DSCI-558 HW3 Report

Task 1

Task 1.1:

- `movie title / name`, I use 'Metric Longest Common Subsequence' similarity to calculate the similiaty between IMDB and AFI entries's title field. I use the lower case for comparison.
    - I used a library for this: python-string-similarity
- `release date / year`: Because only the year information is available in IMDB, I extract the year from AFI, then compare the year between two. If matched, the score is 1. Considered that certain movies may released on different country in different year, if difference is equal or samller than 1, the score is 0.5
- `genre`: I hand written a mapping table between the genres on the IMDB and the genrse on the AFI. If there is a match in the list of two parties, the score is 1.

Task 1.2:

- The title / name is the most important information in this ER task, it takes 60% of the weighted score.
    - The sequel to the movie will have a similar name to the original movie, but they are different movie indeed, if two movies's titile are in high similarity but one or both of them ends with a number, the score will be penalized.
    - Since I am using Longest Common Subsequence as the similarity measure mertic, if two movies have short titles, it's easy to get a higher similarity between them, this will be penalized as well.
- Release time and genre scores are both shared 20% on the weighted score.
    - If one entry's release time or genre information not exist, I will only use title similaity for the weighted score, i.e. release time and genre scores take 0% on weighted score.
- I've tried various combinations and manually reviewed and compared the results of the different combinations. The results of this combination are relatively satisfactory to me.

Task 2

Blocking	Hash	Token
Reduction Ratio	0.994185	0.748263
Pairs Completeness	0.41176	0.882353

I found that for reduction ratio, hash > token, and for pairs completeness, hash < token.

I think the reason is that for the hash method, which uses `title_first_2_letters` to generate features, there are a small number of combinations of just two letters, which increases the Reduction Ratio and lowers the recall. For token method, bi-gram generates more features than hash, Reduction Ratio gets lower, and recall gets higher.

Task 3

Task 3.1

I cannot find a sutieable classes from the schema.org for the following attributes, I defined my own.

- title
- release date
- imdb-rating
- imdb-metascore
- imdb-votes
- gross-income
- cinematographer
- productionCompany (`ref:type schema:Class`)

The resulting model is as follows.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix schema: <http://schema.org/> .
@prefix my_ns: <http://dsci558.org/myfakenamespace#> .

my_ns:title rdf:type rdfs:Property ;
    rdfs:label "Property: film title";
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:text .

my_ns:release_date rdf:type rdfs:Property ;
    rdfs:label "Property: release date" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:date .

