Autonomous Software Agents – The World Avatar

Rishabh Alexander John

School of Computer Science and Engineering

Prof Markus Kraft
Ms. Mei Qi Lim
Ms. Srishti Ganguly
Mr Wilson Ang

Cambridge CARES

Abstract - Real-time data from publicly available government APIs can be studied to infer their influence on the real environment. To facilitate such analysis, autonomous computational software agents can integrate and use data from a knowledge graph to draw further conclusions. This paper presents three such agents that run on a dynamic knowledge graph as part of The World Avatar project. To accurately represent the information obtained, ontologies are developed to define key concepts and the relationships between the different static and dynamic data. The timeseries is stored in a relational database and linked to various subject-predicate-object triples making it connected, discoverable, and query able. One of the agents developed is the CarparkAgent which obtains live statistics of public carparks in Singapore. The agent instantiates data such as the number of available lots, the lot type, the geographic coordinates of the car park, and the pricing of car parks throughout the day. This data may be used to study footfall in Shopping Centres and analyze times of highest traffic. The second agent developed is a Schools Agent that instantiates a wide range of information on government schools in Singapore including key programmes and co-curriculars that are offered. Lastly, a Hawker Centre Agent is developed that instantiates data related to the different licensed stalls that are registered in government hawker centres in Singapore. The agent also instantiates information on the stalls' gradings, and demerit points accrued over time. All the datasets are linked to geospatial representations of buildings in Singapore.

1 INTRODUCTION

Today's ever advancing digital age and the onset of Industry 4.0 has triggered changes in the way we use technology to learn more about our environment and how it can be bettered. Big Data has highlighted the importance of essential information and offers greater statistical power thus enabling us to be more informed in modern day decision making.[11] Real-world Data in particular serves as an interesting avenue to explore – as it lets us glean key insights as to how seemingly everyday figures can play a larger role than meets the eye.

The World Avatar project was envisioned to serve as a host for a multitude of cross domain information and a platform to enable interoperability. Resting on the foundations of knowledge graph technology, The World Avatar can amass vast amounts of data on a host of domains for analysis and inference. This project was designed to serve as a digital twin of the real-world through which key studies and statistical inferences can be drawn.

In order to populate The World Avatar knowledge graph with more information, autonomous software input agents are designed. Through this research project, a host of such agents were developed, tested, and run to increase the breadth of information in the knowledge graph.

1.1 AIM & PRINCIPLE

The aim of this research project was to contribute to the making and expansion of The World Avatar base world. The main focus was the development of various software agents that would serve as the proverbial "Eyes and Ears" of The World Avatar and help input data into the knowledge graph.

Over the past year, three autonomous software agents were developed to fetch information from publicly available and free Application Programming Interfaces (APIs). The information obtained was subsequently cleaned sorted and

instantiated into the knowledge graph. For this, an open-source triple store called Blazegraph [13] is used. It is an ultra-high performace graph database supporting up to 50 billion edges on a single machine. Static, semantic data in the form of Subject-Predicate-Object triples were instantiated in Blazegraph helping connect and form relationships between the different types of data. Various times series associated with the data were stored in PostgreSQL [12]. PostgreSQL is an open-source object-relational database that emphasizes extensibility and Structured Query Language (SQL) compliance. [16]

The three agents developed were a Carpark Agent, a Schools Agent as well as a Hawker Centre Agent. Respective Ontologies were also developed for the three agents to help define key concepts regarding the data stored.

1.2 DATA SOURCES

The source of data for these agents were the APIs hosted by Singapore Government Data.[1][2][3][4][5][6][7]

The first data source used by the Carpark Agent is the Land Transport Authority (LTA)'s DataMall. The API feeds real time information back to the agent and the data format is JavaScript Object Notation (JSON) which makes it convenient for data cleaning and sorting. Use of the API is protected via an account key.

The second data source is data.gov.sg, an initiative under GovTech that serves as a one stop portal for static and dynamic datasets provided by public agencies. The data is easy to interpret and it too is returned in a JSON format. This source is utilized by the Schools and Hawker Centre Agent.

Both data sources fall under the Singapore Open Data License and constitute acceptance of the Terms of Service for API and Software Development Kit(SDK).

3 THE CARPARK AGENT

The Carpark Agent was the first autonomous software agent that was developed as part of this project. Key reference was taken from the CARES DAVIS Weather Station Agent[8], and it was subsequently modelled similarly to it.

The focus of this agent was to obtain information regarding real time availability of lots in car parks around Singapore. The first step was developing an ontology – OntoCarPark. The ontology defines the different static and dynamic data and the different relationships between them. A TBox Generator [9], a Java module developed under The World Avatar project, was subsequently used to attain the attributes from the initial ".csv" file to create the ".owl" file needed to define the relationships. For the Carpark Agent, static data that was stored were attributes such as Location (Latitude & Longitude), the Agency in charge of the car park and the Carpark ID. The prices of the car park over different days (weekday, weekend, Public Holiday) were also obtained. Besides the Unique ID, the name of the car park was also stored.

