Ontology evolution from RDF streams using possibilistic axiom scoring

Alda Canito[a,*], Jérôme David[b], Juan Corchado[c] and Goreti Marreiros[a]

[a] *GECAD/LASI, ISEP, Polytechnic of Porto, rua Dr. António Bernardino de Almeida, 4249-015 Porto, Portugal*

[b] *Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France*

[c] *Department of Computer Science, University of Salamanca, Salamanca, Spain*

Abstract. Evolving an ontology involves re-learning, re-enriching and re-validating knowledge in the face of changes to the domain, and techniques applied for them can be adapted to ontology evolution. The possibilistic approach to axiom scoring has been applied over complete and large datasets in ontology learning. This paper presents an adaptation of the possibilistic approach to axiom scoring to the context of RDF data streams for ontology evolution, a scenario which forcefully deals with incomplete and time-dependent data. Possibilistic axiom scoring is used in two distinct scenarios: (1) with previously known property axioms, allowing for the exploration of the effectiveness of the approach in a scenario in which no incorrect data was present; and (2) in an evolving knowledge scenario, in which neither the properties nor the axioms were known and the dataset was obtained from publicly available sources, possibly both incomplete and with errors. Results show the effectiveness of the approach in accepting/rejecting axioms for the ontology's properties. The different approaches to possibility and necessity proposed in literature are recontextualized in terms of their bias towards selective confirmations or counterexamples – showing that some axioms benefit from a more lenient approach, while others present a lower risk of introducing inconsistencies by having harsher acceptance conditions.

Keywords: Ontology Evolution, Time-Sensitive Data, Data Streams, Property Axioms

1. Introduction

Ontology Evolution – especially when performed automatically or semi-automatically – cannot be detached from other subfields of ontology studies in computer science, such as ontology learning, enrichment, and validation [1,2]. Many steps and techniques are transversal to these fields; consider, for example, how evolutionary processes occur: new knowledge needs to be learned so it can be formalized into the ontology, and data can be used to further enrich the evolving schema into a more expressive and precise result. Ontology evolution is, in a way, the process of re-learning, re-enriching and re-validating an ontology in the face of changes to the domain, particularly when these are triggered by the data itself.

While many ontology learning approaches imply the acquisition of data through text, more recently there has been a shift towards using continuous streams of (structured and semi-structured) data [1–3]. Data streams carry, implicitly or explicitly, a time dimension that can and must be taken into consideration if it is meant to guide evolutionary processes [1,3–5]. This means the ontology is not learned once, but that learning and evolving the ontology are inextricably linked and the data used to trigger those processes is both limited and transient.

Ontology learning and evolution solutions tend to focus on the identification and materialization of changes in concepts and roles and the hierarchies between them, but the same attention has not been given to the axiomization of said structures [6] – particularly when dealing with the inherent incompleteness that comes with data obtained via streams.

The TICO (Time Constrained instance-guided Ontology Evolution) [7] tool is an ontology evolution framework that analyses new ontology individuals to

[*] Corresponding author. E-mail: alrfc@isep.ipp.pt.

understand if the concepts defined on the ontology have changed over time. The framework implements a set of operators that compare definitions present in an ontology to the structure of individuals incoming through a Resource Description Framework (RDF) stream. The result of the application of these operators is a set of evolutionary actions that can be triggered to produce changes in the ontology – generating new versions of specific concepts and properties to suit the patterns found in the individuals. If sufficient difference between the individuals of the concepts being analyzed and the version of them asserted in the ontology is identified, TICO uses a 4-D Fluents [8] approach to reify new, disjoint definitions of the concept for each time period – or Time Slice – in a strictly positive monotonic fashion; the tool aims to iteratively evolve an ontology as a result of the analysis of small numbers of individuals. The architecture of TICO has been described in [7] and allows for the analysis of streams of RDF individuals to extract potential evolutionary actions that add temporally-bound axioms to the ontology. In this paper, the authors use the architecture of TICO as a base for stream-guided ontology evolution with a focus on the identification of ontology property constructors. This is done by analysing extensional evidence for and against each of the ontology roles' constructors and ascertain if the data shows enough support for their inclusion in newer versions of the ontology.

The work presented in this paper aims to assess the degree to which it is possible to identify changes in OWL property axioms through the analysis of incomplete, unbounded and changing RDF data provided by streams, and to establish which metrics and assumptions/constraints are more suitable for this task. To do so, axiom testing analysis from both statistical and possibilistic perspectives will be executed. Additionally, when applying the solution to an ontology evolution scenario, this work aims to assess if and how the knowledge already present in the ontology should affect the decision to include/exclude suggested axioms.

The main contributions of this work are therefore:

1. Adaptation of the possibilistic approach to axiom testing as described in the works of Tettamanzi et al [9] to the context of streams of RDF individuals/instances, followed by an extensive, in-depth analysis of the robustness of the proposed metrics. This includes the analysis of the effects of the number and variety of the individuals that can be analysed simultaneously when searching for potential axioms in RDF data streams.
2. Combining ARI with other metrics in order to accommodate for the potential existence of errors in data: the percentage of selective confirmations w.r.t the support and an evolving form of the acceptance/rejection index which is informed by the knowledge present in previous versions of the ontology. For this purpose, the approaches will be tested against an ontology generated from publicly available data, for which no *a priori* information about property characteristics is known.

The rest of the paper is organized as follows: Section 2, Background, which contextualizes the work in the field of ontology learning/evolution and describes existing works on axiom scoring; Section 3, Property Axiom Scoring, which describes the solution and details the definitions of the axioms pertaining to each of the property characteristics in OWL and how to evaluate their presence in an RDF data stream. Section 4, Effects of Sliding Window size and suitability of ARI for axiom scoring, establishes the suitability of the possibilistic approach and the Acception/Rejection Index to axiom scoring in RDF streams and compares its performance with that of traditional information-retrieval metrics; Section 5, Accommodating for errors in data and the effects of previous knowledge in axiom-inclusion decisions, in which the combination of ARI and percentage of selective confirmations w.r.t. support are compared with an evolving form of ARI to assess how they deal with potentially incomplete and noisy data obtained from public datasets. Finally, Section 6 presents the Conclusions.

2. Background

Changes to domain knowledge must first be identified and quantified, so that they can be materialized into executable actions that will modify an ontology into a new version of itself – i.e., evolutionary actions [10,11]. To identify if there is a need for a change at the ontology level – if and which evolutionary actions should be considered – it is often necessary to analyze data that is external to the ontology [12,13]. Examples of this include identifying changes in the way end users apply the concepts in the ontology to the data [11], evaluating corpora regarding the domain to extract new concepts and compare them to existing ones [13–15] (frequently through the application of Natural Language Processing algorithms) and the analysis of datasets of structured data [10,11] (such as RDF datasets). It is possible to conceive ontology evolution as ontology learning with extra steps: new concepts and roles must be derived from existing data, but they also must be compared and made consistent with previously established ones, or otherwise update their definitions.

