Experiences of Using WDumper to Create Topical Subsets from Wikidata

Seyed Amir Hosseini Beghaeiraveri
Alasdair Gray
Fiona McNeill

2nd International Workshop On Knowledge Graph Construction
ESWC - 2021
Outline

• Why Subsets?
• Topical Subsets
• WDumper, Easy and Accurate tool
  • Strengths and Weaknesses
• Research Challenges
• Future Works
Why Subsets?

Huge Size of KGs

Wikidata 2014
3.5 GB

Wikidata 2021
100 GB

Timed-Out Queries
Why Subsets?

Reducing the Overall Costs  Reproducible Experiments
Topical Subsets

Select a set of entities, filtered based on a given topic (e.g. life science, politics, academia, etc.)

+ All their facts

Original Graph  Selecting topic related entities  Final subset
Topical Subsets

Select a set of entities, filtered based on a given topic (e.g. life science, politics, academia, etc.)
+ All their facts
Use Cases

Politicians
- One-type
- No Conditions

General (military)
Politicians
- One-type
- More Conditions

UK Universities
- One-type
- Small Output

Gene Wiki*
- 17 types
- Wikipedia project

<table>
<thead>
<tr>
<th>Method</th>
<th>Comment</th>
<th>Limitation(s) regarding topical subsetting</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2GML (Matsumoto et al. 2018)</td>
<td>conversion of RDF graphs into Property Graphs</td>
<td>No RDF output</td>
</tr>
<tr>
<td>Context Graph (Mimouni et al. 2020)</td>
<td>Captures all adjacent nodes and edges within a given radius</td>
<td>Not practical for the concept of topic</td>
</tr>
<tr>
<td>ShEx</td>
<td>Schema Language for describing RDF graphs</td>
<td>Not Practical for large-scale data</td>
</tr>
</tbody>
</table>
WDumper

• Java-backend + Flask-frontend
• Based on Wikidata Development Kit
• Inputs:
  • A JSON specification file (Filters)
  • A JSON.gz Wikidata complete dump
• Operation:
  • Applying filters on JSON.gz dump
• Output:
  • Custom RDF dump
Evaluation

- Test platform: Two different Wikidata dumps

<table>
<thead>
<tr>
<th>Release Time</th>
<th>Size (GB)</th>
<th>Total Items (milion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>April 2015</td>
<td>~ 4.5</td>
<td>~ 18</td>
</tr>
<tr>
<td>November 2020</td>
<td>~ 90</td>
<td>~ 91</td>
</tr>
</tbody>
</table>
Evaluation

Test Conditions:

- Checking the number of entities that must be on the output → COUNT queries
- Checking the number of statements in each entity → DESCRIBE queries
- Checking the extraction qualifiers and references
Evaluation

Table 4. Results of performing DESCRIBE queries on the selected entity.

<table>
<thead>
<tr>
<th>Use case</th>
<th>Entity</th>
<th>2015 Dump</th>
<th>2020 Dump</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Output</td>
<td>Input</td>
</tr>
<tr>
<td>Politicians</td>
<td>Q23</td>
<td>408</td>
<td>776</td>
</tr>
<tr>
<td>General (military) Politicians</td>
<td>Q355643</td>
<td>104</td>
<td>150</td>
</tr>
<tr>
<td>UK Universities</td>
<td>Q1094046</td>
<td>64</td>
<td>108</td>
</tr>
<tr>
<td>Gene Wiki</td>
<td>Q30555</td>
<td>12</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 6. Number of qualifiers and references for the selected property of the selected entity in the output and input of WDumper (2020 dump).

<table>
<thead>
<tr>
<th>Entity</th>
<th>Property</th>
<th>Qualifiers</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Output</td>
<td>Input</td>
</tr>
<tr>
<td>Q23</td>
<td>P26</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Q355643</td>
<td>P485</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Q1094046</td>
<td>P355</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Q17487737</td>
<td>P680</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>
WDumper Strengths vs. Weaknesses

- Can use any type of wikidata properties as filter (no number limit)
- Can extract all desired entities, their statements and qualifiers/references

- Can not use property paths like P31/P279 and path extensions like P279*
- Can not detect Type Hierarchy
- Can not have connections between filters
Research Challenges

- Huge size of 2021 dump
- Syntax errors in Turtle dumps
  bad chars in JSON dumps
- Different implementations of
  DESCRIBE in Jena vs. WDQS
Future Works

• Adding Type Hierarchy Detection to WDumper
  An initial try:
  https://github.com/seyed1411/wdumper/blob/master/extensions/add_subclasses.py

• Combining WDumper Core with ShEX/SPARQL for better flexibility

• How to build live subsets?

• How about topically subsetting RDF KGs?
Summary

• Subsets are useful
  • Cost reduction + Ease of access + Flexibility

• Topical subsets: Set of entities + their facts around a given topic

• WDumper is a reliable tool for extracting (some) Topical Subsets
  + Good for topics with simple type structure
  - Only applicable on Wikidata
  - Not flexible