Handling Uncertainty in RDF Graphs for numismatic Use Cases

Data Science research project by
Jens Weigel
Submitted to
Karsten Tolle

Goethe University Frankfurt

January 2, 2024

Contents

1 Introduction

With the World Wide Web we have access to a seemingly unlimited amount of data. Data is referred to as the new gold or to be even more valuable because you can reuse the same data again [3]. The problem is that the data is often unstructured, difficult to merge when having multiple sources and it is hard to do more complex queries instead of doing text searches. The Resource Description Framework (RDF) tackles those issues by providing a framework to structure and link data on the internet. It is designed represent information in a machine understandable way so searching queries with a semantic meaning is now possible.

Uncertainty in RDF data can have many reasons such as imprecise measurements, incomplete knowledge or conflicting information. When uncertainty is not represented or handled accordingly while working with the data these uncertainties can negatively impact the quality of data, the results of search queries and the trustworthiness of semantic web applications. Currently there is no clear standard how to model uncertainty (section 2) in RDF or how to handle them during reasoning (section 3). This paper shows multiple approaches for reasoning in uncertain RDF graphs and tests if they can be used for ancient coinage use cases.

2 Motivation

When dealing with archaeological artefacts like ancient coinage there are multiple sources for uncertainty [15]. There are corroded coins like figure 1 that experts have trouble classifying. Even if a coin is classified with high confidence to a coin type those coin types could also have vaguely defined time periods for the mint date which adds another layer of uncertainty. There is also the case that the researchers have different opinions when classifying the coin that adds inconsistencies to the graph. The W3C classifies uncertainty in Ambiguity, Empirical, Vagueness, Inconsistency and Incompleteness [17]. Using the W3C uncertainty types those uncertainties when dealing with ancient coinage can be categorized as vagueness (vaguely defined time periods), incompleteness (coins can't be classified) and inconsistency (conflicting classification from experts). To tackle this there is research how to model uncertainties in RDF ontologies [15, 11, 9] and how to reason in uncertain RDF graphs [2, 6, 7, 16]. In this paper we will define several use cases, model them as RDF graphs and determine whether the current approaches are suitable for ancient coinage.

Figure 1: Coin with corrosion. From [15].

Figure 2: Example for similar coin types from [8]. `ric.1(2).aug.4A` on the top and `ric.1(2).aug.5` on the bottom. The legend on the top left is "IMP CAESAR AVGVST" and on the bottom left "IMP CAESAR AVGVSTV".

2.1 Use Case 1: Coin Type and Coin Similarity

In this use case the goal is to find similar coins or coin types like the example shown in figure 2. This could be done by domain experts that look at the coin types and define a similarity value between 0 and 1 or using machine learning models. Computer vision algorithms could be used to cluster the coin images or NLP could help by parsing the coin descriptions to RDF statements which are then used for reasoning [14]. There could be multiple models that are used to calculate coin type similarity.

For example, there is a coin that is most likely coin type `ric.1(2).aug.4A` but you only know it with a certainty level of 0.8. Another coin is classified as `ric.1(2).aug.5` with a confidence of 0.7. Coin type `ric.1(2).aug.4A` looks similar to coin type `ric.1(2).aug.5` with a factor 0.9. How likely is it for the coin to be coin type `ric.1(2).aug.5` instead of `ric.1(2).aug.4A`, how similar are those two coins and how to deal with multiple similarity values between coin types when reasoning in the RDF graph? Another question is how to deal with corroded coins if you can't classify them to a single coin type but to a group of similar types.

2.2 Use Case 2: Issuer/Depicted Person

Due to corrosion the issuer or depicted person on a coin can be hard to determine. Thats why the AFE-Web tool [1] allows alternative issuers and depicted person (`Issuing for` in AFE) which is shown in figure 3. Additionally, there is check box that states whether the person adding the coin is sure that the right issuer or depicted person is selected or at least in the given alternatives. From domain experts AFE has a list of issuers with depicted persons they had on their coins. The question is how to deal with the uncertainty regarding alternatives, the uncertain check box and issuers that don't match the depicted persons from the domain knowledge.