The identification and execution of evolutionary actions when they concern the addition of classes and properties – and the hierarchies between them – are relatively well documented in literature [6,11,16]. The identification of the axioms that could enrich them (e.g. class expression restrictions and property axioms), however, is not as popular – potentially stemming from the fact that many ontologies are often used as taxonomies and lack axiomatic complexity [16]. This axiomatic complexity, however, is particularly relevant for lower-level ontologies, which need to describe the intricacies of their domains with varying degrees of detail and can be used to determine the consistency of data they describe, and to derive implicit information using reasoners [17]. For that, it is necessary not only to identify changes in concepts and roles and their relative novelty, but also to enrich them with axioms: ontology evolution is the more useful the more it applies the precepts of ontology learning. Furthermore, [6,16] detail the relative popularity of different types of evolutionary actions as they are described in literature, noting that while identifying new properties and property hierarchies is often considered, identifying and changing their property axioms in particular is still a relatively understudied field.

Ontology learning can be defined as the processes and techniques applied to design ontologies either automatically or semi-automatically [2]. According to [18], said techniques can be classified into two main categories: (1) linguistic-based approaches and (2) machine learning-based approaches, which can be further divided into statistics or logic-based.

Linguistic-based approaches focus on the analysis of large corpora of text to identify potential concepts and the relationships between them [18–20]. To do so, they seek for patterns and syntactic information in the text – making them particularly language-dependent. Machine learning-based approaches, on the other hand, can use different types of input data for their training: both structured and unstructured. Statistics-based approaches are usually applied to identify the co-occurrence of terms, association rules and hierarchies [16,18]. Logic-based approaches, on which the work described in this paper is grounded, use logical inference or inductive logic programming to derive rules from positive and negative examples found in structured datasets. This approach is particularly suited for learning rules and formalizing axioms. On the Ontology Learning Layer Cake [21], which is used to describe the layers of the learning process and, by extension, the possible tasks it encompasses, rule and axiom extraction is depicted as the final sub-task in the process and the least explored in literature [18].

In OWL, an axiom is a statement that expresses what is true in the domain described through the ontology. The number and type of axioms directly affect the expressivity of the ontology by adding more information to the description of classes, properties, and assertions, among others, and the relationships between them. For example, an ontology that can specify if a certain property is a mandatory element of a class, or how many times that role can be applied to an individual, is more expressive (and potentially more complete) than one in which those assertions are not established [2].

Axiom testing is the process of evaluating the credibility of a given hypothesis concerning the relationships in a domain – the property axiom – by assessing whether the individuals of said domain (e.g. facts of an RDF dataset) confirm or deny a hypothesis [9] – i.e., whether they are confirmations or counterexamples of the axiom. The selective confirmation [22] principle can be put into effect as well: a fact selectively confirms a hypothesis when not only it favours that hypothesis but also fails to confirm its negation. Not all facts are equally relevant for axiom testing, and, as such, the number of examples needs to be considered not in the context of all analysed instances, but only of those that do entail the relationships under scrutiny [9].

Property characteristics, from a perspective of axiom suggestion for ontology enrichment purposes, have been described in [23]. This work describes enrichment methods that have been implemented as part of the DL-Learner framework. The approach involves using different queries to look specifically for axiom support in a triple store, considering both the count of examples and the average of the 95% confidence interval when suggesting axioms to the user. However, while the approach tackles the discovery and materialization of ontology axioms, including property axioms, it depends on (large) knowledge bases containing a complete collection of samples that can be analysed as a whole – and by analysing only these samples to generate possibilities it is, in a way, working under a closed world assumption. Similarly, [24] describes how to identify and materialize changes in property axioms but requires large and contained datasets to do so, falling under the same assumption.

The possibilistic approach has been applied in the works of Tettamanzi et al. [9,25–28] for axiom testing against RDF facts in ontology learning and validation/evaluation scenarios. As the name indicates, the possibilistic approach deals with the degree of possibility of an event, such as an axiom – which falls in a range between impossible and possible [0,1]. Possible, unlike probable (from probability distributions), does not mean that the axiom must be true: only that it is compatible with the known state of the world. While probability theory is suited for the representation of random and observed phenomena, the possibilistic theory better reflects how to deal with incomplete knowledge. In [28] (and, by extension, [9]), the authors detail how using such an approach is more suitable for candidate axiom scoring than statistical/frequentist approaches, claiming that the nature of the inductions necessary for ontology learning and evolution makes it such that statistical analysis ends up with largely arbitrary and unobjective results. Through the analysis of the results obtained in Section 4, the authors of this paper reached the same conclusion, and the possibilistic approach will be used to complement the statistical analysis. Being an unsupervised approach, the possibilistic approach is particularly suited for application to RDF streams and can accommodate for the incomplete nature of the data they provide. However, to the best of our knowledge, the possibilistic approach has only been applied for ontology validation, and its applicability to suggest new axioms from data analysis is either understudied or non-existent.

3. Property Axiom Scoring

3.1 Problem Definition

The application scenario of the TICO framework involves the existence of several streams of sensor data arriving in near-real time which are subject to transformations by data scientists, introducing new and unpredicted changes that are not in the original ontology. These changes, observed over a period of time, are used to trigger ontology evolution processes. While the original application scenario was in the field of Predictive Maintenance, TICO is agnostic and can suggest changes to any ontology following changes in the individuals of the streams under analysis. TICO receives as input an ontology and a stream of individuals – through a set of sequential RDF facts – and outputs a set of evolutionary actions that can be executed over the original ontology to generate a new, evolved version of it.

This paper focuses on the changes introduced to TICO that go beyond the detection of new properties and into recognizing the property axioms that should be added with them. Unlike other logic-based approaches, TICO does so by analyzing positive and negative evidence regarding a set of known and predetermined rules – the property axioms – to ascertain if they should be added to the ontology. To better understand the property axiom testing process on which this paper focuses, consider Figure 1 and the descriptions that follow.

