Contextualization via Qualifiers

Peter F. Patel-Schneider[*]

Nuance AI and Language Lab
pfpschneider@gmail.com

Abstract. A common method for contextualizing facts in knowledge graph formalisms is by adding property-value pairs, qualifiers, to the facts in the knowledge graph. Qualifiers work well for information that is additional to the base fact but pose an unwarranted burden on consumers of the information in knowledge graphs when the qualifier instead contextualizes the base fact, as in limiting the applicability of the fact to some time period or providing a confidence level for the fact. Contextualization should instead by done in a more principled manner and accompanied by tools that lessen the burden on consumers of knowledge graphs.

A common method for contextualizing facts, particularly the subject-predicate-object facts in knowledge graphs, is to add extra pieces of information to the data structure that carries these facts, perhaps after modifying the data structure to include a place for these extra pieces.[1] In Wikidata[2] there are qualifiers, adding extra information directly to the statements that represent facts. In schema.org[3] there are roles, which interpose an extra item between the subject and object to which additional information about the relationship can then be added. Singleton properties is an extension to RDF that adds special property nodes that can then hold additional information about the relationship they represent. RDF itself has reification, where a statement node represents a triple and can be used as the subject of additional information about that triple. In each case the additional information can be considered to *qualify* the base fact.

In each of these methods the qualifying information is, in principle, unrestrained, representing anything from additional arguments to the relationship (such as the role played by a cast member or the administrative district of a government position) to location information (such as the location in which a fact holds) to temporal information (particularly the temporal period in which the fact holds). Qualifiers are simple and effective for some of these purposes, but have distinct problems for others.

[*] This work will be published as part of the book *Emerging Topics in Semantic Technologies. ISWC 2018 Satellite Events*. E. Demidova, A.J. Zaveri, E. Simperl (Eds.), ISBN: 978-3-89838-736-1, 2018, AKA Verlag Berlin.

[1] There are other methods for contextualizing facts, such as the use of named RDF graphs [1]. I generally concentrate here on methods that directly add to the fact data structure and are widely known.

[2] https://wikidata.org

[3] https://schema.org

1 Additive Qualifiers

Consider first using qualifiers to represent the role played by a cast member of a film. This information is somewhat awkward to represent in triple-based formalisms because it is a three-place relationship. So Wikidata and schema.org use qualifiers to attach the role played to the base information that an actor is in the cast of a film.

RDF reification and singleton properties can use qualifiers for this as well. In RDF reification [2, 4], the reified cast membership triple is represented as four triples, using built-in RDF vocabulary, as in

```
_:rt rdf:type rdf:Statement .
_:rt rdf:subject :GWTW .
_:rt rdf:predicate :castMember .
_:rt rdf:object :ClarkGable .
```

where _:rt is a statement node that represents the triple

```
:GWTW :castMember :ClarkGable .
```

The statement node can then have the other information associated with it, as in

```
_:rt :characterRole :RhettButler .
```

It is important to note that the reification of a triple in RDF does not entail the triple, i.e., it is possible to have the reification of a triple in RDF without implying that the triple itself is true so there is no implication from the above that Gone with the Wind has Clark Gable as a cast member, nor that he plays Rhett Butler in the movie.

Wikidata [6] is built around items and property-value statements about them, thus forming in essence subject-predicate-object facts. Each statement can have associated qualifiers,[4] which are predicate-object pairs. So Wikidata should contain the following information[5]

```
Gone with the Wind
  instance of: film
  cast member: Clark Gable
    character role: Rhett Butler
```

The intended meaning of this is that Gable is a cast member of Gone with the Wind and that the character role of this cast membership is Rhett Butler. Here the underlying "triple" is indeed a fact, as opposed to the situation with RDF reification, but, as Wikidata does not have a formal semantics, there is no formal statement of this intended meaning. As qualifiers are placed directly on regular statements any tools that use Wikidata information don't need any special code to handle the statements that have attached qualifiers.