2.3 Use Case 3: Closing date for a Hoard

Determining the closing date for a coin hoard can difficult when dealing with multiple coin types and coins that are hard to classify. For example, one coin is poorly preserved and therefore hard to classify. The

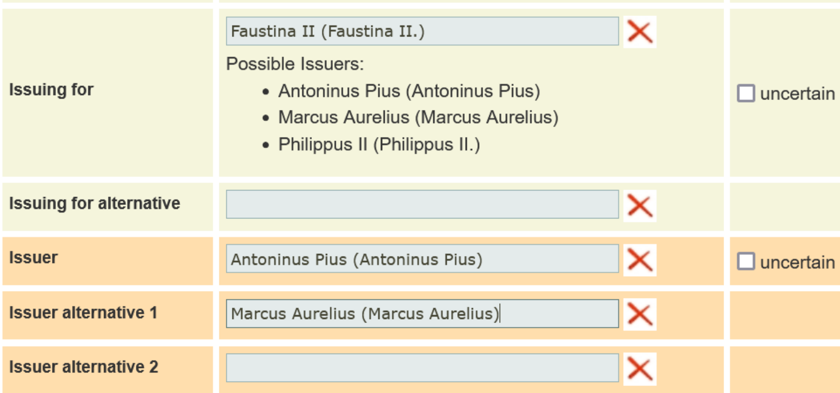

Figure 3: Uncertainty representation in AFE.

expert classifies it as `cn type 21047`[1] with a confidence of 0.6. The second coin, on the other hand, is classified as `cn type 21142`[2] with a high confidence around 1, but the date range for this coin type is vague. The last coin is also corroded and thus only classified with a low confidence of 0.55 to `cn type 20885`[3] but also has a vague time range. The question of this use case is how to deal with the dates from poorly preserved coins and vaguely dated coin types like you can see in table 1 to determine the closing date of the find.

Coin Type	**Date**	**Confidence**
cn type 20885	c. 85-70 BC	0.55
cn type 21047	73-72 BC	0.6
cn type 21142	c. 100-30 BC	1

Table 1: Coin types from an example coin find with 3 coins.

3 Literature

In this section we look at RDF reasoning approaches from different domains to find suitable solution for the numismatic use cases:

1. Reasoning in context-aware Systems
2. Query-Time Reasoning using Soft and Hard Rules
3. Reasoning in Fuzzy RDF Graphs
4. Academic Meta Tool

[1] `https://www.corpus-nummorum.eu/types/21047`
[2] `https://www.corpus-nummorum.eu/types/21142`
[3] `https://www.corpus-nummorum.eu/types/20885`

3.1 Reasoning in context-aware Systems

The article [2] demonstrates an approach to handle uncertainty in context-aware systems which the authors use to detect persons and their activities in assisted living facilities. The authors propose a procedure to measure uncertainty from the sensor level based on hardware characteristics. They extend the semantic modelling approach to include this uncertainty which allows the expression of confidence levels in events received from sensors. This uncertain information is then used to derive contextual information.
Figure 4 shows the ontology from the paper with uncertainty. The authors use "sensor A6 hasCurrentState