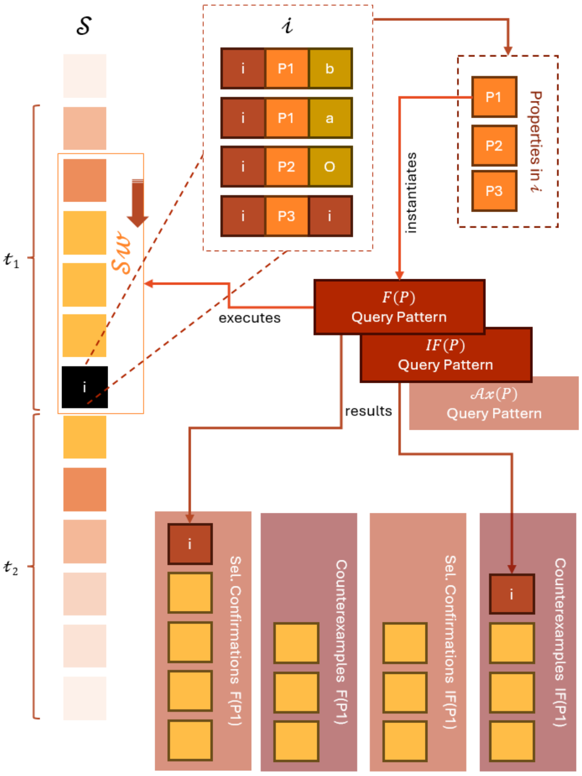


Figure 1 – Data structures concerning the property axiom testing process

The analysis is triggered by the arrival of a new individual on the stream. The visualization provided by Figure 1 shows:

- The timeframe t, the period in which the stream $\mathcal{S}$ is analysed and from which to draw conclusions
- $\mathcal{Sw}$, the sliding window upon which queries are executed

- The named individual i, which is composed by a set of RDF triples with the same Subject
- The use of i and its properties to instantiate query patterns for specific $\mathcal{A}x(P)$
- The query results being used to classify i as selective confirmation or counterexample for each $\mathcal{A}x(P)$.

TICO receives as input a stream of individuals $\mathcal{S}$, arriving continuously. Each named individual i in $\mathcal{S}$ is considered complete upon arrival (no new facts about i are analysed a posteriori, guaranteeing each individual is only analysed once). Because $\mathcal{S}$ is unbounded – with unknown start and ending points and potentially infinite –, it is necessary, at some point, to assume sufficient analysis has been done and decisions that may result in ontology evolution can be made. With this in mind, the analysis the individuals delivered by $\mathcal{S}$ is executed during a particular period of time (or timeframe) t, at the end of which an evaluation of the results is made to generate evolutionary actions. t represents a period in the stream that can be determined by the timestamps on the individuals themselves, a specific number of individuals or an actual time interval. From this setup, **Definition 1** and **Definition 2** can be elicited, such that:

Definition 1 (RDF Stream) – *An RDF stream is a (possibly infinite) sequence of triples <Subject, Predicate, Object>, in which: Subject is an IRI or blank node; Predicate is an IRI, and Object is a IRI, blank node or literal.*

Definition 2 (Timeframe) – *An RDF stream can be divided into (finite) subsequences (t_1, ..., t_n) of triples called timeframes. Data about a particular individual i (i.e., triples for which i is the Subject) are bounded to a single timeframe, and arrive sequentially.*

The consideration noted in **Definition 2** steams from TICO's original application scenario – real-time sensor data – in which each individual encapsulates a sensor reading that took place at a particular moment, and no new data is added to previous readings. The remaining elements present in Figure 1 are described next. Each individual is only analysed once per timeframe. Should the same individual feature in more than one timeframe, they are considered distinct by TICO.

Sliding Window: With the exception of Irreflexivity, all property axioms pertain to the relation between one individual and individuals other than itself. Since individuals arriving on a stream are otherwise lost after arrival, and keeping a permanent copy of each is not sustainable nor desirable, they are stored in a temporary structure upon which the analysis is executed – in a sliding window $\mathcal{S}w$ of fixed length. In a first-in-first-out fashion, every time $\mathcal{S}$ delivers a new individual, it is added to $\mathcal{S}w$, until its maximum size is reached – i.e., the size of the sliding window is measured by the total number of individuals it can store, regardless of how many triples compose them. Afterwards, for each new arrival, the oldest individual is removed from $\mathcal{S}w$ and forgotten.

Property Constructor Axiom: property axioms $\mathcal{A}x$ - or *characteristics* [23] - can be used to describe how the property must be employed. There are seven property axioms in OWL, namely: Functionality, Inverse Functionality, Transitivity, Reflexivity, Irreflexivity, Symmetry and Asymmetry. The definition of Reflexivity, however, is too strong to evaluate properly considering the constraints of this solution – requiring the analysis of all individuals at the Class level, which is not compatible with the property-oriented approach – and it is not in the scope of this paper. Focusing exclusively on the application of property axioms in Object Properties, $\mathcal{A}x$ denotes one of the possible OWL property axioms among:

- FunctionalObjectProperty (F),
- InverseFunctionalObjectProperty (IF),
- TransitiveObjectProperty (T),
- SymmetricObjectProperty (S),
- AsymmetricObjectProperty (AS),
- IrreflexiveObjectProperty (IR)

and therefore:

$$\mathcal{A}x \in \{\, F, IF, T, IR, S, AS \,\}$$

All property axioms for a given property P can be expressed as first order logic implications in the form:

$$\mathcal{A}x(P){:}\, \forall i, \forall x_1, \dots, \forall x_n$$
$$B(P, i, x_1, \dots, x_n) \rightarrow H(P, i, x_1, \dots, x_n)$$

where i is an individual in the domain of a property P. For example, the functionality of P could be written as such an implication, in which the body B shows the simultaneous application of the same property more than once for the same individual – $B(P, i\,, x_1, x_2)$: $P(i, x_1) \wedge P(i, x_2)$ – and the head H is the resulting implication that the ranges of said property must therefore be the same – $H(P, i, x_1, x_2) : x_1 = x_2$.

Confirmation, Selective Confirmation and Counterexample: In testing a property axiom hypothesis $\mathcal{A}x(P)$, we assume the original ontology is consistent and does not entail neither $\mathcal{A}x(P)$ nor its negation. Contrary to Tettamanzi's approach [9], substitutions occur not at the triple level, but at the individual level: a named individual i counts as either one confirmation or one counterexample, regardless of how many triples compose it.

For a named individual a_0 and a property P, substitutions $i/a_0, x_1/a_1, \dots, x_n/a_n$ for $B(P, i, x_1, \dots, x_n)$ and $H(P, i, x_1, \dots, x_n)$ may be found, such that:

- If there's at least one substitution for B and the negation of H under the Unique Name Assumption (UNA): the named individual a_0 is a **counterexample**;
- There is a substitution for both B and H, and no substitution for the negation of H: a_0 **selectively confirms** the hypothesis;
- There is a substitution for B, but no substitution for H nor for $\neg H$: a_0 does not contain enough information to selectively confirm nor deny the hypothesis, and therefore is a ***weak* counterexample**;
- If there are no substitutions for B, a_0 is a confirmation, but not a selective one, and thus not considered and it is ignored.

