Recommendation System on Video Game using Ontology

Semantic Web
Final Project

Submitted in partial fulfillment
of the requirements for the degree of
MSc Natural Language Processing

by

Jorge Luis Vasquez Mercado, Colm Rooney,
Soklong Him, Juba Ait Abdelmalek

Instructed by:
Mathieu d'Aquin

Academic year 2021-2022

Contents

Chapter 1

Introduction

This project report details the culmination of our work to create a User-Game recommendation system using RDF triples and DBpedia. As a team we created both vocabularies and ontology's for Video Game and User, linked our graphed data to DBpedia triples and extracted data using SPARQL queries to fill our Video Game and User instances. This project deemed to be challenging, as a team we encountered issues and difficulties throughout however, we successfully managed to cooperate together and isolate solutions for our problems. The name we choose for our team is Cloud-Nine, which was used as an IRL to identify our ontology's during creation and querying.

1.1 Scope of Project

Taking time constraints into account, as a team we discussed that it would not be feasible to created 'the perfect system' covering all aspects of Video Games and User preferences. Due to this we agreed it was important to plan out our goals for the project and what we deemed important to focus our overall attention on.

Taking inspiration from previously created data, we based the our VideoGame ontology on the pre-existing ontology located on DBpedia. Moreover, for the User ontology we created based on our personal views on which aspects best define a gamer from our own experiences.

1.2 Tools

Throughout the duration of this project, as a team we made use of various online software and applications to achieve high quality results and efficent version control

- Protégé: Protegé was used to create the vocabularies and ontologies for Video Game and User.
- Github: Github was used to collate and share our work together.
- Discord: Discord was used to communicate amongst ourselves.
- Blazegraph: Blazegraph was used for testing SPARQL Query with our vocabulary.
- WebVOWL: Web Vowl is an excellant web applicant which allows us to visualise and modify our ontologies.

Below are the applications we used to extract data from DBpedia. We used Python and SPARQL construct Query along with some various pythonian libraries:

- RDFLib: useful for building RDF graphs.
- SPARQLWrapper: Used for creating queries and converting the extracted data into turtle format.

Chapter 2

Methodology

Before we began building our ontology, we discussed the task, and split into two groups: One tasked with building the User ontology and the other constructing the Video Game ontology.

As a group we reviewed previously created ontology's for Games and User to gain an understanding of the construction and proper modes of practice to take.

Taking inspiration from previously created data, we based our Video Game ontology on the pre-existing ontology located on DBpedia[1]. Moreover, for the User ontology we created this based on our personal views on which aspects best define a gamer from our own experiences.

2.1 Define the Ontology

First we created a video game vocabulary which defined the structure our of video game ontology. We modeled our vocabulary around information we viewed as quantifiable when recommending games to a user. This led us to creating properties which were easily matched to game quality and user preferences:

- Game rating,
- where the inspiration of the game came from (for ex. historical event, book, movie),
- Which genre and series it belonged to
- Publisher, as a user may enjoy a game created by certain developers
- Which gaming platform is the game supported on, this was matched to what console the User owned.

Following the completion of the vocabulary, we expanded upon it to define our ontology.

As in figure 1, Our ontology for video games consisted of 26 Class definition and 22 properties.

Next, We also defined the User Ontology. The User ontology consisted of 10 classes and 10 object properties, Some properties we included were; play history, preferred play modes, platforms they own, gender etc. We agreed that the ontology we constructed provided us with a useful vocabulary to define specific users and their preferences.

After completing the VG and User vocabulary and ontology. we created a list of players/users containing data on their usernames, play history and previous games they enjoyed playing as in figure 2.

2.2 Data Extracting

We used SPARQLWrapper library in python to construct the rdf graph from DBpedia Ontology. As in figure 3a. we construct the graph of video game where a video game is a DBpedia ontology video game. Video game which has dbp:title , dbo:genre and some optional property on dbp:platforms , dbo:abstract dbo:developer. we also set filter on abstract by selecting only English langauge. Then, we used instand graph from RDFLib library in Python to create the graph by put the result from above query in turtle format.

2.3 Designing and creating SPARQL query

This step was an essential part of our project. not only did we create queries to get the video game recommendations but it also allowed us testing about the quality of our vocabulary. we use Blazegraph as the triplestore and than we applied our query into this triplestore to see the result. As we can see in figure 3b for recommendation query.