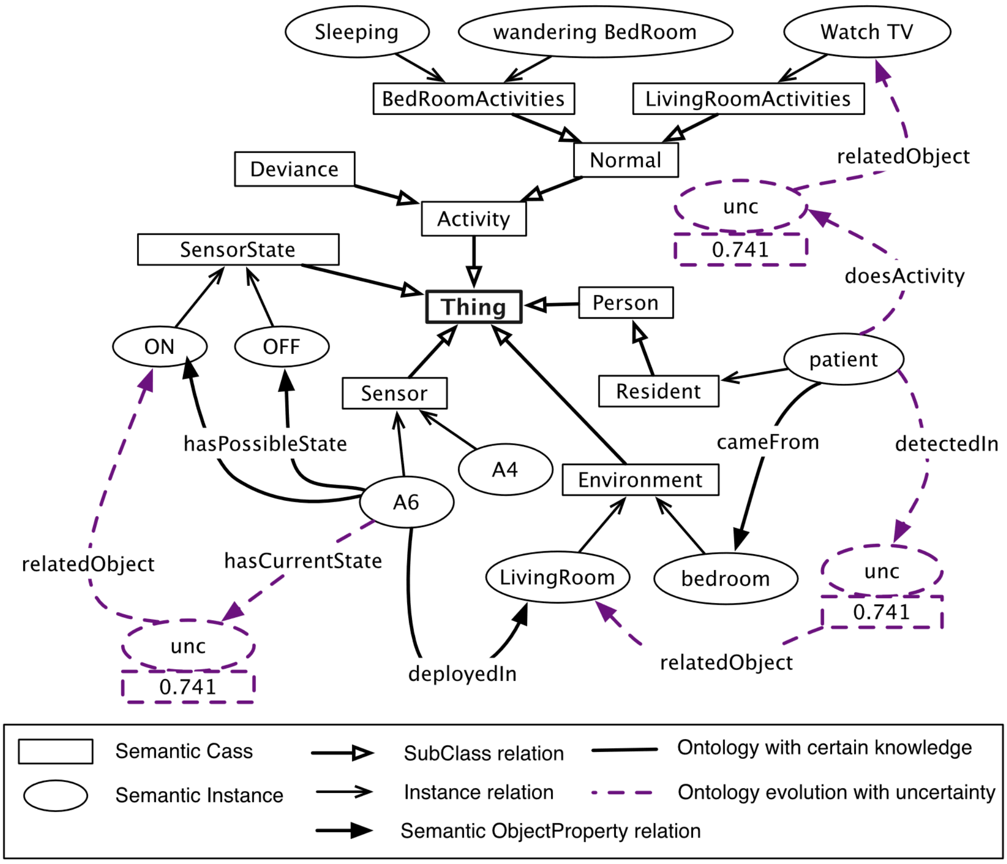

Figure 4: Ontology with uncertainty. From [2].

ON with a certainty level of 74.1%" to infer the derived information "patient detectedIn LivingRoom with a certainty level of 74.1%" and "patient doesActivity WatchTV with a certainty level of 74.1%". Conflicts can happen due to malfunction from the sensors where sensors from different rooms also trigger. Therefore, authors also present a decision-making mechanism to manage conflicts among the sensors. The authors choose Dempster-Shafer theory (DST) [12] as the decision-making method in their approach. They explain that the theory allows for the incorporation of uncertainty from different sources and provides better performance than previous approaches like Bayesian Inference for reasoning with uncertainty [18].
They use data from Monte Carlo simulations and real deployments to validate their uncertainty management approach. The results show that the performance of the system in activity detection is influenced by the choice of hardware characteristics used to identify faulty sensors and the number of faulty sensors. A good selection of characteristics lowers the confidence granted to faulty sensors and improves the

performance. Additionally, are high-coupled sensors (sensor fires $\rightarrow$ one activity detected) better than low-coupled (sensor fires $\rightarrow$ many activities detected) for exact activity detection.
Overall, the authors show the importance of quantifying uncertainty in context-aware systems. Their approach improves context-aware reasoning through methods for resolving conflicts and lowering confidence in faulty hardware. They emphasize the need for careful sensor selection (low and high coupled sensors) and selection of hardware characteristics to detect faulty sensors.

3.2 Query-Time Reasoning using Soft and Hard Rules

The article [7] introduces another way of reasoning with uncertain RDF data through a proposed framework, URDF. The framework combines soft rules (deduction rules) and hard rules (consistency constraints) to handle uncertain RDF data. Inconsistencies are dynamically resolved at query-time using a modified approximative weighted MaxSAT algorithm from [5].
The soft rules are weighted Horn clauses that will be maximized by the modified weighted MaxSAT algorithm. An example soft rule could be "if two persons have the same address, they know each other with a confidence of 0.6". Hard rules on the other hand have no weight but are constraints that need to be satisfied. An example hard rule would be that no person could have two birth dates.
The experiments were conducted with handcrafted, synthetic or inductively learned soft and hard rules to confirm that URDF is robust and significantly improves runtime compared to other state-of-the-art techniques.
Overall, the article shows that URDF is an effective approach for query-time reasoning over uncertain RDF data. The framework improves runtime and provides query answers with an approximation guarantee of up to 0.83, which means their MaxSAT approximation gets at least 83% of the weight of the optimal solution, depending on the use case.