Singleton properties [5] is an extension to RDF that creates a new kind of property, one that can only have one triple in its extension. This singleton property can then

[4] https://www.wikidata.org/wiki/Help:Qualifiers

[5] Descriptive names are used here in Wikidata information structures—as opposed to the actual opaque numeric identifiers—so that the meaning of the structures is easier to see.

be used as the subject of other triples that are considered not to be about the singleton property itself but instead are about its single triple. So in singleton properties one might say

```
:castMember_1 rdf:singletonPropertyOf :castMember
:GWTW :castMember_1 :ClarkGable .
:castMember_1 :characterRole :RhettButler
```

This looks like a syntactic abbreviation for RDF reification but it has one important difference—there is an entailment in singleton properties from the singleton property to its regular property. That is, in the singleton properties semantics the above triples entail

```
:GWTW :castMember :ClarkGable .
```

Because singleton properties uses a special data structure (the singleton property) a tool for singleton properties needs to have special code that makes the inference from triples with singleton properties to the triple with the correct regular property.

schema.org contains a special class `Role` (`https://schema.org/Role`) that is used for additional information about a relationship.[6] Instead of a property directly linking from a subject to an object, the property links from the subject to a role item and then again from the role item to the object, as in (again recast slightly)

```
:GWTW :castMember _:r .
_:r rdf:type schema:Role .
_:r :castMember :ClarkGable .
_:r :characterName :RhettButler .
```

The intended meaning of this is again that Gone with the Wind has Clark Gable as a cast member, and that the character role of this cast membership is Rhett Butler. (Indeed schema.org has a specialization of Role, PerformanceRole, that is designed specifically for this particular kind of extra information.) So here again the underlying "triple" is indeed a fact, as opposed to the situation with RDF reification. As schema.org does not have a formal semantics, there is no formal statement of this intended meaning. Any tool for schema.org data needs to handle Role items specially, short-circuiting the double link to recover the underlying normal meaning

From this example it appears that each of Wikidata, singleton properties, and schema.org have qualifiers working very nicely. They all permit additional information to be associated with their analogue of triples while retaining the underlying truth of the triple. The special processing required to recover this underlying truth is not onerous. RDF reification appears the loser here as it does not directly retain the underlying truth while associating extra information with a triple. If the triple itself is supposed to be true then it has to be asserted independently, and connecting the additional information to this triple also has to be done.

[6] `http://blog.schema.org/2014/06/introducing-role.html`

2 Contextual Qualifiers

In the above example the qualifer is adding extra information about the fact, namely that the cast member is playing a particular role. This extra information does not interfere in any way with the cast membership, so tools (and queries) that are interested in cast membership can safely ignore the qualifier so long as they handle the modifications to the underlying data structure to handle qualifiers. This is true even in RDF, so long as the base triple is also present.

However, many, perhaps most, uses of qualifiers in Wikidata[7] and schema.org contextualize the underlying fact, i.e., they limit the contexts in which the underlying fact is true. One of the most important contextualizations is temporal contextualization, which is generally handled in these schemes via a start and end qualifier.

So in singleton properties representing the temporal aspects of Bob Dylan's marital information would be (adapted from [5]):

```
:isMarriedTo#1 rdf:singletonPropertyOf :isMarriedTo .
:BobDylan :isMarriedTo#1 :SaraLownds .
:isMarriedTo#1 :hasStart "1965-11-22"^^xsd:date .
:isMarriedTo#1 :hasEnd "1977-06-29"^^xsd:date .
:isMarriedTo#2 rdf:singletonPropertyOf .
:BobDylan :isMarriedTo#2 :CarolDennis .
:isMarriedTo#2 :hasStart "1986-06-04"^^xsd:date .
:isMarriedTo#2 :hasEnd "1992-10-01"^^xsd:date .
```

In Wikidata this is

```
Bob Dylan
  spouse: Sara Dylan
    end time: 29 June 1977
    start time: 22 November 1965
  spouse: Carolyn Dennis
    end time: October 1992
    start time: 4 June 1986
```

In singleton properties there is the entailment from the above triples to

```
:BobDylan :isMarriedTo :SaraLownds .
:BobDylan :isMarriedTo :CarolDennis .
```

i.e., that Bob Dylan is a bigamist. In Wikidata there is also the implied intent that there are two spouses of Bob Dylan, but although this is somewhat misleading there is perhaps not quite the same force to the conclusion of bigamy because of the name of the

[7] Of the ten most-used qualifier properties in Wikidata as of July 2018. (See https://tools.wmflabs.org/sqid/#/browse?type=properties.) four provide temporal context (point in time, start time, valid in period, and end time), one provides spatial context (chromosome), one can be considered to provide a certainty context (determination method), three are ”additive” (taxon author, points scored, and matches played), and one provides no information content (stated as).

relationship. As there is no formal semantics for Wikidata there is no formal support for drawing both conclusions. The qualifier methodology of schema.org would act the same as Wikidata.

