Australian
National
University

Entity Reconciliation In Knowledge Graphs

Haoran(Rohan) Song
u6688461@anu.edu.au

Supervised by:

Sergio Rodríguez Méndez

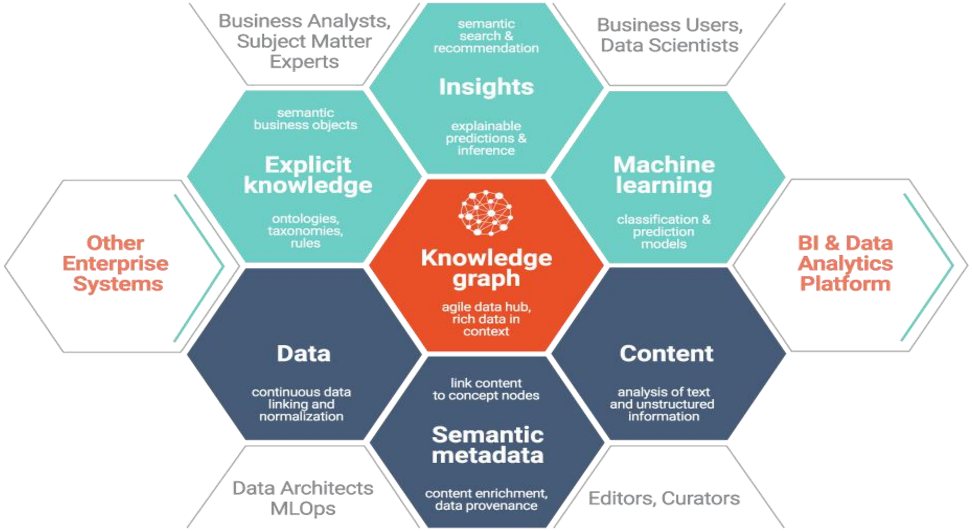
Armin Haller

1. Background

1.1 What is a Knowledge Graph

The knowledge graph (KG) represents a collection of interlinked descriptions of entities – real-world objects and events, or abstract concepts (e.g., documents)[1].

Image1. What is a Knowledge Graph[1]


Business Analysts, Subject Matter Experts
semantic search & recommendation
Insights
explainable predictions & inference
Business Users, Data Scientists
semantic business objects
Explicit knowledge
ontologies, taxonomies, rules
Machine learning
classification & prediction models
Other Enterprise Systems
Knowledge graph
agile data hub, rich data in context
BI & Data Analytics Platform
Data
continuous data linking and normalization
Content
analysis of text and unstructured information
link content to concept nodes
Semantic metadata
content enrichment, data provenance
Data Architects MLOps
Editors, Curators

[1] "What is a Knowledge Graph? | Ontotext Fundamentals", Ontotext, 2020. [Online].Available:https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/. [Accessed: 26- Oct- 2020].

1. Background

1.1 What is a Knowledge Graph

Image2. Dbpedia[2].

Image3. WikiData[3].

Image4. Microsoft Academic Knowledge Graph[4].

Sources:
[2] https://wiki.dbpedia.org/
[3] https://www.wikidata.org/wiki/Wikidata:Main_Page/
[4] http://ma-graph.org/

1. Background

1.2 What is Entity Reconciliation

ER is an operational intelligence process through which organizations can unify different and heterogeneous data sources in order to correlate possible matches of non-obvious entities[1].

[1] "What is a Knowledge Graph? | Ontotext Fundamentals", Ontotext, 2020. [Online].Available:https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/. [Accessed: 26- Oct- 2020].

1. Background

1.3 Why Entity Reconciliation

· Many knowledge graphs are only for certain specific fields, and iresearch requires knowledge in different fields.

· The proliferation of knowledge graphs (KGs) on the Web is incomplete and inaccurate.

2. Related Works

2.1 Holistic Entity Matching Across Knowledge Graphs(M. Pershina,2016)[5]

(1) Considering the semantic contribution of each word to the whole entity in each entity, the lower the IDF score of a word is, the more entities this word shares.