3.3 Reasoning in Fuzzy RDF Graphs

In [6] the authors describe a way to reason in uncertain RDF graphs using the Ontology Web Language (OWL). The OWL extends the RDFS semantics by adding a full set of entailment rules which can be computed in polynomial time. Fuzzy RDF is introduced to model vagueness by giving the fuzzy RDF triples a degree of uncertainty $0 < x \leq 1$. The paper extends the pD* semantics [13] of OWL to allow vagueness in the graph using fuzzy logic [4].

$$\begin{array}{ll}
I(\phi \wedge \psi) = I(\phi) \otimes I(\psi) & I(\phi \vee \psi) = I(\phi) \oplus I(\psi) \\
I(\phi \rightarrow \psi) = I(\phi) \Rightarrow I(\psi) & I(\neg\phi) = \ominus I(\phi) \\
I(\exists x.\phi(x)) = \sup_{c \in \Delta^I} I(\phi(c)) & I(\forall x.\psi(x)) = \inf_{c \in \Delta^I} I(\phi(c))
\end{array}$$

Figure 5: Fuzzy Interpretation Extension. From [6].

Fuzzy logic in figure 5 allows statements that are not strictly true or false but can have fuzzy degree of n attached to them which means the degree of truth is at least n. $\oplus$, $\otimes$ and $\ominus$ are combination functions like triangular norms.

A fuzzy RDF triple is in the form of $(s, p, o)[n]$ where (s, p, o) is a normal subject-predicate-object triple and n denotes the degree of uncertainty. The paper provides a set of entailment rules derived from the fuzzy logic and the pD* semantics. A part of those is seen in figure 6 where $\otimes$ is a combination norm

	Condition	Constraint	Conclusion
f-rdf1	$(v, p, w)[n]$		$(p, \texttt{type}, \texttt{Property})[1]$
f-rdfs2	$(p, \texttt{domain}, u)[n]$ $(v, p, w)[m]$		$(v, \texttt{type}, u)[n \otimes m]$
f-rdfp4	$(p, \texttt{type}, \texttt{TransitiveProperty})[n]$		
	$(u, p, v)[m]$ $(v, p, w)[l]$		$(u, p, w)[n \otimes m \otimes l]$
f-rdfp5a	$(v, p, w)[n]$		$(v, \texttt{sameAs}, v)[1]$
f-rdfp5b	$(v, p, w)[n]$		$(w, \texttt{sameAs}, w)[1]$
f-rdfp6	$(v, \texttt{sameAs}, w)[n]$	$w \in U \cup B$	$(w, \texttt{sameAs}, v)[n]$
f-rdfp7	$(u, \texttt{sameAs}, v)[n]$ $(v, \texttt{sameAs}, w)[m]$		$(u, \texttt{sameAs}, w)[n \otimes m]$
f-rdfp11	$(u, p, v)[n]$ $(u, \texttt{sameAs}, u')[m]$ $(v, \texttt{sameAs}, v')[l]$	$u' \in U \cup B$	$(u', p, v')[n \otimes m \otimes l]$

Figure 6: Fuzzy pD*-Entailment Rules (Part). From [6].

again. Using the entailment rules fuzzy RDF triples can be derived from a fuzzy graph.
The authors define the Best Degree Bound (BDB) n of a derived triple (s, p, o) from a fuzzy RDF graph as the highest fuzzy degree that can be obtained using the given entailment rules. Example: You have the triples (coin1, depicts, apollo)[0.1], (coin1, sameAs, coin2)[0.8] and (coin2, depicts, apollo)[0.3] you can derive (coin1, depicts, apollo)[0.1] and (coin1, depicts, apollo)[$0.8 \cdot 0.3 = 0.24$] when using the product rule as $\otimes$. (coin1, depicts, apollo)[0.24] would be the Best Degree Bound. The authors showed that a BDB always exists for a fuzzy RDF triple derived from a graph. They also defined a way for a partial pD* closure which can be computed within polynomial time.