This distinction between confirmation and selective confirmation follows from the selective confirmation principle [22]: a selective confirmation is a confirmation in which i is in the domain of P, i.e., selective confirmations are a subset of all possible confirmations. Table 1 summarizes the possible classifications for a given individual w.r.t. how it can be used in substitutions in $B \rightarrow H$.

Table 1 - Classification of substitutions

B	***H***	**Classification**
True	True	Selective Confirmation
True	False	Counterexample
True	Unknown	Weak Counterexample

Considering both UNA and the fact that only individuals in the domain of the property are counted, functionality, inverse functionality, irreflexivity and asymmetry cannot generate weak counterexamples – the imposed constraints guarantee that an individual is always classified as either a selective confirmation or counterexample – there are always substitutions for either H or $\neg H$. The unknown data brought upon by the constraints of stream and the sliding window does not affect the classification of individuals, as they are assumed to arrive complete – furthermore, counterexamples rely on more individuals being accessible in $\mathcal{Sw}$, meaning there is no uncertainty regarding their correct classification.

On the other hand, the openness of the world and unknown facts about individuals in the range of the properties greatly affect the classification process for the transitivity and symmetry axioms – consider, for instance, that the individuals in the range of the properties may not be present in the sliding window when the query is executed, and therefore it is not possible to assess if that individual is also in the domain of the property (necessary for both axioms). Selective confirmations are indisputable, as their mere existence implies that the data was, in fact, complete. Counterexamples, on the other hand, imply proving a negative, making them particularly harder to identify under the constraints and assumptions of the methodology employed. As such, any substitutions that do not selectively confirm these axioms are treated as *weak* counterexamples.

Constructor Query Pattern: To classify i w.r.t. $\mathcal{A}x(P)$, it is first necessary to ascertain if P is used by i. If that is the case, i is then used to instantiate a specific constructor query pattern – a SPARQL query which enforces the pattern associated with $\mathcal{A}x(P)$. This query will enforce the substitutions mentioned previously, meaning reasoners are not employed. Code snippet 1 shows one possible query (for symmetry), in which `iURI` and `pURI` are known and correspond, respectively, to i and P. Each i is used to instantiate the query patterns only once, although it may influence the results of queries coming afterwards (as Object).

Code snippet 1 – Query representing Symmetry

```
SELECT * WHERE {
  {
    SELECT (COUNT(?obj) as ?B) WHERE
      {  <iURI> <pURI> ?obj . }
  }

  {
    SELECT (COUNT(?obj) as ?H) WHERE
       {  <iURI> <pURI> ?obj .
          ?obj <pURI> <iURI> . }
  }
  FILTER (?B = ?H) }
```

When the query is executed over $\mathcal{Sw}$, any results it obtains classify i as a selective confirmation of $\mathcal{Ax}(P)$ (i.e. both the triple and its mirror are present). In this case, as it is symmetry, if the query does not obtain at least one result, then i is considered a *weak* counterexample of $\mathcal{Ax}(P)$.

3.2 Property Constructor Analysis

The analysis of real-time data can unravel patterns regarding a property's usage that may hint at its formal definition. Each property axiom, per virtue of its definition, corresponds to a different pattern in the data (cf. Table 2). Consider the generic task $computeAxiom$, which aims to gather statistics about property axioms and is summarized in Algorithm 1. E^+ and E^- reflect, respectively, how many individuals are categorized as selective confirmations and strong counterexamples of $\mathcal{Ax}(P)$ in $\mathcal{S}$ during $\mathcal{t}$.

Support (E^0) is provided by the sum of $E^+$and E^- and reflects the total number of individuals in the domain of P.

The details of how each property axiom is defined and the formula to search for in the data are described in Table 2.

Algorithm 1 – computeAxiom

```
Input: $\mathcal{S}$, $\mathcal{Sw}$, $\mathcal{Ax}$, $P$
Output: $E^+$, $E^-$
for each $i$ in $\mathcal{S}$ do
   if $\mathcal{Sw}$ is full then
     pop $\mathcal{Sw}$
   push $i$ to $\mathcal{Sw}$;
   $dP$ ← get distinct properties used by $i$;

   if $P$ in $dP$ then
     $Q\mathcal{Ax}$ ← get query pattern for $\mathcal{Ax}$;

     instantiate $Q\mathcal{Ax}$ using $i$ and $P$;

     if $Q\mathcal{Ax}$ has results in $\mathcal{Sw}$ then
        increase $E^+$;
     else
        increase $E^-$;
```

Table 2 – Definitions, formula, selective confirmation and counterexamples for each property axiom

$\mathcal{Ax}$	**Definition**	**Formula**	**Selective Confirmation (E^+)**	**Counterexamples (E^-)**	**Assumptions**
F	P can have one and only one value per i. P may be employed more than once by i provided the ranges are the same individual.	$\forall\, i, y_1, y_2 : P(i, y_1) \land P(i, y_2) \rightarrow y_1 = y_2$	i for which P has either only one value.	i for which P is used more than once and the ranges of P are distinct. Counterexamples of $F(P)$ are strong counterexamples.	Partial closure of the world through UNA allows for the existence of strong counterexamples.
IF	P cannot have the same entity on its range for more than one i (e.g. a unique ID cannot be shared).	$\forall\, i_1, i_2, y$: $P(i_1, y) \land P(i_2, y) \rightarrow i_1 = i_2$	i for which the value of y has not been the range of P in any previously analyzed individual in $\mathcal{Sw}$.	i for which the value of y was found in the range of P of other previously seen individuals. Counterexamples of $IF(P)$ are strong counterexamples.	
T	If P holds between any two individuals in a sequence, it holds between all individuals of that sequence.	$\forall\, i, y, \mathrm{z} : P(i, y) \land P(y, z) \rightarrow P(i, z)$	i for which P has propagated between any set of three individuals i, y, z, for which a $P(i, z)$ exists.	i for which P does not propagate between any set of three individuals i, y, z. Non-confirmations of $T(P)$ are considered **weak** counterexamples, as a strong counterexample would have to prove $\neg P(i, z)$.	Closed World Assumption allows for the elevation of non-confirmations to weak counterexamples.
IR	Characteristic of a property that cannot relate i with itself: the domain and range of the P must not be the same individual.	$\forall\, i :$ $\neg P(i, i)$ or $\forall\, i_1, i_2 : P(i_1, i_2) \rightarrow i_1 \neq i_2$	i in which P is used to relate with individuals other than itself.	i in which P is used to relate with itself. Counterexamples of $IR(P)$ are strong counterexamples.	Partial closure of the world through UNA allows for the existence of strong counterexamples.
S	Property which is its own inverse: meaning that if $P(a, b)$, then $P(b, a)$ must also be true.	$\forall\, i, y : P(i, y) \rightarrow P(y, i)$	i in which the range of P relates back to the individual through P.	i for which the range of P does not relate back to the individual through P. Non-confirmations of $T(P)$ are considered **weak** counterexamples, as a strong counterexample would have to prove $\neg P(y, \mathrm{i})$.	Closed World Assumption allows for the elevation of non-confirmations to weak counterexamples.
AS	Property which cannot be its own inverse: if $P(a, b)$, then $P(b, a)$ cannot hold true.	$\forall\, i, y : P(i, y) \rightarrow \neg P(y, i)$	i for which the range of P does not relate back to the individual through P.	i for which the range of P relates back to the individual through P. Counterexamples of $AS(P)$ are strong counterexamples.	Partial closure of the world through UNA allows for the existence of strong counterexamples.