Figure 1. Similarity calculation formula

$$\langle e_1, e_2, sim(\langle e_1, e_2\rangle) = \frac{1}{||e_1||\cdot||e_2||}\sum_{w\in e_1\cap e_2} idf_1(w)\times idf_2(w)\rangle$$

(2) Set of seed Pairs:

· it selects the set of seed Pairs by computing the similarity of entity attributes based on IDF.

· it uses the connecting entity with the seed pair and adds necessary new entity Pairs and edges to the original seed pair in the Graph to extend the original seed pair.

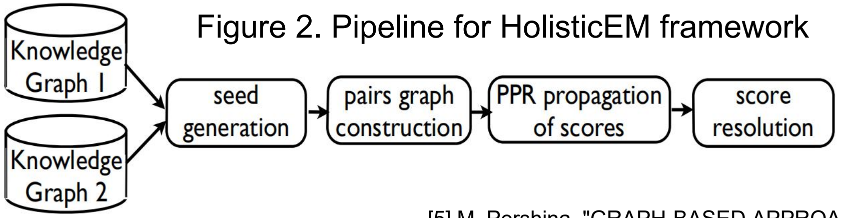


Figure 2. Pipeline for HolisticEM framework

[5] M. Pershina, "GRAPH-BASED APPROACHES TO RESOLVE ENTITY AMBIGUITY", A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Department of Computer Science New York University, 2016.

2. Related Works

2.2 RDF-AI: an Architecture for RDF Datasets Matching, Fusion and Interlink(Yanbin Liu, Chunguang Zhou)[6]

(1) RDF-AI implements a reconciliation framework composed of preprocessing, matching, fusion, interconnection and post-processing modules, and proposes an attribute-based entity pair matching algorithm: fuzzy string matching algorithm based on sequence alignment and word sense similarity algorithm.

Figure 3. Similarity calculation formula

$$Similarity(SW_i, SW_j) = \frac{1}{No(SWi) \times No(SWj)} \times \frac{\sum\limits_{w_i \in \{Wsi\} \cap \{Wsj\}} Ks \times IDF(w_i)^2 + \sum\limits_{w_i \in \{Wci\} \cap \{Wcj\}} Kc \times IDF(w_i)^2 + \sum\limits_{w_i \in \{Wei\} \cap \{Wej\}} Ke \times IDF(w_i)^2}{\sqrt{\sum\limits_{i \in Q_U, K \in \{Ks, Kc, Ke)} K \times IDF(w_i)^2} \times \sqrt{\sum\limits_{j \in Q_v, K \in \{Ks, Kc, Ke)} K \times IDF(w_j)^2}}$$

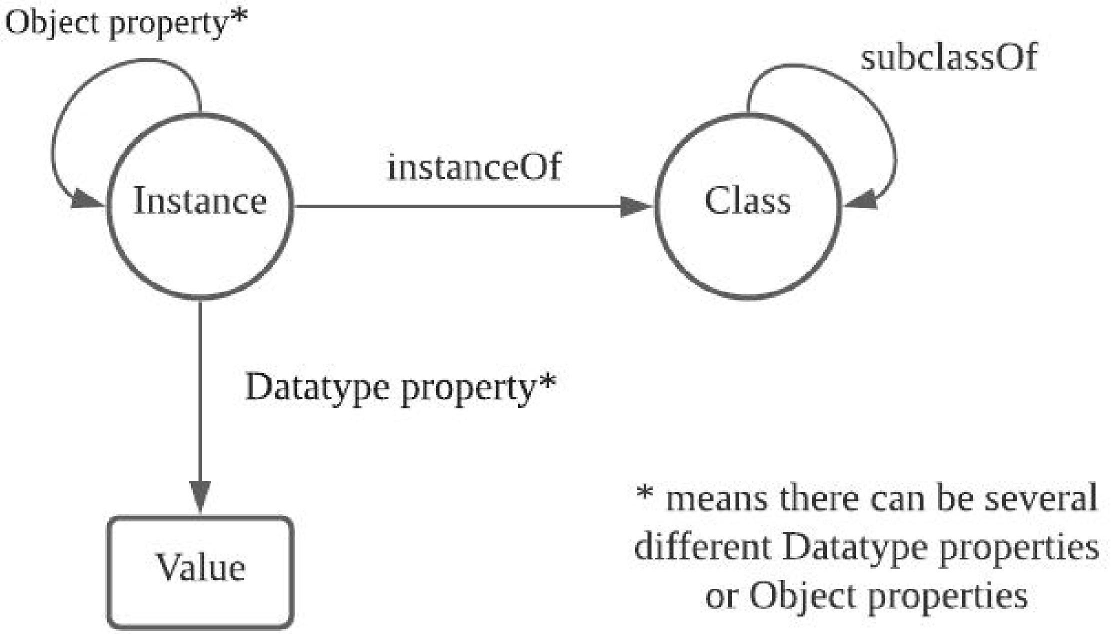
(2) The above method is used to calculate the attribute matching similarity to obtain all possible alignment attribute pairs in the two images, and the entity similarity is obtained by summing the attribute pair similarities. The one with the highest degree of final entity similarity is considered an entity.