3.4 Academic Meta Tool

In the paper [16] the authors describe another way to reason in uncertain graphs using context dependent axioms that vary per use case. They show some example use cases in their interactive web tool.
To use their Academic Meta Tool (AMT) you first define the concepts which are categories for the nodes in your RDF graph. Then you define the roles that represent the relationships between the categories and have a certainty value between one and zero. The last step is to define axioms which the AMT uses for the reasoning. There are role chain axioms like in figure 7 that are similar to the owl:ObjectPropertyChain

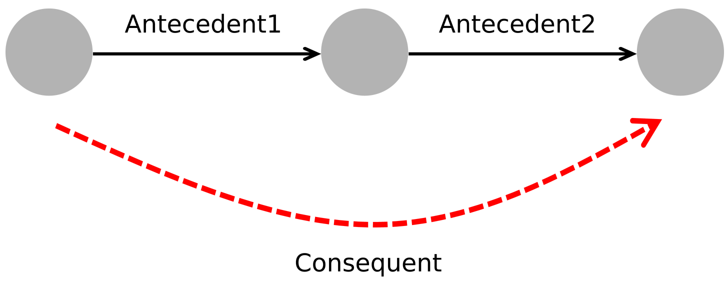

Figure 7: Role chain axiom illustration. From [16].

in OWL 2 but in AMT you can also define which logic to use for the chain. You can choose between Lukasiewicz, Product and Goedel logic. With the inverse axiom in figure 8 you can describe that two roles are inverse to each other like owl:inverseOf. For consistency checking you can also define disjoint axioms and self-disjoint axioms.

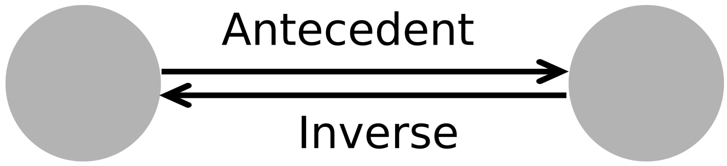

Figure 8: Inverse axiom illustration. From [16].

4 Methods

This section is about RDF (and its extensions) and how to use it to model uncertainty using reification and RDF-star.

4.1 RDF, RDFS and OWL

RDF (Resource Description Framework) and RDFS (RDF Schema) are fundamental technologies in the field of Semantic Web and Linked Data. They provide a structured way to represent and describe data on the web as a graph, enabling machines to process more complex queries with semantic meaning. One of the core concepts of RDF are the triples, which represent subject-predicate-object relationships. These relationships, also called statements, allow data to be structured in a way that captures not just the content but also the broader context. In a RDF graph, a node can be anything from a simple textual string to a complex data structure, and these nodes are connected through predicates that define the relationships between them. Every RDF resource including the properties (predicates) have a Uniform Resource Identifier (URI).
RDFS extends RDF by introducing a basic ontology system. RDFS defines essential concepts such as classes, property description and class hierarchies, allowing for the creation of simple ontologies. Ontologies are knowledge models that help to further define the meaning and context of RDF resources. With RDFS you can define classes to categorize resources and properties to describe relationships between them. For example, there is an ontology with the classes `Coin` and `Mint`. To describe the property `hasMint` the properties `rdfs:range` and `rdfs:domain` can be used to express that the usage is `Coin hasMint Mint`.
The Web Ontology Language (OWL) further extends the ontology system from RDFS to for example express inequality or equality between two classes or properties. It also supports description of property characteristics like inverse, transitive or symmetric.

4.2 Reification and RDF-star

Reification in the RDF context is a mechanism that allows you to represent statements about statements using a blank node and `rdf:Statement`, `rdf:subject`, `rdf:predicate` and `rdf:object`. In figure 9 you can see a reification example where the statement `ex:coin1 nmo:hasMint nm:rome` is represented and further described with a certainty value `ex:certaintyValue` and a model `ex:accordingTo` that created the statement.

RDF-star [10] is another way of representing statements about statements in RDF. It is currently only a draft and not part of the W3C RDF recommendation, but it is already implemented in open-source applications like Apache Jena. Reification uses four triples to describe the statement plus two triples with additional information about the statement for our example in in figure 9. RDF-star on the other hand allows the description of statements directly without a blank node so you only need three RDF statements

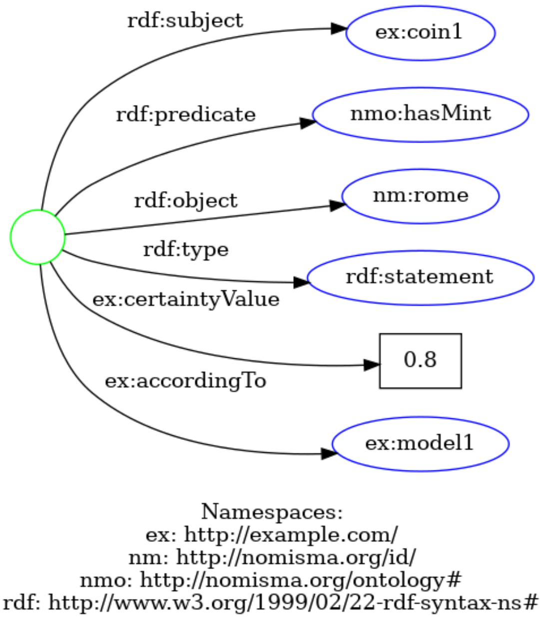

Figure 9: Reification example.