[1] https://www.dbpedia.org

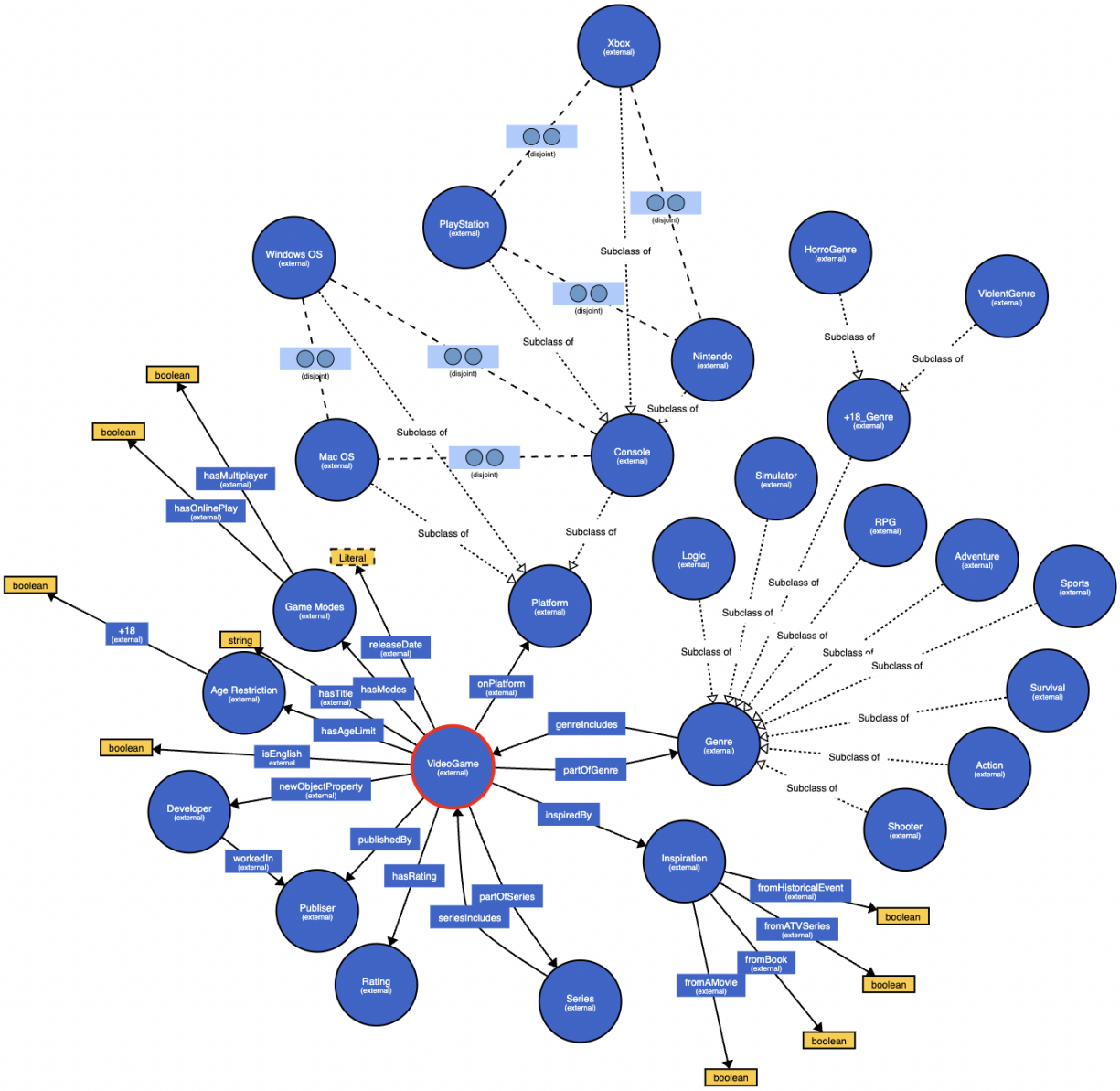

Figure 1: Video Game Ontology

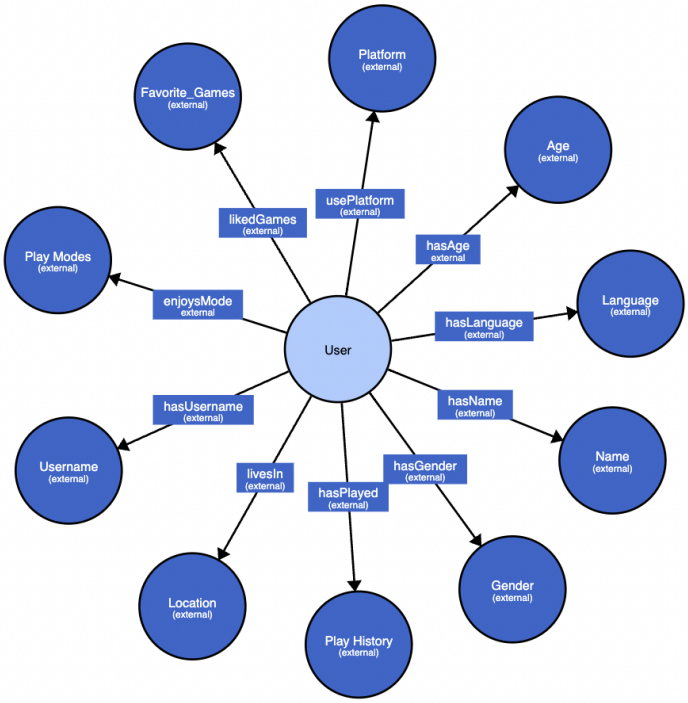

Figure 2: User Ontology

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX schema: <http://schema.org/>
PREFIX c9s:   <http://cloudnine.sw/schema#>
CONSTRUCT {
    ?game  c9s:hasTitle ?title .
    ?game c9s:onPlatform ?platform.
    ?game  c9s:abstract ?abstract.
    ?game c9s:partOfGenre ?genre.
    ?game c9s:developedBy ?developer.
    ?game c9s:publishedBy ?publisher.
    ?game c9s:releaseDate ?date.
}WHERE {
    ?game a dbo:VideoGame;
    dbp:title ?title.
    OPTIONAL{?game dbp:platforms ?platform}.
    OPTIONAL{?game dbo:abstract ?abstract}.
    ?game dbo:genre ?genre.
    OPTIONAL{?game dbo:developer ?developer}.
    OPTIONAL{?game dbp:publisher ?publisher}.
    OPTIONAL{?game dbp:lastReleaseDate ?date}.
    FILTER (lang(?abstract) = 'en')
  }
```

(a) Construct Query from DBpedia

```
PREFIX c9s: <http://cloudnine.sw/schema#>
PREFIX c9e:         <http://cloudnine.sw/entity/>

SELECT DISTINCT ?title ?newGame ?genre ?platform
WHERE{
  ?user c9s:isPlayer c9e:ID02.
  c9e:ID02 c9s:hasPlayed ?game.
  c9e:ID02 c9s:usePlatform ?platform.
  ?game c9s:partOfGenre ?genre.
  ?newGame c9s:onPlatform ?platform.
  ?newGame c9s:partOfGenre ?genre.
  ?newGame c9s:hasTitle ?title.
  FILTER (?newGame!=?game)
}ORDER BY ?newGame
```

(b) Recommendation Query

Figure 3: SPARQL Queries

Chapter 3

Results and Conclusion

In this part we will show about the result that got from the query which we discussed in section 2.3 and final conclusion after finish this project.

3.1 Results

Here is the result of our recommendation system base on condition in our query.

title	newGame	genre	platform
Borderlands	<http://dbpedia.org/resource/Borderlands_(video_game)>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>
Bulletstorm	<http://dbpedia.org/resource/Bulletstorm>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>