[6] F. Scharffe, "Rdf-ai: an architecture for rdf datasets matching, fusion and interlink", Semantic Technology Institute, University of Innsbruck, Austria, 2009.

3. Design and Implementation

3.1 Basic concepts

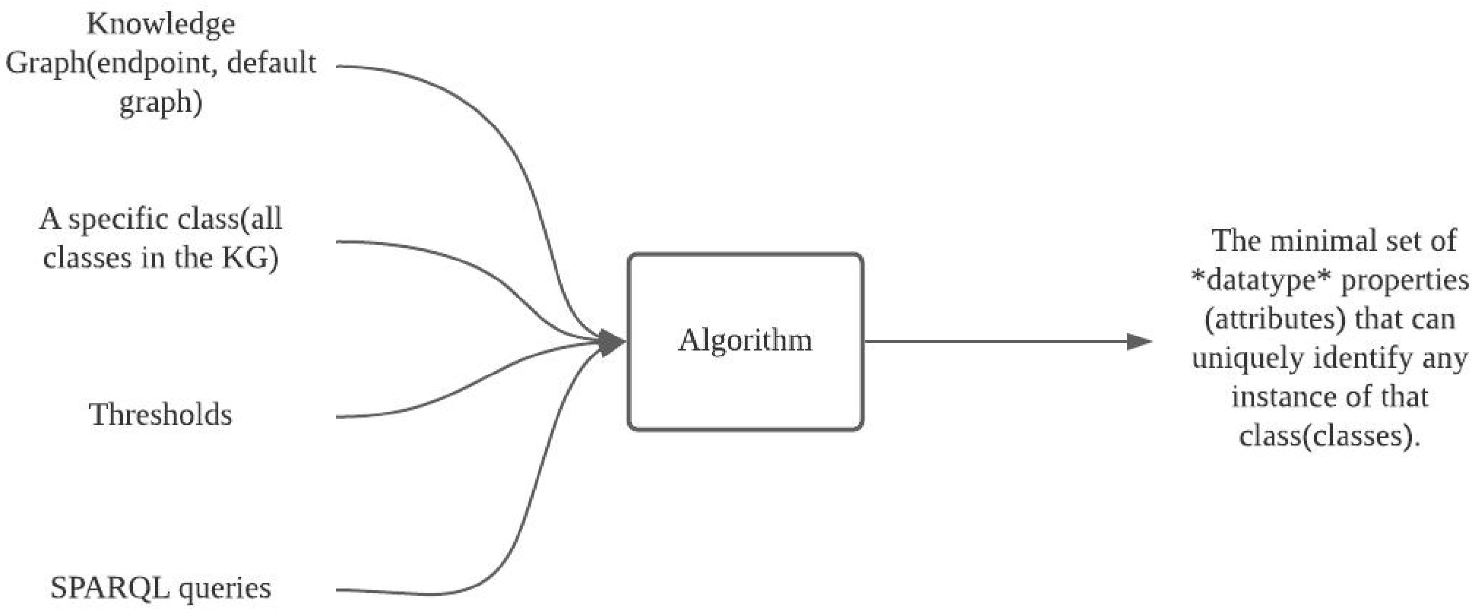
Figure 4. basic concepts


Object property*
subclassOf
instanceOf
Instance
Class
Datatype property*
Value
* means there can be several different Datatype properties or Object properties