RDF reification, on the other hand, can represent this information as in

```
:rt1 rdf:subject :BobDylan .
:rt1 rdf:predicate :isMarriedTo .
:rt1 rdf:object :SaraLownds .
:rt1 :hasStart "1965-11-22"^^xsd:date .
:rt1 :hasEnd "1977-06-29"^^xsd:date .
:rt1 rdf:subject :BobDylan .
:rt1 rdf:predicate :isMarriedTo .
:rt1 rdf:object :CarolDennis .
:rt1 :hasStart "1986-06-04"^^xsd:date .
:rt1 :hasEnd "1992-10-01"^^xsd:date .
```

which does not imply that Bob Dylan is a bigamist.

3 The Problem with Contextual Qualifiers

The underlying problem is that contextual information does not add to the base information but instead modifies the it, stating in which context (temporal or otherwise) the information is true or providing information as to how likely the base information is to be true. Consumers of the information need to always be aware of whether the context that they are (perhaps implicitly) working in is not one to which the qualifiers attached to a particular piece of information apply.

This might not be so hard if the only contextual information is temporal and that information is carried only by start and end dates. Tools (and queries against the underlying data) working in a particular time point can explicitly exclude facts with temporal qualifiers that do not cover that time point. Tools (and queries) working in an implicit now can exclude any fact with an end time qualifier (assuming of course that no end date is in the future, which could be a requirement for temporal qualifiers).

However, each and every contextual qualifier has to be considered by every tool (and query against the underlying data). So there can be multiple exclusions, such as for contextual location, confidence, and certainty, and tools (and queries) will have to be updated whenever new such qualifiers are added. Wikidata has over one hundred and fifty qualifiers so to determine whether a fact is true in a context in Wikidata each and every one has to be examined to see if it is a contextual qualifier and for those which are their interaction with the current context or the implicit context has to be determined. This is a high bar indeed, made even worse as new qualifiers are added on an ongoing basis.

4 Solutions for Contextual Qualifiers

The right solution is to not show contextual qualifiers to consumers. Ideally, instead replace contextual qualifiers with a formal theory of the contexts so that basic tools (contextualizers, reasoners) can be written that correctly take context into account. Alternatively, create low-level tools that remove facts that are not valid in the contexts that a consumer wants to use.

The formal approach requires building logics and theories for the contextual constructs. Building and implementing these formal theories is certainly not a trivial task—handling temporal contexts, for example, requires a temporal logic[8] —but the result will provide firm underpinnings of the constructs, avoiding problems with differing views of their meaning by different consumers of the information. Making the complexities inherent in contexts part of the base capabilities of knowledge graphs also means that more of the complexities are handled by the producers of contextualized knowledge graphs and fewer are inflicted on the consumers of information from knowledge graphs.

The non-formal approach requires less initial effort. For example, instead of being given full temporal reasoning tools, consumers would be given an interface that produces only the information valid in a particular context or a particular range of contexts. This is somewhat similar to RDF extracts produced from Wikidata [3] and the truthy statements in the Wikdata RDF dumps described in `https://www.wikidata.org/wiki/Wikidata:Database_download/en` although these extracts are not based on an interesting theory of contextual qualifers and do not fully distinguish between qualifiers that provide extra information and those that provide contextualization. This interface, however, does not allow consumers to work in multiple contexts (or ranges of contexts) at one, limiting the kind of work they can do.

It is quite reasonable to combine the two approaches, providing extraction tools that are based on a formal theory of the contexts. Consumers that only need simple contextual processing can use the extraction tools; consumers that need more complex contextual processing can use the full reasoning tools.

References

1. Jeremy Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs. *Journal of Web Semantics*, 3(4):247–267, 2005.
2. Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 concepts and abstract syntax. W3C Recommendation, `http://www.w3.org/TR/rdf11-concepts`, 25 February 2014.
3. Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing wikidata to the linked data web. In *Proceedings of the Thirteenth International Semantic Web Conference*, pages 50–65, 2014.
4. Patrick Hayes and Peter F. Patel-Schneider. RDF 1.1 semantics. W3C Recommendation, `http://www.w3.org/TR/rdf11-mt/`, 25 February 2014.
5. Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like RDF reification? making statements about statements using singleton property. In *Proceedings of the 23rd international conference on the World Wide Web (WWW 2014)*, 2014.
6. Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[8] If all time points are constants on a single time line, the required temporal reasoning might be simple, but still a temporal logic should be used to underpin the temporal reasoning algorithm.