my_ns:imdb_rating rdf:type rdf:Property ;
    rdfs:label "Property: rating" ;
    rdfs:domain my_ns:Movie ;
```

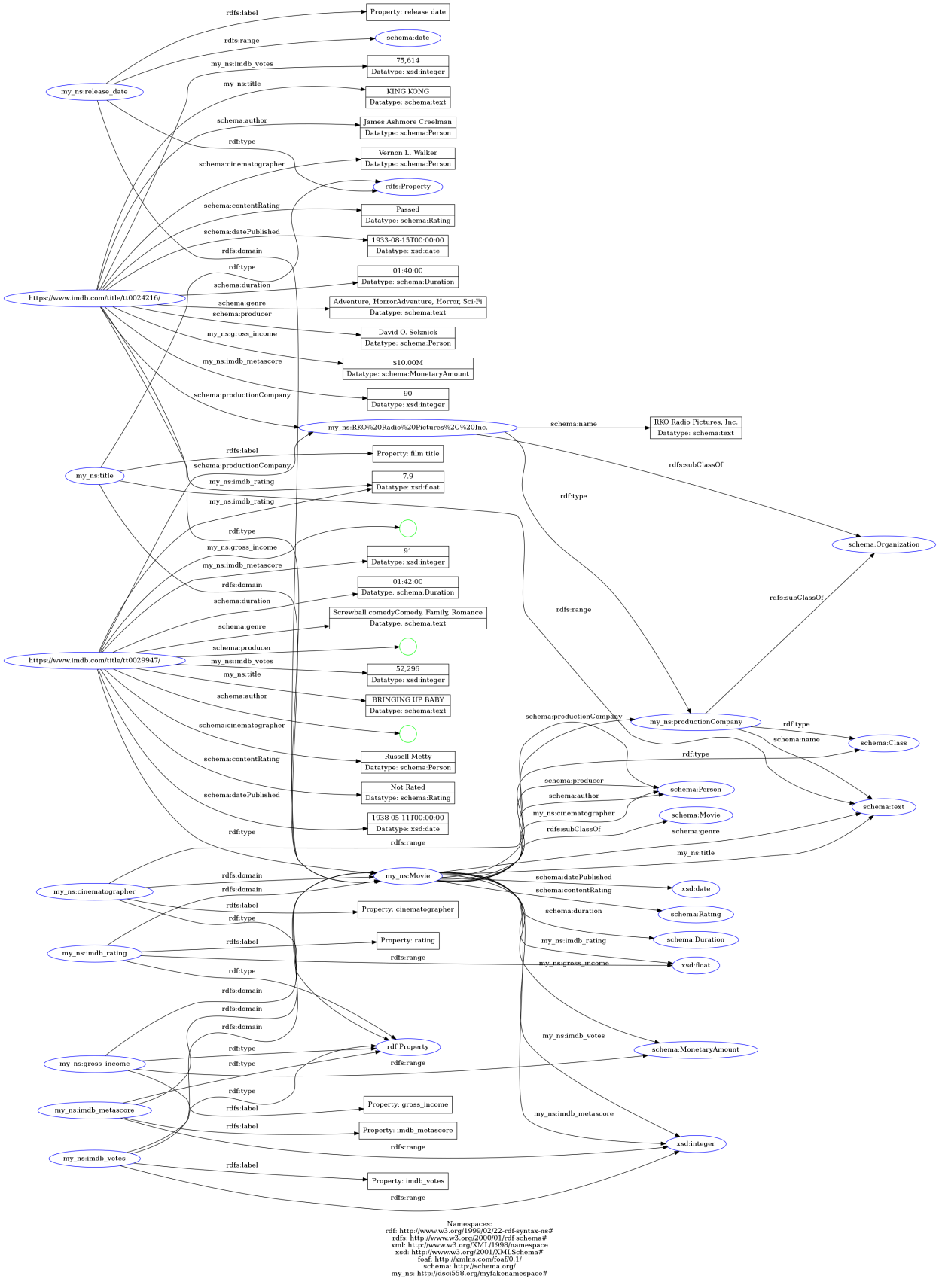
```
    rdfs:range xsd:float .

my_ns:imdb_metascore rdf:type rdf:Property ;
    rdfs:label "Property: imdb_metascore" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range xsd:integer .

my_ns:imdb_votes rdf:type rdf:Property ;
    rdfs:label "Property: imdb_votes";
    rdfs:domain my_ns:Movie ;
    rdfs:range xsd:integer .

my_ns:gross_income rdf:type rdf:Property ;
    rdfs:label "Property: gross_income" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:MonetaryAmount .

my_ns:cinematographer rdf:type rdf:Property ;
    rdfs:label "Property: cinematographer" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:Person .

### Production Company Class ####
my_ns:productionCompany rdf:type schema:Class ;
    rdfs:subClassOf schema:Organization ;
    schema:name schema:text .