3. Design and Implementation

3.2 Design

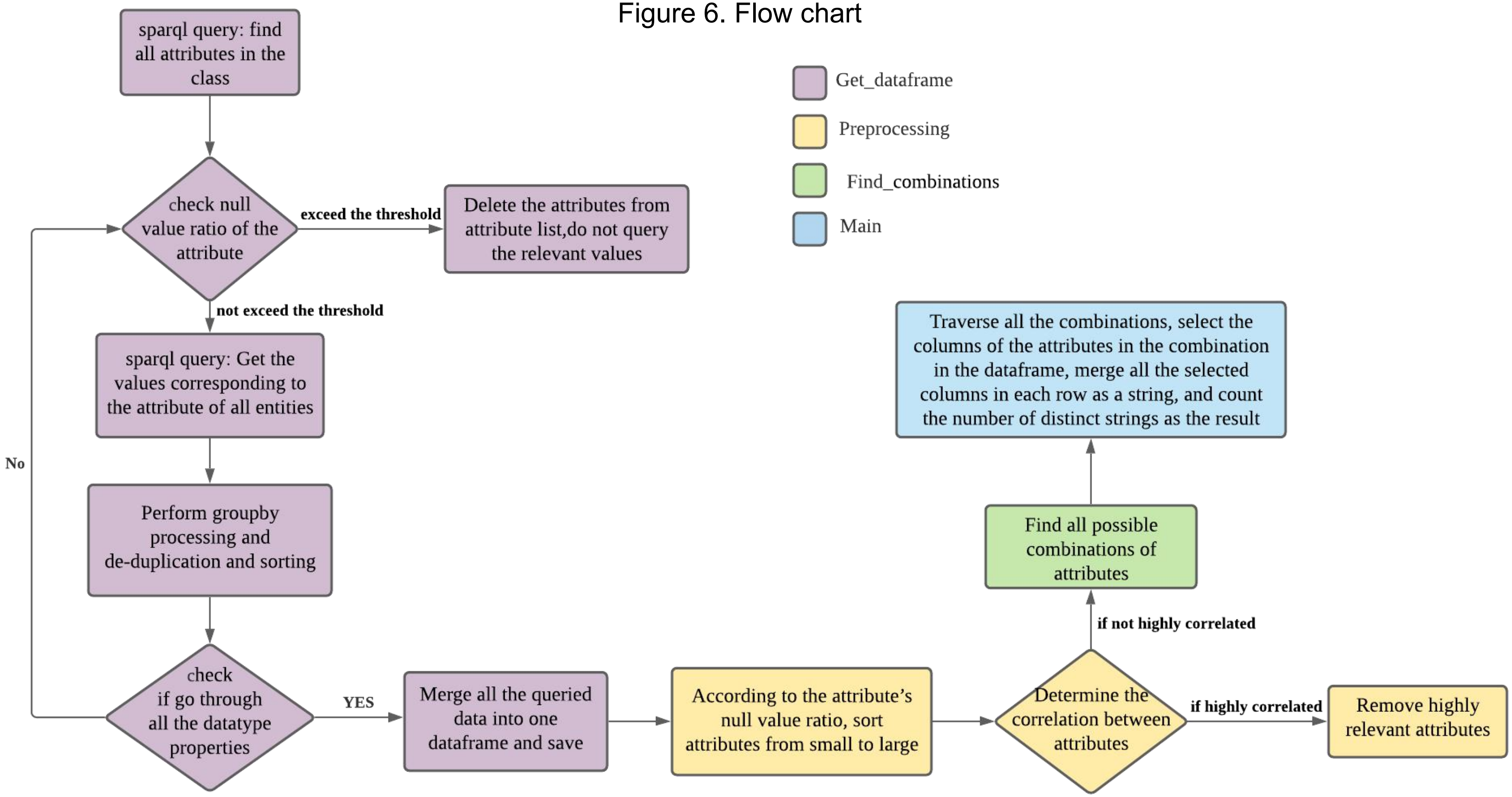
Figure 5. Inputs and output of the algorithm.