Public carparks are segregated according to the different types of lots they possess and so this attribute was stored as a subclass of the LotType attribute. The three types of lot types were Cars, Motorcycles or Heavy Vehicles. LTA defines Heavy Vehicles as vehicles with a Maximum Laden Weight (MLW) higher than 3500kg.[14]

The most critical data attribute that was stored was the number of Available Lots. Since this attribute is real-time data, it was stored as a time series.

Figure 1 details the ontology that was developed with concepts shown in the boxes and the arrows indicating the relationships between the concepts.

The development of all three agents follows a similar four-pronged approach. Detailed below are the four Java classes developed for the Carpark Agent.

3.1 API CONNECTOR

The API Connector class is designed to retrieve the raw data from the API source. The agent verifies the Uniform Resource Locator (URL) as well as the API Account Key before retrieving the data using a CloseableHTTPClient. In the event of an unexpected server error, a notification would also be provided.

Information was obtained from two datasets: one provided information on the carpark and its available lots, while the other provided prices of these car parks over the different days of the week, fetched as JSON raw data.

3.2 API INPUT AGENT

The API Input Agent class is responsible for instantiating the timeseries i.e., the Available Lots, using the TimeSeries Client [10] library developed under The World Avatar project. A list of all the car parks is stored in a mappings folder and that is used to initialize IRIs for the various car parks. The readings are then parsed through and are selectively sieved to obtain the Available Lots for each car park. The data is stored in a HashMap with the car park mapping, the available Lots as well as the current timestamp

3.3 API QUERY BUILDER

The API Query Builder class creates triples to store the data in a query able manner. The relationships and classes are defined as IRIs while verifying the update and query endpoints.

The IRIs from the mapping folder are obtained and run through individually. If the car park instance was not previously instantiated, the IRI is instantiated to be of type Carpark and the various static data attributes are stored as triples.

E.g.:

Carpark A is instantiated as a class of type Carpark. This car park IRI is matched to its ID with the relationship “hasID”.

In a similar fashion, the rest of the available data is instantiated with the appropriate relationships to the Car park as is specified in the ontology.

For the prices of the car park contained in a different dataset, the car parks must be matched according to their label name. This cross-referencing is done using FuzzyMatching. The Fuzzy Wuzzy Java implementation, available as a Maven Dependency was used for this purpose. This method compares two strings and throws out a similarity ratio. The ratios can be either Simple, Partial or Token sorted.

For the car park labels, we use a mix of token and partial ratios to obtain perfect consistency in initializing the car parks accurately. Once a match is obtained, the prices are obtained from the dataset, matched with the car park IRI under their respective relations i.e., hasWeekdayRates, hasSaturdayRates and their data triples are created.

3.4 API AGENT LAUNCHER

The API Agent Launcher serves as the coordinating class that calls the respective methods in sequential order. It first verifies the appropriate configuration files.

The Agent Launcher then initializes the three classes described above: API Connector, API Input Agent and API Query Builder. Using these classes in order, the raw data is obtained from the API, parsed into relevant triples, and updated into the knowledge graph.

4 SCHOOLS AGENT

The Schools Agent was developed to instantiate information regarding public schools into The World Avatar. Data was drawn from the data.gov.sg portal and it contained a host of information including but not limited to the school faculty, location, and the student classification (Boys, Girls, CO-ED). The data for school was all static and hence did not require any time series instantiation. The ontology developed was solely to store triples for this static data.

In order to extend the information regarding the schools, two more datasets were used which contained the specialized MOE programmes that certain schools offered as well as the different types of CCA groups and the types of CCAs within that group that were available for students to pursue.

The development of the Agent was similar to the Carpark Agent with the only exception being the absence of the Input Agent file as no timeseries client was required here.

Fuzzy Matching was again conducted in the QueryBuilder file in order to accurately match the three datasets and the school label was used as a common reference before the data triples were constructed.

5 HAWKER CENTRE AGENT

The Hawker Centre agent was developed to instantiate information on various hawker centres in Singapore. The data was obtained through two static datasets from data.gov.sg. The first dataset contained information on individual hawker centres and included data attributes such as the Owner of the Hawker Centre (Housing & Development Board or Government), the number of food stalls, the type of Hawker Centre whilst the second dataset contained information on all the NEA licensed food stalls in Singapore.

The stalls are instantiated and linked to the hawker centre instances whilst each have their individual data attributes such as the NEA grading, Licensee name and License Number.

Fuzzy Matching is conducted in the Query Builder file to map the individual stalls to the Hawker Centres that they belong to. The label of the Hawker Centre as well as the Address attribute of the Stall are compared to determine a match.