Consider a Sw with a size of 8, whose current individuals are described in Table 3. Individuals are complete but may contain references to individuals that are not in the Sw at the time $computeAxiom$ is executed.

Table 3 - Example individuals in a Sw and respective properties

ID	hasName	hasEvolution Group	hasType	hatches
i_1	Onix	Onix & Steelix Line	Rock, Ground	Mineral EG
i_2	Ditto		Normal	Ditto EG
i_3	Caterpie	Caterpie Line	Bug	Bug EG
i_4	Metapod	Caterpie Line	Bug	Bug EG
i_5	Tropius		Flying, Grass	Grass EG, Mineral EG
i_6	Bug EG			Caterpie, Metapod
i_7	Grass EG			Tropius, Maractus
i_8	Maractus		Grass	Grass EG

Considering the potential Inverse Functionality of the $hasEvolutionGroup$ property, i.e. $IF(hasEvolutionGroup)$, the individuals are classified as follows:

- Samples: i_1, i_3 and i_4, as they use the property being analysed. E^0 = 3.
- Selective confirmations: i_1, as the domain of the property is only used once per value. E^+= 1.
- Strong counterexamples: i_3 and i_4, as they have the same property value for different individuals. E^-= 2.

Individuals i_2, i_5, i_6, i_7 and i_8 do not use the property and are not used in the analysis of $IF(hasEvolutionGroup)$. They may be used in the analysis of axioms for different properties.

Consider the potential symmetry of the $hatches$ property. For $S(hatches)$:

- Samples: i_1, i_2, i_3, i_4, i_5, i_6, i_7 and i_8. E^0 = 8.
- Selective confirmations: i_7, i_8, as both $hatches(i_7,\ i_8)$ and $hatches(i_8,\ i_7)$ are present. E^+= 2.
- Weak counterexamples: i_1, i_2, i_3, i_4, i_5 and i_6 as they use the property in relation to other individuals, but whether those individuals also use it in relation to them is not made explicit in the data. E^-= 6.

3.3 The Possibilistic Approach to Axiom Scoring

The work described in [9] also introduces the concept of an acceptance/rejection index (ARI) and its application to axiom scoring: positive values suggesting acceptance, negatives suggesting rejection and values close to zero indicating ignorance. For any $Ax(P)$, it is possible to calculate its necessity N: the degree to which the axiom is corroborated by the data while not being contradicted by it; and its possibility Π: the degree to which it is not contradicted by the data. ARI can thus be calculated using [28]:

$$ARI(Ax(P)) = N(Ax(P)) + \Pi(Ax(P)) - 1 \in [-1,1]$$

N and Π are calculated differently depending on the approach to the data being used. The original reasoning behind them can be found in [28]. For the purposes of this paper, the computation of N and Π depends on the relative weight given to selective confirmations and counterexamples.

If counterexamples are strong, a single counterexample is sufficient to quash N, regardless of how many selective confirmations are found. Ergo, counterexamples have more weight in the decision-making process than selective confirmations – when *strong* counterexamples are available, the strong form of N and Π is applied, as follows [28]:

$$N\big(Ax(P)\big) = \begin{cases} 1, if\ \mathrm{E}^-_{\mathrm{Ax(P)}} = 0 \\ 0, if\ E^-_{Ax(P)} > 0 \end{cases}$$

and

$$\Pi(Ax(P)) = 1 - \sqrt{1 - \left(\frac{\mathrm{E}^+_{\mathrm{Ax(P)}}}{\mathrm{E}^0_{\mathrm{Ax(P)}}}\right)^2}$$

in which $E^0_{Ax(P)}$ is the total number of individuals in the domain of P, corresponding to the sum of $E^+_{Ax(P\ )}$ – the number of selective confirmations, – and $E^-_{Ax(P)}$ – the number of counterexamples.

In the cases in which counterexamples are not sufficient to exclude a hypothesis – being *weak* counterexamples – the possibility of an axiom being true is directly related to the existence of selective confirmations. N and Π are calculated according to [28]:

$$N(Ax(P)) = \sqrt{1 - \left(\frac{\mathrm{E}^-_{Ax(P)}}{\mathrm{E}^0_{\mathrm{Ax(P)}}}\right)^2}$$

and

$$\Pi(Ax(P)) = \begin{cases} 0, if\ \mathrm{E}^+_{\mathrm{Ax(P)}} = 0 \\ 1, if\ E^+_{Ax(P)} > 0 \end{cases}$$

This second set of formulas ensures the decision is influenced more by the selective confirmations – with Π being at its highest whenever selective confirmations are present. N cannot be set to 0, as counterexamples are *weak* and the assumption that they are, in fact, actual counterexamples cannot be made under the Open World Assumption (OWA). For the sake of simplicity, the first set of formulas are considered the stronger forms and the second the weaker forms, after the strength of their respective counterexamples.

Table 4 summarizes which forms of N and Π (ARI form for brevity) to use per each property axiom. These reflect not only the assumptions made in Table 2, but also the constraints of the approach itself (under UNA, weak confirmations become strong for functionality and inverse functionality, for example).

Table 4 – ARI form to apply for each property axiom

Property Axiom	**ARI form**
Functionality	Strong
Inverse Functionality	Strong
Transitivity	Weak
Symmetry	Weak
Asymmetry	Strong
Irreflexivity	Strong

Returning to the example in Table 3, not all individuals in the stream are equally available to be analysed – e.g. there is mention of an individual *"Caterpie Line"* that is not among the individuals in $\mathcal{S}w$. E.g., in the case of the *hatches* property, one can see that it relates several individuals in a symmetrical fashion: i_8 relates to i_7 using this property and vice-versa. These would, indeed, count as selective confirmations of $S(hatches)$ (and for $T(hatches)$). However, all individuals in the sliding window use the property, which results in the data in Table 5.