Knowledge Graph(endpoint, default graph)
A specific class(all classes in the KG)
Thresholds
SPARQL queries
Algorithm
The minimal set of *datatype* properties (attributes) that can uniquely identify any instance of that class(classes).

3. Design and Implementation

3.2 Design

Figure 6. Flow chart



🟪 Get_dataframe

🟨 Preprocessing

🟩 Find_combinations

🟦 Main

sparql query: find all attributes in the class

check null value ratio of the attribute

exceed the threshold

Delete the attributes from attribute list,do not query the relevant values

not exceed the threshold

sparql query: Get the values corresponding to the attribute of all entities

Perform groupby processing and de-duplication and sorting

check if go through all the datatype properties

No

YES

Merge all the queried data into one dataframe and save

According to the attribute's null value ratio, sort attributes from small to large

Determine the correlation between attributes

if highly correlated

Remove highly relevant attributes

if not highly correlated

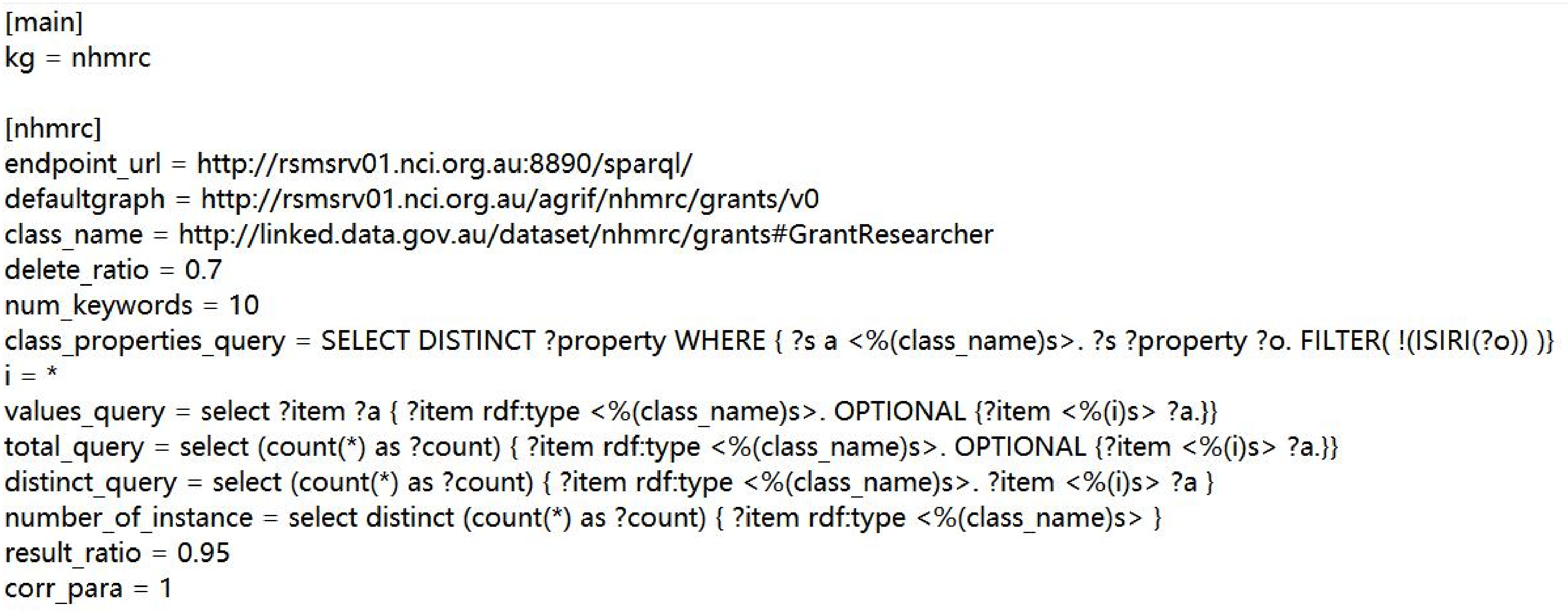
Find all possible combinations of attributes