for our example. In figure 10 you can see the example written in Turtle-star notation. Research from [9] has shown significant improvements in runtime when switching from reification-based modelling to using RDF-star. However, RDF-star is currently in development and not yet fully implemented in most systems.

```
@prefix ex: <http://example.com/> .
@prefix nm: <http://nomisma.org/id/> .
@prefix nmo: <http://nomisma.org/ontology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<< ex:coin1 nmo:hasMint nm:rome >> ex:certaintyValue "0.8" .
<< ex:coin1 nmo:hasMint nm:rome >> ex:accordingTo ex:model1 .
```

Figure 10: Example from figure 9 as Turtle-star.

5 Numismatic Use Cases in RDF

In this section we will look at the use cases from section 2 and try to solve them using RDF and currently available tools from section 4. The URDF framework from section 3.2 showed no potential use case yet because the use cases are more about handling and merging uncertainty values rather than eliminating inconsistencies using constraints. URDF is also using an approximative algorithm, so you probably won't get the optimal solution. The Fuzzy RDF approach from section 3.3 isn't used either because the use cases don't depend on using complex OWL hierarchies so using the simpler AMT approach is sufficient.

The Academic Meta Tool and the context-aware approach are both using reification like figure 9 to represent the uncertainty in the graph. When RDF-star is added to the W3C RDF recommendations and gets broader support in RDF tools you could also modify those models to work with RDF-star to improve the runtime and graph size.

5.1 Use Case 1: Coin Type and Coin Similarity

Generally, this use case could be solved using the Academic Meta Tool when only one AI model or expert opinion is used to determine the similarity between coin types. The problem with multiple models is that the AMT only uses the path with the highest probability. An example is in figure 2. For the connection between coin type 1 and coin type 2 the AMT would only consider model 3 because it has the highest similarity value among the connections.
To tackle this there can be a preprocessing step that merges those 3 similarity values on all connections between coin types. The Dempster–Shafer theory from the context-aware approach is not applicable for this because the similarity is not a classification where all values add up to one. Similarity only represents the connection between two coin types and the similarity to other coin types is irrelevant. That's why the preprocessing step can just take the average or median of those values, create a new connection between the coin types and delete the old ones so they don't interfere while reasoning with the AMT. Looking at

Model	**Coin Type 2**	**Coin Type 3**	**Coin Type 4**
Model 1	0.5	0.3	0.7
Model 2	0.3	0.4	0.5
Model 3	0.6	0.1	0.9
Merge Func			
Average	0.47	0.27	0.7
Median	0.5	0.3	0.7

Table 2: Example connection values between coin type 1 and other coin types according to different models.

the example from 2.1 we had coin 1 classified as the coin type `ric.1(2).aug.4A` with confidence of 0.8 and coin 2 is classified as `ric.1(2).aug.5` with a confidence of 0.7. The similarity value between the coin types is 0.9. Using AMT role chain axioms with product logic we can calculate the probability of coin 1 being `ric.1(2).aug.5` and with this new connection you can define another role chain axiom to calculate the similarity between coin 1 and coin 2 like in figure 11. Additionally, you could normalize the connections between coins and coin types so they sum up to one and calculate the probability of coin 1 being `ric.1(2).aug.5` instead of `ric.1(2).aug.4A`: $\frac{0.72}{0.72+0.8} = 47\%$.
If a coin is to corroded to classify them as, for example, either `ric.1(2).aug.5` or `ric.1(2).aug.4A` you could connect the coin to both coin types directly with a confidence of 1 divided by the number of alternatives $1/2 = 50\%$.

5.2 Use Case 2: Issuer/Depicted Person

For this use case AMT can't deal with the uncertain checkbox or issuers that don't match the depicted person. This time the Dempster-Shafer theory [12] from the context-aware approach can be used the list with issuers and depicted person adds context to handle.