Far Cry 5	<http://dbpedia.org/resource/Far_Cry_5>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>
Metro: Last Light	<http://dbpedia.org/resource/Metro:_Last_Light>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>
Pier Solar and the Great Architects	<http://dbpedia.org/resource/Pier_Solar_and_the_Great_Architects>	<http://dbpedia.org/resource/Role-playing_game>	<http://dbpedia.org/resource/PlayStation_4>
Thief	<http://dbpedia.org/resource/Thief_(2014_video_game)>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>
Wolfenstein II: The New Colossus	<http://dbpedia.org/resource/Wolfenstein_II:_The_New_Colossus>	<http://dbpedia.org/resource/First-person_shooter>	<http://dbpedia.org/resource/PlayStation_4>

Figure 4: Result of Query

3.2 Conclusion

Throughout the past month working on this project we encountered multies difficulties, such as defining the vocabularies in a way which was compact and consistent as well as technical issue with the protégé software. We we unable to run protegé on Mac M1 chipset, thankfully two of our team mates were using Windows OS which supported the application. Additonally we can attribute the limit of the scope of this project to time contraints, as a group we could have improved the quality of our ontology's and implemented a frontend application for this project if we had more time.

However, we still accomplished our goal with high quality and solved the problems mentioned. As a team we gained a lot of experience and knowledge of the semantic web, such as defining and creating vocabularies and ontologies, building the RDF graphs, associating those graphs with a proper vocabulary and query writing.

For our future perspective, we hope that we can continue the update on this project to more reliable and scalable which can integrate with other existing ontology and also with more scenario than this version.