Table 5 - Axiom scoring for the *hatches* property using the strong forms of N and Π

$Ax(\boldsymbol{hatches})$	E^+	E^-	***N***	***Π***	**ARI**
F	6	2	0	0,34	-0,66
IF	5	3	0	0,22	-0,78
AS	3	5	0	0,07	-0,93
IR	8	0	1	1	1

By not having information about individuals such as *Mineral EG* and *Ditto EG*, it is not possible to have all the data needed for the remaining individuals to count as selective confirmations. In the cases of T and S, incomplete data may lead to false negatives, and selective confirmations are much harder to find as they require more individuals to be on the same sliding window. When using the weaker forms to compute ARI for T and S, the results reflect the potential for those axioms to be present (see Table 6).

Table 6 - Axiom scoring for the *hatches* property using the weak forms of N and Π

$Ax(\boldsymbol{hatches})$	E^+	E^-	***N***	***Π***	**ARI**
T	5	3	0,93	1	0,93
S	2	6	0,48	1	0,66

While ARI can be updated with each new individual on the stream, it only influences the ontology evolution process at the end of $\mathcal{t}$. Algorithm 2 shows how the decision to include or exclude a property axiom from an ontology can be made, considering the existence of different thresholds for the weak and strong forms of ARI (wf_t and sf_t, respectively).

Algorithm 2 – ARI Decision and evolving ontology

```
Algorithm is executed at the end of $\mathcal{t}$.
Input: $ontology$, set of <$\mathcal{A}x(P)$,$E^+$,$E^-$>, $sf_t$, $wf_t$
Ouput: $ontology$ (evolved)
for each $\mathcal{A}x(P)$ do
  if $\mathcal{A}x$ one of $\{T, S\}$ then
     compute $N(Ax(P))$ using weak form;
     compute $\Pi(Ax(P))$ using weak form;
     t ← $wf_t$;
  else
     compute $N(Ax(P))$ using strong form;
     compute $\Pi(Ax(P))$ using strong form;
     t ← $sf_t$;

  $ARI(Ax(P))$ ← $N(Ax(P))$ + $\Pi(Ax(P))$ - 1;

  if $ARI(Ax(P))$  >= t then
     add $\mathcal{A}x(P)$ to $ontology$;
  else
     remove $\mathcal{A}x(P)$ from $ontology$;
```

3.4 *Applying ARI to evolving ontologies*

Considering the ontology evolution application scenario, in which a potential new version (or at least for of some of its axioms) is created at the end of each $\mathcal{t}$, any conclusions reached over the application of the queries must be made in the context of available knowledge – i.e., previous known versions of the axioms in the ontology. As such, we use a weighted average between the newly calculated ARI and the previous state of the axiom, which will be referred to as Evolving ARI (ARI_e).

$$ARI_e\big(Ax(P)\big) = \\ = \begin{cases} ARI\big(Ax(P)\big), & t = 0 \\ d\big(Ax(P)\big)_{t-1} * w_p + ARI\big(Ax(P)\big) * \big(1 - w_p\big), & t \geq 1 \end{cases}$$

in which:

1. ARI is the Acception/Rejection Index calculated during t;
2. $d\big(Ax(P)\big)_{t-1}$ is one of [0,1], depending on whether the axiom in question was (or was not) missing from the previous known version of the ontology;
3. w_p is the relative weight of previous knowledge [0-1]. Lower weights should allow for more versatility in the evolutionary process, favouring new axioms, while higher ones value a more conservative approach.

A w_p of 0 would imply that previous knowledge does not affect the current decision-making, but a w_p of 1 would equally mean that no changes to the axioms in the ontology would ever be possible, regardless of how much evidence for it is found.

Both ARI and ARI_e are on the [-1,1] interval, with positive results suggesting acceptance of the axiom and the inverse for negative results. When it comes to decision-making, however, this may not be sufficient (maybe not all positive results are strong enough to force asserting an axiom, and it may be interesting to allow for errors in the data to exist). For strong-form using properties, the decision d to accept or reject an axiom is informed by:

$$d\big(Ax(P)\big) = \begin{cases} accept, & E^{+}_{Ax(P)} > cf_t \\ accept, & E^{+}_{Ax(P)} \leq cf_t \ and\ ARI > sf_t \\ reject, & otherwise \end{cases}$$

in which $\mathrm{E}^{+}_{Ax(P)}$ is the percentage of selective confirmations (w.r.t. the Support), cf_t the minimum percentage of selective confirmations required, and sf_t is the threshold to be applied for the strong form of ARI computation. For weak-form using properties (i.e., T and S) the decision is informed exclusively by ARI and a minimum threshold wj_t. Since the existence of counterexamples does not have the same weight as for the strong form, the reasoning to include the percentage of support in the decision-making process does not apply. As such, axiom acceptance is informed by:

$$d\big(Ax(P)\big) = \begin{cases} accept, & ARI > wf_t \\ reject, & otherwise \end{cases}$$

As F and T axioms are incompatible, and F is computed on stricter terms (using the stronger-form), F is considered to take precedence over T in case of both being flagged for inclusion. As such, whenever the algorithm concludes that the same property could be both F and T, it will only classify it as F. Similarly, an axiom cannot be simultaneously T, S and IR. In these cases, precedence is given to T and S.

The following sections will illustrate the application the possibilistic approach to axiom scoring in two different scenarios:

- **Experiment I**: in this experiment, ARI will be compared to traditional information retrieval metrics to ascertain its applicability to axiom scoring in an RDF stream scenario. For this purpose, existing ontologies for which the property axioms are known will be used. This experiment will also assess the effects of different sliding window sizes in the proper classification of individuals as selective confirmations or counterexamples.
- **Experiment II**: experiments will be conducted using an ontology automatically generated from publicly available datasets to establish the applicability of the solution in an ontology evolution/learning scenario. Information regarding property axioms is not known *a priori*, and the suitability of the proposed axioms will be analysed by employing a reasoner and verifying if inconsistencies are introduced. This experiment will show the suitability of the solution in cases in which the data is both potentially incomplete and with errors. Furthermore, the individuals analysed will be segregated into different, pre-established timeframes, allowing for the application of Evolving ARI between them.

4. Experiment I: Effects of Sliding Window size and suitability of ARI for axiom scoring

To establish the strength of the devised solution, experiments were first conducted against a dataset in which certain property constructors were previously known. With the ontologies known and the datasets curated, there is no expectation of counterexamples to be found for the assertions present in the ontology.