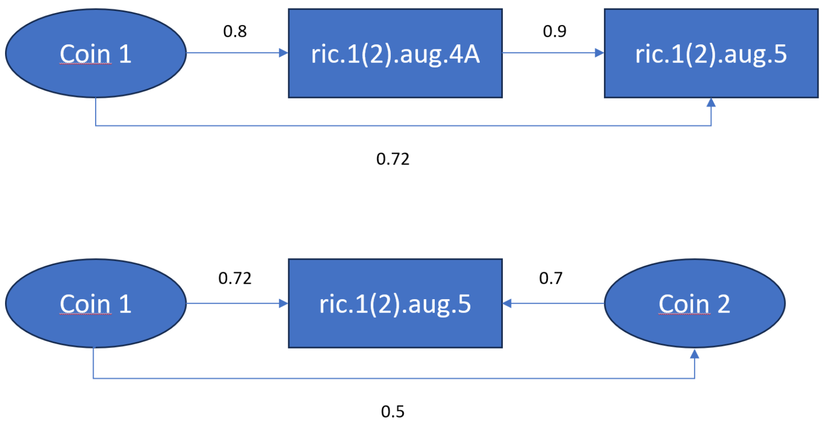

Figure 11: Coin similarity use case as AMT role chain axioms with product logic.

The first mass function is the probability of the n alternatives which is $\frac{1}{n}$. For the uncertain checkbox could be represented as another subset which contains all issuers and is also another alternative. The domain knowledge is another mass function which is uniformly distributed on the issuers for the given depicted person. In table 3 you can see an example with 3 alternatives and the uncertain checkbox checked. You know that only issuer alternatives 1 and 2 already have coins with the selected person. To simplify only one depicted person is selected with a confidence of 1 (no alternatives and uncertain checkbox not selected). Alternative depicted persons will add another layer which needs additional handling.
In the final graph you can use the joint mass as the connection value between coin and issuer. One problem with this approach is how the Dempster-Shafer theory handles non shared subsets like in table 4. The second mass function from the domain knowledge may need a predefined fraction of mass distributed over all issuer to tackle this.

Subset	**Probability**	**Domain Knowledge**	**Joint Mass**	**Plausibility**
{Alternative 1}	0.25	0.5	0.5	0.5
{Alternative 2}	0.25	0.5	0.5	0.5
{Alternative 3}	0.25	0	0	0
{All issuers}	0.25	0	0	1

Table 3: Example AFE use case with issuer alternatives using DST with normalization. The first three subsets only contain a single element and {All issuers} contains all issuers that can be selected.

Subset	**Probability**	**Domain Knowledge**	**Joint Mass**	**Plausibility**
{Alternative 1}	1	0	0	0
{Alternative 2}	0	1	0	0
{All issuers}	0	0	0	0

Table 4: Problematic use case for DST.

5.3 Use Case 3: Closing Date for a Hoard

For this use case AMT has no axioms to handle dates or date ranges and the fuzzy entailment rules can't handle those either. It is also not solvable using soft and hard rules because you don't want to resolve inconsistencies by enforcing constraints. Only the context-aware approach could work when you use the years in the subsets to model the date ranges. The problem will be coins with no overlapping date ranges which results in the same problem as seen in table 4. In table 5 you can see the example from table 1 solved using the Dempster-Shafer theory. The last step could be to choose the earliest year from the subsets and assign it with the subsets joint mass value to get the possible stop dates (on the year level) with their probability:

- 73: 0.6
- 85: 0.22
- 100: 0.18

Subset	**cn type 20885**	**cn type 21047**	**cn type 21142**	**Joint Mass**
{72, 73}	0	0.6	0	0.6
{70, ..., 85}	0.55	0	0	0.22
{30, ..., 100}	0	0	1	0.18
{All years in scope}	0.45	0.4	0	0

Table 5: Stop date use case from table 1 solved using DST. Subsets contain years BC.