#### Movie Class ####
my_ns:Movie rdf:type schema:Class ;
    rdfs:subClassOf schema:Movie ;
    my_ns:title schema:text ; # title aif
    schema:datePublished xsd:date ; # release_date afi
    schema:contentRating schema:Rating ; # certificate imdb
    schema:duration schema:Duration ; # runtime imdb
    schema:genre schema:text ; # genre both
    my_ns:imdb_rating xsd:float ; # imdb_rating
    my_ns:imdb_metascore xsd:integer ; #imdb_metascore
    my_ns:imdb_votes xsd:integer ; # imdb_votes
    my_ns:gross_income schema:MonetaryAmount ; #gross_income imdb
    schema:producer schema:Person ; # producer afi
    schema:author schema:Person ; # writer afi
    my_ns:cinematographer schema:Person ; #cinematographer afi
    schema:productionCompany my_ns:productionCompany ; # production_company
afi
```

Task 3.2

- Some movies have more than one production company, therefore cannot simply take the entire production company field.
- Certain attributes are exclusive in AFI : `time`, `release-date`, `producer`, `writer`, `cinematographer` and `production-company`
- Certain attributes are exclusive in IMDB: `certificate`, `runtime`, `imdb-rating`, `imdb-metascore`, `imdb-votes`, `gross-income`
- `genre` exists in both sources, so I merged the two together
- Since picked `title` rather than `name`, if one IMDB movie does not have a match, the `title` will be empty.
- I use `rdflib.term.BNode` to refer an empty field.

Task 3.3

Here is the graph I got:



This is the source code used to generate the grpah:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix schema: <http://schema.org/> .
@prefix my_ns: <http://dsci558.org/myfakenamespace#> .

my_ns:title rdf:type rdfs:Property ;
    rdfs:label "Property: film title";
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:text .

my_ns:release_date rdf:type rdfs:Property ;
    rdfs:label "Property: release date" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:date .

my_ns:imdb_rating rdf:type rdf:Property ;
    rdfs:label "Property: rating" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range xsd:float .

my_ns:imdb_metascore rdf:type rdf:Property ;
    rdfs:label "Property: imdb_metascore" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range xsd:integer .

my_ns:imdb_votes rdf:type rdf:Property ;
    rdfs:label "Property: imdb_votes";
    rdfs:domain my_ns:Movie ;
    rdfs:range xsd:integer .

my_ns:gross_income rdf:type rdf:Property ;
    rdfs:label "Property: gross_income" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:MonetaryAmount .

my_ns:cinematographer rdf:type rdf:Property ;
    rdfs:label "Property: cinematographer" ;
    rdfs:domain my_ns:Movie ;
    rdfs:range schema:Person .

### Production Company Class ####
my_ns:productionCompany rdf:type schema:Class ;
```

```
    rdfs:subClassOf schema:Organization ;
    schema:name schema:text .

#### Movie Class ####
my_ns:Movie rdf:type schema:Class ;
    rdfs:subClassOf schema:Movie ;
    my_ns:title schema:text ; # title aif
    schema:datePublished xsd:date ; # release_date afi
    schema:contentRating schema:Rating ; # certificate imdb
    schema:duration schema:Duration ; # runtime imdb
    schema:genre schema:text ; # genre both
    my_ns:imdb_rating xsd:float ; # imdb_rating
    my_ns:imdb_metascore xsd:integer ; #imdb_metascore
    my_ns:imdb_votes xsd:integer ; # imdb_votes
    my_ns:gross_income schema:MonetaryAmount ; #gross_income imdb
    schema:producer schema:Person ; # producer afi
    schema:author schema:Person ; # writer afi
    my_ns:cinematographer schema:Person ; #cinematographer afi
    schema:productionCompany my_ns:productionCompany . # production_company
afi

<http://dsci558.org/myfakenamespace#RKO%20Radio%20Pictures%2C%20Inc.> a
my_ns:productionCompany ;
    rdfs:subClassOf schema:Organization ;
    schema:name "RKO Radio Pictures, Inc."^^schema:text .

<https://www.imdb.com/title/tt0029947/> a my_ns:Movie ;
    my_ns:gross_income [ ] ;
    my_ns:imdb_metascore 91 ;
    my_ns:imdb_rating "7.9"^^xsd:float ;
    my_ns:imdb_votes "52,296"^^xsd:integer ;
    my_ns:title "BRINGING UP BABY "^^schema:text ;
    schema:author [ ] ;
    schema:cinematographer "Russell Metty"^^schema:Person ;
    schema:contentRating "Not Rated"^^schema:Rating ;
    schema:datePublished "1938-05-11T00:00:00"^^xsd:date ;
    schema:duration "01:42:00"^^schema:Duration ;
    schema:genre "Screwball comedyComedy, Family, Romance"^^schema:text ;
    schema:producer [ ] ;
    schema:productionCompany
<http://dsci558.org/myfakenamespace#RKO%20Radio%20Pictures%2C%20Inc.> .

<https://www.imdb.com/title/tt0024216/> a my_ns:Movie ;
    my_ns:gross_income "$10.00M"^^schema:MonetaryAmount ;
    my_ns:imdb_metascore 90 ;
    my_ns:imdb_rating "7.9"^^xsd:float ;
```

```
    my_ns:imdb_votes "75,614"^^xsd:integer ;
    my_ns:title "KING KONG "^^schema:text ;
    schema:author "James Ashmore Creelman"^^schema:Person ;
    schema:cinematographer "Vernon L. Walker"^^schema:Person ;
    schema:contentRating "Passed"^^schema:Rating ;
    schema:datePublished "1933-08-15T00:00:00"^^xsd:date ;
    schema:duration "01:40:00"^^schema:Duration ;
    schema:genre "Adventure, HorrorAdventure, Horror, Sci-Fi"^^schema:text ;
    schema:producer "David O. Selznick"^^schema:Person ;
    schema:productionCompany
<http://dsci558.org/myfakenamespace#RKO%20Radio%20Pictures%2C%20Inc.> .
```