For a better understanding and discussion of the results, they have been separated according to the following questions:

1. How many individuals should be included in the sliding window in order to effectively learn

property axioms and how does the number of samples (i.e., individuals that use the property under scrutiny) in the stream influence this number? Furthermore, the following subquestion will be investigated:

   a. How does ARI compare with precision, recall and f-measure? How does it compare to the application of selective axiom confirmations exclusively?

2. How well do the weak forms of N and Π compensate for the difficulty in identifying selective confirmations under OWA?

The experiments described below were carried out using the set of ontologies in Table 7, which were selected for their property axioms and population sizes.

Table 7 – Properties and their respective axioms by class

	Type	Property	*Ax*
CMT [29]	Paper (4301 individuals)	hasAuthor	F
		hasDecision	F
		readByMetaReviewer	F
		rejectedBy	F
	Person (4255 individuals)	addedBy	F
		addProgramCommitteeMember	IF
		assignedByReviewer	F
		assignExternalReviewer	IF
		rejectPaper	IF
		writePaper	IF
		writeReview	IF
WINE [30]	Region (36 individuals)	adjacentRegion	S
	Region (as range)	locatedIn	T
Plant [31,32]	Thing (82 individuals)	part_of/has_part	T

Each populated ontology was split into samples of 3000 random individuals, which are then sequentially subjected to the analysing queries. To simulate an RDF stream scenario, the individuals of the ontology are queued (and dequeued when necessary) into the sliding window one at a time. Queries are executed over the sliding window, guaranteeing they never have access to all existing individuals in the ontology simultaneously and therefore operate under the original restrictions of the TICO solution. For performance reasons, only the properties that displayed the characteristics mentioned above were analysed: as it is impossible for a property to employ all property axioms simultaneously, they can serve as counterexamples for those they do not display.

For the purposes of this work, a modified version of the queries described in [23] are used as confirmation queries, in which the differences account for the streaming nature of the use case and the granularity of the search (here at the individual level and not at the triple level). Two different types of queries can be used for each property axiom: the confirmation queries (CQ), which ascertain if a property axiom could be present, and the negation queries (NQ), which ascertain the opposite. Each class of query (CQ or NQ) provides its own confusion matrix, and the meaning of their solutions varies depending on whether the true class corresponds to the existence or non-existence of the axiom. This reasoning is illustrated in Table 8.

Table 8 – Confusion Matrix and respective query results

		Has solution?	**Target**	
			AXIOM	¬AXIOM
Applied Query	CQ	Yes	True Positive (TP)	False Positive (FP)
		No	False Negative (FN)	True Negative (TN)
	NQ	Yes	False Positive (FP)	True Positive (TP)
		No	True Negative (TN)	False Negative (FN)

The confirmation query looks for positive cases of functionality: i.e., if a given individual is compatible with the axiom, either by having only one use of the property or the object being a duplicate. An individual that can be selected with this query is a selective confirmation of the functionality axiom. If the property is indeed functional – i.e., if the true class in Table 8 is AXIOM – then this result is a true positive. On the other hand, if the axiom was present but the query does not yield any results, it is a false negative. If the true class is ¬AXIOM – i.e., it is known that the functionality axiom is not present – and the query returns a result, it is a false positive. Following the same reasoning, an empty set here is the correct result for the individual and therefore a true negative.

The negation query, on the other hand, looks for explicit counterexamples: for an individual to result in a non-empty set when queried, it must definitively have at least two uses of the property, and their objects be distinct. If the true class in Table 8 is AXIOM and the negation query returns a non-empty set of solutions, it must forcefully be a false positive – there should not have been any solutions for the query, as functionality is present. If it returns an empty set, it is a correct identification and a true negative. In the same vein, the true class is ¬AXIOM – and the query has results, it is doing so correctly and, therefore, a true positive. If it returns none – claiming the axiom is present when it should not be – it accounts for a false negative.

Code snippet 2 and Code snippet 3 show one possible difference between confirmation and negation queries, using the functionality axiom as an example. Each query is executed for each individual being tested and generates a result set with a size of either 0 or 1 query solutions.

Code snippet 2 – Confirmation Query for Functionality

```
SELECT ?o1 WHERE
{
   <iURI> <pURI> ?o1.
   FILTER NOT EXISTS
       { <iURI> <pURI> ?o2.
         FILTER ( ?o1 = ?o2 ) }
}
```

Code snippet 3 – Negation Query for Functionality

```
SELECT ?o1 WHERE
{
   <iURI> <pURI> ?o1. <iURI> <pURI> ?o2.
   FILTER ( ?o1 != ?o2 )
}
```

Traditional information retrieval statistics, e.g. precision, recall and f-measure are computed as follows:

$$precision = \frac{TP}{TP + FN} \quad recall = \frac{TP}{TP + FP}$$

$$f - measure = \frac{2 * precision * recall}{precision + recall}$$

To ascertain how “quickly” and effectively the queries can identify the possibility/probability of an axiom being present for a given individual, three different sliding windows sizes are analysed: 10, 50 and 100 individuals (corresponding, roughly, to 0.2%, 2.3% and 4.7% of all individuals in the sample, respectively). Theoretically, the more individuals in the sliding window that can be used to answer a query, the more precise the classification of each individual as a selective confirmation or counterexample should be. However, for performance reasons, it is interesting to see if reliable conclusions can be made from as little data as possible.

While all properties used by each class were studied, for the sake of brevity, we will focus on presenting the results for a property from a class with a high variance in the usage of its properties – i.e., not all individuals of the same type will use the same properties – and those for a class with lower variance, in which all individuals always use the same properties.

4.1 Section I – Effects of window size and relevance of individuals in axiom support

This section analyses the effects of both the window size and the influence of support in the sample. As previously explained, for an individual to constitute support, it must be the domain of the property being investigated. However, even if a stream is comprised only by individuals of the same type, there is no guarantee that all of them will employ the same properties. For the following experiments, two classes from the CMT ontology were selected – *Paper* and *Person* – as the first always employs all its properties and the second has more variety to it, with some properties being used in a very low fraction of its individuals. To ascertain the influence of the window size (w_s) of the $\mathcal{Sw}$, three different values (of 10, 50 and 100 individuals) are applied to each of the classes.

Considering that the dataset is clean, and no counterexamples can be found, the results in favour of an axiom are fairly evident regardless of the window size applied (perfect precision combined with necessity, possibility, and ARI of 1). However, it is important to investigate the more interesting cases in which counterexamples are indeed possible, by analysing the evolution of the metrics in the cases where an axiom is known not to be present.