6 Conclusion

This paper is a broad direction how to solve numismatic use cases using RDF and existing tools and approaches. Three use cases were defined and four tools from different domains were presented handle those use cases. The Academic Meta Tool and the context-aware approach using the Dempster-Shafer theory showed promising results for numismatic RDF graphs while the Fuzzy RDF graphs and the MaxSAT algorithm for soft and hard rules from URDF had no potential use case yet. Fuzzy RDF graphs could be used for more complex OWL hierarchies in the future but for the current use cases the simpler AMT approach is sufficient. URDF uses constraints (hard rules) to eliminate inconsistencies but for the defined use cases you rather need to merge and aggregate multiple probability values from the inconsistencies.

Further research could extend AMT to handle and aggregate multiple connections for the use case 1 similarity models. Additionally, the use of RDF-star could be implemented to test the performance against the current reification approach. Use case 2 and 3 also show the potential use of the Dempster-Shafer theory to combine probability values but further research is needed to verify these findings and translate the use cases into mass functions.

References

[1] *AFE-Web*. URL: http://www.bigdata.uni-frankfurt.de/afe-web/ (visited on 11/25/2023).

[2] Hamdi Aloulou, Mounir Mokhtari, Thibaut Tiberghien, Romain Endelin, and Jit Biswas. "Uncertainty handling in semantic reasoning for accurate context understanding". In: *Knowledge-Based Systems* 77 (2015), pp. 16–28.

[3] Thomas Bendig. *Viel wertvoller als Öl und Gold*. 2020. URL: https://www.fraunhofer-innovisions.de/big-data/lebendige-zukunft/ (visited on 09/14/2023).

[4] Petr Hájek. In: *Metamathematics of fuzzy logic* (1998).

[5] David S Johnson. "Approximation algorithms for combinatorial problems". In: *Proceedings of the fifth annual ACM symposium on Theory of computing*. 1973.

[6] Chang Liu, Guilin Qi, Haofen Wang, and Yong Yu. "Fuzzy reasoning over RDF data using OWL vocabulary". In: *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. Vol. 1. IEEE. 2011, pp. 162–169.

[7] Ndapandula Nakashole, Mauro Sozio, Fabian M Suchanek, and Martin Theobald. "Query-time reasoning in uncertain RDF knowledge bases with soft and hard rules." In: *VLDS* 884.6 (2012), pp. 15–20.

[8] *Online Coins of the Roman Empire (OCRE)*. URL: http://numismatics.org/ocre/ (visited on 11/25/2023).

[9] Jan Luca Pöpperl. "Modellierung von Unsicherheiten in Daten: Benchmarktests verschiedener Ansätze". In: (2023).

[10] *RDF-star Working Group Charter*. URL: https://www.w3.org/2022/08/rdf-star-wg-charter/ (visited on 11/05/2023).

[11] Ram Sabah and Zeena Sabah. "Modeling Nomisma ontology and Comparing Solutions for Uncertainty". In: (2022).

[12] Kari Sentz and Scott Ferson. "Combination of evidence in Dempster-Shafer theory". In: (2002).

[13] Herman J. ter Horst. "Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary". In: *Journal of Web Semantics* 3.2 (2005). Selcted Papers from the International Semantic Web Conference, 2004, pp. 79–115. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2005.06.001. URL: https://www.sciencedirect.com/science/article/pii/S1570826805000144.

[14] Karsten Tolle and Sebastian Gampe. "Creating an Additional Class Layer with Machine Learning to counter Overfitting in an Unbalanced Ancient Coin Dataset". In: 2023. DOI: https://doi.org/10.5281/zenodo.8298078.

[15] Karsten Tolle and David Wigg-Wolf. "Uncertainty handling for ancient coinage". In: 2014, pp. 171–178.

[16] Martin Unold, Florian Thiery, and Allard Mees. "Academic Meta Tool. Ein Web-Tool zur Modellierung von Vagheit". In: *Zeitschrift für digitale Geisteswissenschaften, Die Modellierung des Zweifels—Schl und-Konzepte zur Graphbasierten Modellierung von Unsicherheiten, Sonderband* 4 (2019).

[17] *W3C Uncertainty Reasoning for the World Wide Web XG*. URL: `https://www.w3.org/2005/Incubator/urw3/wiki/UncertaintyOntology.html` (visited on 11/05/2023).

[18] Huadong Wu, Mel Siegel, Rainer Stiefelhagen, and Jie Yang. "Sensor fusion using Dempster-Shafer theory [for context-aware HCI]". In: *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No. 00CH37276)*. Vol. 1. IEEE. 2002, pp. 7–12.