genre, character (Not good at Series)
- But **High variance between Movie - people attribute**
  such as Movie- actor and Movie - director(worse), Movie - Distribution (Worse), Actor - Distribution (the worst)

2. Intrinsic word Embedding - People based Knowledge Graph

[Full Network Relation :100- Dimension]



[Selected Network entity :100- Dimension]



- Relation Result show similar result with Content based Knowledge Graph
- Entity Embedding result show that Actor & Director - Movie relation closely located to each other
- It proves that it contain the movie - people attribute relation semantic context

Conclusion

Finding

- Selected network based on the movielens data analysis shows the better result in embedding evaluation
- Each KGs show the **different strong point** to capture the domain knowledge attribute
    a. People attribute based KG : People - Movie
       (Actor, Director, Writer, Director, Producer, Editor)
    b. Content attribute based KG : Content - Movie
       (Genre, Subject, Series, Character)

Meaning

- Selected attribute based movielens data analysis understanding domain specific knowledge and correlation
- Using two different knowledge graph to capture different aspect of Domain knowledge

Need to improvement

- Shortage of GPU and limited training epoch (epoch 30)