Traverse all the combinations, select the columns of the attributes in the combination in the dataframe, merge all the selected columns in each row as a string, and count the number of distinct strings as the result

3. Design and Implementation

3.3 Implementation

Figure 7. Config file



```
[main]
kg = nhmrc

[nhmrc]
endpoint_url = http://rsmsrv01.nci.org.au:8890/sparql/
defaultgraph = http://rsmsrv01.nci.org.au/agrif/nhmrc/grants/v0
class_name = http://linked.data.gov.au/dataset/nhmrc/grants#GrantResearcher
delete_ratio = 0.7
num_keywords = 10
class_properties_query = SELECT DISTINCT ?property WHERE { ?s a <%(class_name)s>. ?s ?property ?o. FILTER( !(ISIRI(?o)) )}
i = *
values_query = select ?item ?a { ?item rdf:type <%(class_name)s>. OPTIONAL {?item <%(i)s> ?a.}}
total_query = select (count(*) as ?count) { ?item rdf:type <%(class_name)s>. OPTIONAL {?item <%(i)s> ?a.}}
distinct_query = select (count(*) as ?count) { ?item rdf:type <%(class_name)s>. ?item <%(i)s> ?a }
number_of_instance = select distinct (count(*) as ?count) { ?item rdf:type <%(class_name)s> }
result_ratio = 0.95
corr_para = 1
```

