

Разработка графовой базы данных игр

Студент: Соколов М.В
Группа: Р41142

Санкт-Петербург 2022

Предметная область

- Граф знаний по играм
- Информация о названии, описании, дате выхода и т.д.
- Обогащение за счет рецензий пользователей
- Возможное применение в системах по рекомендации игр



Источники

Основные:

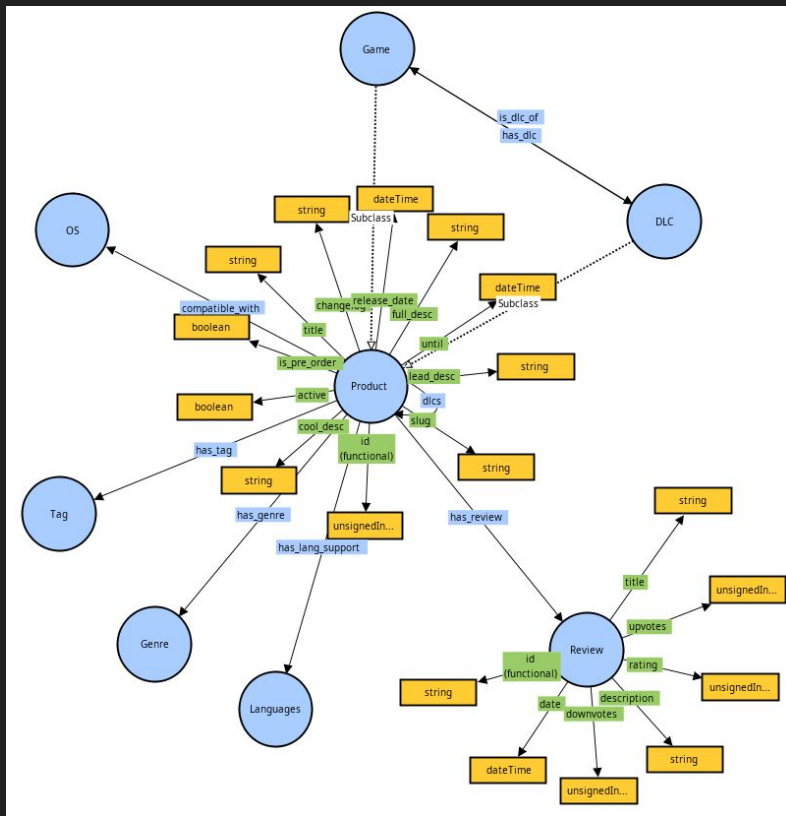
- GOG Games
- Steam

По возможности:

- Epic Games Store
- Uplay
- Origin



Разработанная онтология



Встреченные сложности

- Невозможность получить информацию с большинства источников.
Остается только: GOG Games
- Необходимость в предобработке полученных данных
- Продумывание дизайна итоговой онтологии: классов, свойств и связей

Наполнение графа

Metrics

Axiom	30153
Logical axiom count	27184
Declaration axioms count	2935
Class count	8
Object property count	8
Data property count	18
Individual count	2901
Annotation Property count	1

Class axioms

SubClassOf	2
EquivalentClasses	0
DisjointClasses	5
GCI count	0
Hidden GCI Count	0

Object property axioms

SubObjectPropertyOf	2
EquivalentObjectProperties	0
InverseObjectProperties	1
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0

Компетентностные вопросы

1. Топ 10 жанров
2. Топ 10 тегов
3. Количество выпущенных игр в жанре “инди” по годам
4. Средний рейтинг пользовательский рецензий по годам
5. Какая игра схожа с выбранной?

Reasoning

- Использован HermiT
- Исправлены даты в исходных данных

Документация

made by **pyLODE** 2.13.2

Games

URI
<http://www.semanticweb.org/naymoll/ontologies/2022/3/games.owl>

Ontology RDF
[RDF \(xml\)](#)

Table of Contents

- [1. Classes](#)
- [2. Object Properties](#)
- [3. Functional Properties](#)
- [4. Datatype Properties](#)
- [5. Named Individuals](#)
- [6. Namespaces](#)
- [7. Legend](#)

Overview

Pictures say 1,000 words

Figure 1: Ontology overview

Classes

[DLC](#) [Game](#) [Genre](#) [Languages](#) [OS](#) [Product](#) [Review](#) [Tag](#)

DLC	# Classes
URI http://www.semanticweb.org/naymoll/ontologies/2022/3/games.owl#DLC	

VoID

- void:triples 30155
- void:properties 31
- void:classes 13
- void:entities 5798

SHACL

```
[ ] from pyshacl import validate
```

```
    r = validate(graph,  
                  inference='rdfs',  
                  abort_on_first=False,  
                  allow_infos=False,  
                  allow_warnings=False,  
                  meta_shacl=False,  
                  advanced=True,  
                  js=False,  
                  debug=False)
```

```
[ ] conforms, results_graph, results_text = r  
    print(results_text)
```

```
Validation Report  
Conforms: True
```

SPARQL запросы

```
1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2
3 PREFIX on: <http://www.semanticweb.org/naymoll/ontologies/2022/3/games.owl#>
4 PREFIX pr: <http://www.semanticweb.org/naymoll/ontologies/2022/3/games.owl/product#>
5 PREFIX rv: <http://www.semanticweb.org/naymoll/ontologies/2022/3/games.owl/review#>
6
7 # Средний рейтинг игр по годам
8
9 SELECT ?Year (COUNT(?Rating) AS ?Cout) (AVG(?Rating) AS ?Average)
10 WHERE {
11     ?Game a on:Game ;
12           pr:has_review ?Review .
13
14     ?Review a on:Review ;
15            rv:rating ?Rating ;
16            rv:date ?Date .
17
18     BIND(YEAR(?Date) AS ?Year)
19 }
20 GROUP BY (?Year)
21 ORDER BY DESC(?Year)
```

Run to File Text Visualize 15 Results, 233 ms

Year	Cout	Average
2022	201	3.736318407960199
2021	448	3.3660714285714284
2020	511	3.86692759295499
2019	328	3.8140243902439024
2018	197	3.934010152284264
2017	70	3.6142857142857143
2016	45	4.0
2015	17	3.8823529411764706
2014	35	4.3428571428571425
2013	40	3.925
2012	63	4.317460317460317
2011	80	4.2625
2010	72	4.416666666666667
2009	102	4.450980392156863
2008	106	4.339622641509434

Embeddings - тренировка модели

Данные:

- Тренировочная выборка: 20155
- Тестовая выборка: 10000
- Модель: ComplEx

Полученные метрики:

- MRR: 0.37
- MR: 647.55
- Hits@10: 0.47
- Hits@3: 0.40
- Hits@1: 0.31

```
from ampligraph.latent_features import ComplEx

model = ComplEx(batches_count=50,
                epochs=300,
                k=100,
                eta=20,
                optimizer='adam',
                optimizer_params={'lr': 1e-4},
                loss='multiclass_nll',
                regularizer='LP',
                regularizer_params={'p': 3, 'lambda': 1e-5},
                seed=0,
                verbose=True)
```

Embeddings - классификация

Предсказание оценки рецензии по ее содержанию.

Данные:

- Модель: XGBClassifier
- Тренировочная выборка: 1954 рецензий
- Тестовая выборка: 488 рецензий

Итоговая точность: ~75%