Wikidata: Alternate Reference Model(s)

Arthur Smith
@ArthurPSmith

This session is recorded: Please mute your microphone and camera when you're not speaking.

DATA MODELLING DAYS
November 30
December 2
2023
WIKIDATA

References in Wikidata

Every statement has (independent) references

- This is good - we want our data to be verifiable
- But it is cumbersome to enter the same reference repeatedly
  - tools to help: DuplicateReferences gadget, UseAsRef script
- Size of duplicated references is a problem in item storage, hitting size limits
  - See talk by @Mahir256 “How to make Wikidata smaller” at WikidataCon 2023 video

Example of the size problem - Q21481859

- Almost 3000 authors (P50 or P2093 statements)
  - Potentially all with same reference to ADS - however bot adding these references ran out of room on the item so most P50/P2093 statements missing reference
- Also over 50 “cites work” statements
  - all with same reference to Crossref
- Item is barely editable due to size; merging with another item if it was a duplicate would be impossible.
- RDF format (in WDQS) does not have this problem: combines duplicate references into a single collection of triples, with statements connected via “prov:wasDerivedFrom”.

Possible solutions

1. Just remove redundant references on an item
    a. Saves the most space...
    b. Similar to most wikipedias where references rarely cited more than once in the text.
    c. But - it really is useful to indicate each statement is supported by a source!
2. Change the JSON storage format
    a. This could do something similar to what the RDF does; duplicate references consolidated automatically
    b. Requires developer attention
3. New properties and usage patterns for references
    a. “reference” & “see reference” properties
    b. Alternatives?

On test.wikidata.org - Q232624 example reference:


```
"references": [ { "hash": "51ae109329c13aebb6e83e53e1583cf93312f9e6",
            "snaks": {
              "P149": [{
                 "snaktype": "value",
                 "property": "P149",
                 "hash": "c9e5b4c130d77a5b3378e011baa979d6fa350334",
                 "datavalue": {
                   "value": { "entity-type": "item", "numeric-id": 213518, "id": "Q213518"}
                   "type": "wikibase-entityid"
                 }, "datatype": "wikibase-item" }],
              "P388": [ {
                 "snaktype": "value",
                 "property": "P388",
                 "hash": "f7414f22847b1d03fe9c165f44698d35a046a515",
                 "datavalue": {
                   "value": {"time": "+2023-11-24T00:00:00Z", "timezone": 0, "before": 0, "after": 0, "precision": 11,"calendarmodel":
"http://www.wikidata.org/entity/Q1985727"},
                   "type": "time"
                 }, "datatype": "time"}]
            }, "snaks-order": [ "P149", "P388"]}]
```


Identical reference repeated 4 times, same hash value
“51ae109329c13aebb6e83e53e1583cf93312f9e6”
(also pointed out by @Mahir256 at WikidataCon 2023)

DATA MODELLING DAYS

Solution 2: Alternate JSON storage format

On each referenced statement:

```
“references”: [{“ref_hash”: “51ae109329c13aebb6e83e53e1583cf93312f9e6”}]
```

And then on the item the full references (but only once per hash value):

```
"references": [ { "hash": "51ae109329c13aebb6e83e53e1583cf93312f9e6",
          "snaks": {
            "P149": [{ … }], …}, …]
```

For this example the ref_hash statement is 85 bytes, full ref is 631, so data size change for N copies of this reference is:

631 + N * (85 - 631) or 1553 bytes smaller for N = 4.

If N = 3000, saving is over 1.5 MB

Implications of alternate JSON storage

- Wikidata UI would need to change:
  - How should references on a statement be displayed?
  - Should the full list of references be displayed separately on the item?
  - How should editing references work - allow a change to all usages of a reference at once, or only one by one?
- Would APIs need to change?
- Mapping to RDF format should be simpler as this more closely matches the RDF (change would still be needed)
- Both storage formats would probably need to be supported simultaneously, at least for a transition period
- Other impacts?

Solution 3: New properties and usage patterns

See Q232625 on test.wikidata

- New reference property “see reference” with string value (reference label)
- New main statement property “reference” with string value as label, and rest of reference as qualifiers

Not as efficient as reworking JSON: “see reference” size for this example is 266 bytes, main “reference” statement 838 bytes.
Size change for N refs is:

838 + N * (266 - 635) or 638 bytes saved for N = 4.

For N = 3000 this is still over 1 MB smaller.

Implications of new reference properties

- No UI or underlying code changes necessary
  - Could UI group the “reference” statements at the bottom of the page?
  - Also link from “see reference” label to corresponding reference would be nice (the way wikipedia does it).
- Fetching references via API/RDF becomes more complex:
  - If reference is a “see reference” then “reference” statements need to be fetched for the item, matched to label.
- Reference properties as qualifiers on a reference statement leads to constraint violations - can we fix?
  - Or just add them as reference entries instead of qualifiers?
- Other impacts?

Summary

Redundant wikidata references need attention

- Solution 1: remove redundant references
- Solution 2: change underlying JSON format
- Solution 3: new “reference” and “see reference” properties
- Next steps:
  - Project chat?
  - RFC?
  - Property proposals?
  - Other ideas?