3. Design and Implementation

3.3 Implementation

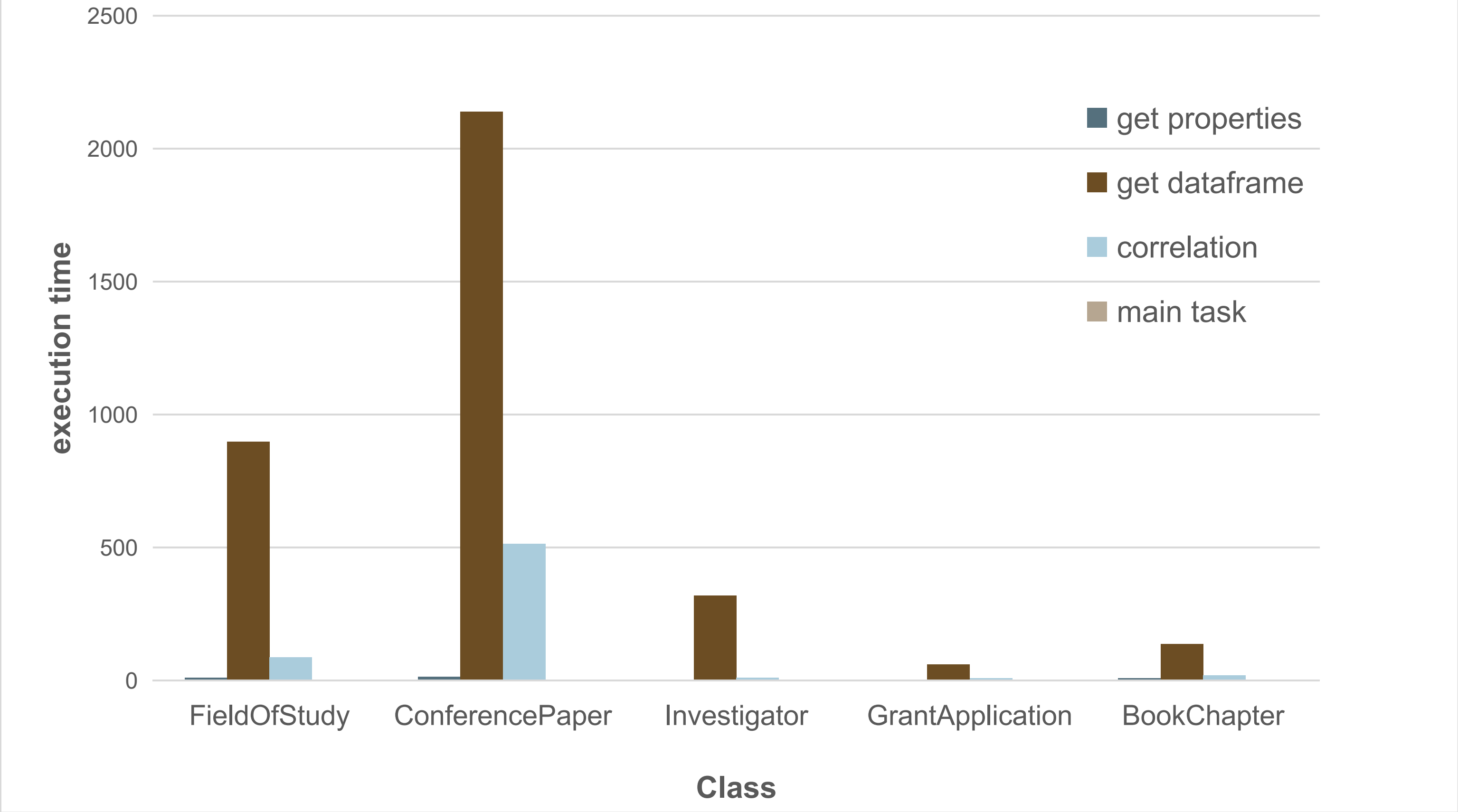
Table 1. Table of time complexity

Time complexity		
Get dataframe	O(mn)	*m: number of properties* *n: number of instances*
Find combinations	O($n*r*C_n^r$)	*r: the length of combination* *n: length of the list*
Correlation	O(n^2)	*n: the length of doc*
Traverse	O(cn)	*c: number of combinations* *n:number of instances*

4. Results

4.1 Test on NHMRC (Unit: second)

Figure 7. Execution time on NHMRC



4. Results

4.1 Test on NHMRC

Table 3. Table of test results on NHMRC

Test Results		
Class Name	**Results**	**Example**
FieldOfStudy	name	*"Modes of convergence"*
ConferencePaper	title	*"CP-Miner: a tool for finding copy-paste and related bugs in operating system code"*
Investigator	fullName	*“A/Pr Ingrid van der Mei”*
GrantApplication	applicationID	*"APP1185426"*
BookChapter	doi	*"10.1007/978-94-007-1333-8_83"*

4. Results

4.2 Test on Wikidata and Dbpedia

Table 4. Table of test results on wikidata and dbpedia

Test Results		
Class Name	**Wikidata Results**	**Dbpedia Results**
films	director, IMDB ID, publication date	IMDB ID, director, runtime
books	author,title	isbn, author, numberOfPages, publisher
animals	taxon name	family, taxonomy, binomialAuthority, genus
country	highest point, GS1 country code	capital, dissolutionDate

5. Discussion

5.1 Advantages

· The algorithm is general and can be applied to most KG, including public KG and custom KG.

· Associated thresholds can be customized as well as SPARQL Query.

5. Discussion

5.2 Limitations

· Algorithm speed still needs to be improved.

· High requirement for network unimpeded.

· Nonsensical attributes cannot be recognized.

6. Future Works

· Optimize the time complexity of the algorithm.

· Determine the value of attributes through Semantic Parsing and Syntactic Parsing.

7. Conclusion

· Prepared for ER

· Learned to use SPARQL Query to get useful information.

· An algorithm based on Python and SPARQL is implemented to find the minimum datatype property set that uniquely identifies a class in KG.

Acknowledgements

- Supervisor:

  Sergio Rodríguez Méndez

  Armin Haller

References

[1] "What is a Knowledge Graph? | Ontotext Fundamentals", Ontotext, 2020. [Online].Available:https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/. [Accessed: 26- Oct- 2020].

[2] https://wiki.dbpedia.org/

[3] https://www.wikidata.org/wiki/Wikidata:Main_Page/

[4] http://ma-graph.org/

[5] M. Pershina, "GRAPH-BASED APPROACHES TO RESOLVE ENTITY AMBIGUITY", A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Department of Computer Science New York University, 2016.

[6] F. Scharffe, "Rdf-ai: an architecture for rdf datasets matching, fusion and interlink", Semantic Technology Institute, University of Innsbruck, Austria, 2009.