Northwestern MSDS-459 Knowledge Engineering

Assignment 3: Database Implementation

Andrew Stevens

May 1, 2022

(1) Abstract

This stage of the project has focused on the construction of the graph database within postgresql, using the AGE extension. In order to make progress in this area, data cleaning and engineering have been the bulk of the work. The data collected from online needed to be recognized & throughput to formats that are organized in a useful manner within the database. The result is a collection of node types and relationship types that compose a structure around tesla as an auto manufacturer, its competitors, and its suppliers.

(2) Introduction

Traditional relational databases have been crucial to modern businesses and highly reliable, though not amenable to complex analytics. Simple queries can be run efficiently over large amounts of data, but the more complex analysis required for conducting network science and building out deep relationships requires something more relational – a graph database or some sort of engine sitting atop an RDBMS.

(3) Literature review

One of the more interesting resources provided for the potential increase of usability of the current scraped data is Ampligraph (Costabello, 2019). This package has the capability of predicting links within a knowledge graph, and may be able to fill in the gaps that exist in the raw scraped data. OpenKE would be considered a secondary candidate to support this research as it seems to require a heavier lift for usability. Finally, Song (2019) provides an approach for enterprise knowledge “build and query” with a methodology that more relies on structured collected data vice crawled and scraped data that is not as easily cleaned.

(4) Methods

The automated crawling for scraping of internet data is, for now, complete. In order to acquire and connect more useful data surrounding a company and its offerings, curated data provides much more value with less effort. CSIMarket hosts data on companies, their competitors, and their suppliers. I was able to pull down a clean set of companies and their basic financial info that supply Tesla. I have yet to find a source which clarifies which parts (within which domain) companies supply to tesla, and whether it's in the correct product line (as the competitor view includes Sunpower which is not relevant to this study).

With the help of Carlson's code, I was able to develop functions for sending pieces of the collected json or csv data to the graph, creating nodes and edges. The node creation function accepts a label for node type and a dictionary of attributes, then returns a cypher query to create the node

```
def create_node_query(lbl, attr_dict: dict = {}):
    attrs = '{' + ','.join([str(item[0]) + ":'" + str(item[1]) + "'" for item in
attr_dict.items()]) + '}'
    cQtmp = f"$$ CREATE (:{lbl} {attrs}"
    cQtmp = cQtmp + f") $$) as (n agtype);"
    return cQtmp
```

The edge query requires data on each of the two vertices, and classifies the edge as a relationship type. Attributes could also easily be added to the generated query, but have not been necessary within the current graph.

```
def create_edge_query(lblA,lblB,filterAkey,filterA,filterBkey,filterB,relType):
    cQtmp = f"$$ MATCH (a:{lblA}), (b:{lblB}) WHERE a.{filterAkey}='{filterA}'
AND b.{filterBkey}='{filterB}'"
    cQtmp = cQtmp + f" CREATE (a)-[rel:{relType}]->(b) RETURN rel $$) as (e
agtype);"
    return cQtmp
```

(5) Results

The resultant data of crawling and scraping websites has not yet proven useful. NLP entity and feature extraction is able to tie documents to some of the companies, but much of the content is either disconnected or irrelevant. The data has been sent to the postgres relational tables and cleaned some, but not transformed into nodes & edges. Initialization is required prior to sending queries to the graph:

```
LOAD '$libdir/plugins/age';
SET search_path = ag_catalog, '$user', public;
```

The structured data on companies, their products and other metadata has fed the following resulting nodes:

Table 1: Node types and counts

"label"	"count"
"""Year"""	"31"
"""Category"""	"8"
"""Make"""	"161"
"""Model"""	"1222"
"""Company"""	"243"
"""Location"""	"28"

Query:

```
select *
from cypher('tesla5forces',
    $$ MATCH (c)
    return label(c), count(*) as ct
$$) as (label agtype, count agtype)
```

Table 2: Edge types and counts

"rel_type"	"count"
"""MANUFACTURED_BY"""	"1265"
"""AVAILABLE_IN"""	"9730"
"""CATEGORIZED_AS"""	"1524"
"""COMPETES_WITH"""	"159"
"""PARENT_OF"""	"5"
"""SUPPLIES_TO"""	"172"
"""HQ_IN"""	"5"

Query:

```
select *
from cypher('tesla5forces', $$
    MATCH (n)-[r]->()
    RETURN label(r), count(*) as ct
$$) as (rel_type agtype, count agtype)
```

(6) Conclusions

The graph now has a rough foundation to be built out, with companies and some products. More industry information will need to be curated and fetched from more structured sources such as wikidata. This will allow for a more solid backbone to be carefully constructed as an ontology with some aspect of taxonomy to structure the nodes and their relationships.

(7) References

1. COSTABELLO, L., PAI, S., VAN, C. L., MCGRATH, R., AND MCCARTHY, N. AmpliGraph: a Library for Representation Learning on Knowledge Graphs, Mar. 2019.

2. Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

3. LiuQiao,LiYang,DuanHong,LiuYao,QinZhiguang. Knowledge Graph Construction Techniques[J]. Journal of Computer Research and Development, 2016, 53(3): 582-600.

4. Song, Dezhao, et al. "Building and Querying an Enterprise Knowledge Graph." IEEE Transactions on Services Computing, vol. 12, no. 3, May 2019, pp. 356–69. DOI.org (Crossref), https://doi.org/10.1109/TSC.2017.2711600.

5. Tesla Inc 's Suppliers, Size by Company for the 12 Months Ending in Mar 31 2022. https://csimarket.com/stocks/competitionNO9.php?supply&code=TSLA. Accessed 23 May 2022..

6. CIGraphs. (n.d.). CIGRAPHS/Carlson-Adafruit. GitHub. Retrieved May 23, 2022, from https://github.com/CIGraphs/carlson-adafruit