The Average Grading of the Hawker Centre is also computed by taking the average grade of its stalls. NEA gradings for the stalls are provided as either an A, B or C grade and using this information, an average grading for the entire hawker centre is computed by considering the most frequently occurring grade.

Both the Schools and Hawker Centre Agents are developed such that they fetch information from various APIs on a regular basis which ensures that the agents are always able to retrieve the most recent data and remain up to date.

6 CONCLUSION

The goal of the research project was to develop autonomous agents to expand the knowledge graph of The World Avatar, and this was successfully achieved with the three agents able to retrieve and instantiate relevant data. Limitations arise in the current project due to its dependence on static datasets which might make the information fetched obsolete. Improvements can be made to obtain real-time data and keep the knowledge graph as relevant as possible.

The three agents contain information surrounding publicly frequented locations and potential use cases could arise pertaining to footfall study and analysis. This too would be aided by the addition of more dynamic data points.

ACKNOWLEDGMENT

I would like to extend my gratitude towards Prof. Markus Kraft for providing me the opportunity to contribute to this exciting project. His continued guidance and oversight were extremely valuable and helped shape this fruitful research experience.

I would like to acknowledge the guidance of Ms. Mei Qi Lim through this journey. Her advice and direction throughout the development process was key to the success of this project.

I would like to acknowledge the support and tutelage of Ms. Srishti Ganguly and Mr. Wilson Ang, with whom I had the pleasure of working closely with. Their supervision and effort helped create a healthy learning environment in which I was able to thrive.

I would like to acknowledge the funding support from Nanyang Technological University – URECA Undergraduate Research Programme for this research project.

REFERENCES

[1] Land Transport Authority (LTA), "Carpark Availability (Dynamic) Dataset," Retrieved from Datamall website: https://datamall.lta.gov.sg/

[2] Singapore Land Authority, "Carpark Rates," [Dataset], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/carpark-rates?view_id=ec04f2a7-735a-4c58-8cd4-ec758ca5f1b7&resource_id=85207289-6ae7-4a56-9066-e6090a3684a5

[3] Singapore Department of Statistics, "School Directory and Information," [Data set], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/school-directory-and-information

[4] Singapore Department of Statistics, "School Directory and Information," [Data set], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/school-directory-and-information?resource_id=dd7a056a-49fa-4854-bd9a-c4e1a88f1181

[5] Singapore Department of Statistics, "School Directory and Information," [Data set], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/school-directory-and-information?resource_id=9a94c7ed-710b-4ba5-8e01-8588f129efcc

[6] Singapore Food Agency, "List of Government Markets & Hawker Centres," [Data set], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/list-of-government-markets-hawker-centres

[7] National Environment Agency, "List of NEA Licensed Eating Establishments with Grades, Demerit Points, and Suspension History," [Data set], Retrieved from Data.gov.sg: https://data.gov.sg/dataset/list-of-nea-licensed-eating-establishments-with-grades-demerit-points-and-suspension-history

[8] Cambridge Cares, "The World Avatar: CARESWeatherStationAgent" [GitHub repository], Retrieved from GitHub: https://github.com/cambridge-cares/TheWorldAvatar/tree/main/Agents/CARESWeatherStationAgent

[9] Cambridge Cares, "The World Avatar: EntityRDFizer/TBoxGenerator" [GitHub repository], Retrieved from GitHub: https://github.com/cambridge-cares/TheWorldAvatar/tree/main/EntityRDFizer/entityrdfizer/tboxgenerator

[10] Cambridge Cares, "The World Avatar: TimeSeriesClient.java" [GitHub repository], Retrieved from GitHub: https://github.com/cambridge-cares/TheWorldAvatar/blob/main/JPS_BASE_LIB/src/main/java/uk/ac/cam/cares/jps/base/timeseries/TimeSeriesClient.java

[11] Sydle. (2023). Big Data: Definition, Importance, and Types. Retrieved from

https://www.sydle.com/blog/big-data-definition-importance-and-types-614b791388e600016afa7fc3

[12] PostgreSQL Global Development Group. (n.d.). PostgreSQL. Retrieved from https://www.postgresql.org/

[13] SYSTAP LLC. (n.d.). Blazegraph. Retrieved from https://www.blazegraph.com/

[14] One Motoring. "Commercial Vehicles." [Online]. Available: https://onemotoring.lta.gov.sg/content/onemotoring/home/driving/commercial-vehicles.html. [Accessed: June 16, 2023].

[15] Xdrop. (n.d.). fuzzywuzzy. [GitHub repository]. Available: https://github.com/xdrop/fuzzywuzzy

[16] Wikipedia. "PostgreSQL." [Online]. Available: https://en.wikipedia.org/wiki/PostgreSQL. [Accessed: June 16, 2023].