Table 9 – Number of individuals on the stream to analyse, window sizes, and percentage of support in each sample for each of the studied properties

Variable	**Value**
individuals analysed	2150
w_s (size of $\mathcal{Sw}$)	10 / 50 / 100
Property	% of Support
readByMetaReviewer	100%
rejectedBy	≈50%
addedBy	≈25%

The values presented in Table 9 are applied on the following experiments, assessing how the relevance of each individual for support affects the evolution of the metrics and the potential results.

The *readByMetaReviewer* property does not feature any of the explored property axioms, but it is present in all individuals of the *Paper* class. The study of the evidence for and against the IF axiom for this property shows how the proper identification of each individual as a selective confirmation or a counterexample is affected by the changes in window size – that as more individuals are analysed, it becomes apparent that the same property is used by more individuals with the same value – and the number of counterexamples starts to increase. Most of the changes occur when analysing the first 60 individuals out of the sample, as shown in Figure 2, with most individuals (around 94%) being incorrectly categorized as selective confirmations due to lack of information to the contrary.

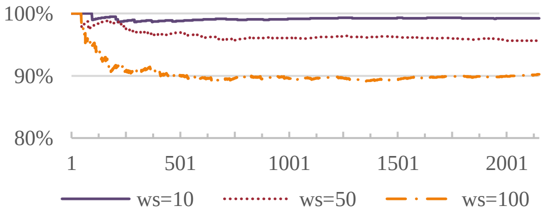


Figure 2 – Evolution of the selective confirmations of IF for the *readByMetaReviewer* property with w_s of 10, 50 and 100

However, by increasing w_s to 50, the categorization improves significantly: the number of selective confirmations lowers to around 82%, and to 71% when w_s is increased to 100.

With recall being 100% regardless of the window size chosen, Figure 3 compares the evolution of precision, showing similar improvements although with a high propensity for misclassification (stabilizing around 10%).

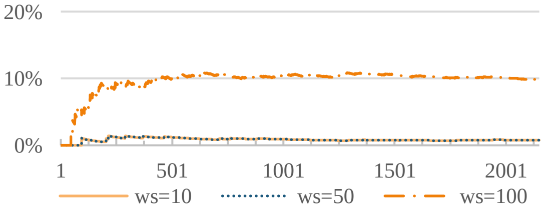


Figure 3 – Evolution of the precision of IF for the *readByMetaReviewer* property with w_s of 10, 50 and 100

ARI is always steadily negative, with the improvements in the classification of individuals changing how negative it skews, improving from -0.36 to -0.71, as shown in Figure 4. The metrics evolve differently when not all individuals in the stream are considered support. Consider the results in Figure 5, which were obtained when searching for inverse functionality of the *rejectedBy* property, which is used by 1279 individuals of the *Paper* class (≈50%).

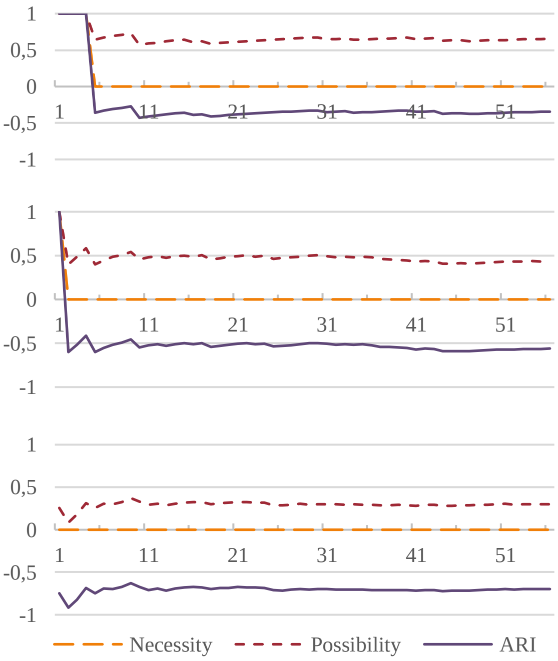


Figure 4 – Evolution of N, Π and ARI of IF for the *readByMetaReviewer* property with w_s of 10 (first graph), 50 (second graph) and 100 (third graph)

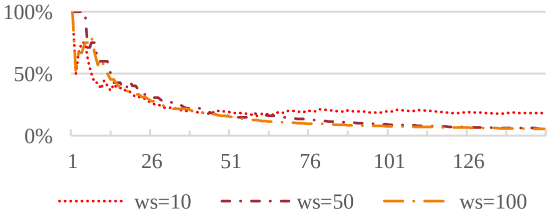


Figure 5 – Evolution of selective confirmations of IF for the *rejectedBy* property with w_s of 10, 50 and 100

Selective confirmations for the IF axiom start high (at 100%), as a single individual with a single use of the property cannot be flagged as a counterexample. After analysing 60 individuals, counterexamples account for more than 70% of the samples; 80% after 120, and their number finally stabilizes around 85% as the analysis progresses. With a sliding window of 50, there should be more available evidence for each query to ascertain if the axiom is present. This seems indeed to be the case: while the 70% threshold obtained with a w_s of 10 is equally obtained after analysing around

60 individuals, with a w_s of 50 the 70% threshold is reached after analysing only 16 samples. The percentage of counterexamples also stabilizes at a higher point (at around 94%, after circa 75 individuals). Increasing w_s to 100 shows very minor increases in both speed and accuracy. The number of counterexamples on the sample reaches the 70% threshold earlier than with w_s of 50 – after 12 individuals – and stabilizing at a negligibly slightly higher point – at 97%.

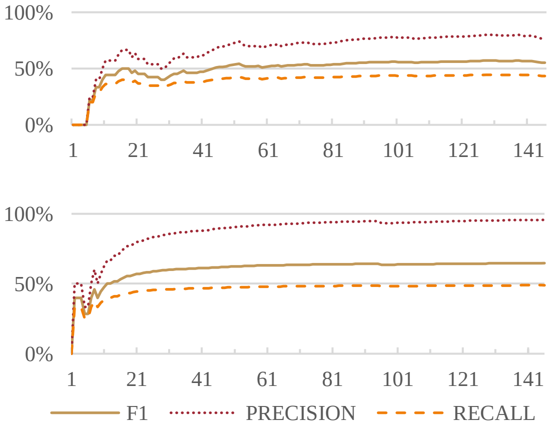


Figure 6 - Evolution of precision, recall and f-measure of IF for the *rejectedBy* property with w_s of 10 (first graph) and 50 (second graph)

Precision, recall and f-measure show similar evolution patterns as those of support, as seen in the first graph of Figure 6. All metrics start low, with a quick but unsteady increase until around 50 individuals have been analysed, and equally stabilizing as the analysis progresses. Precision peaks at around 85%, as expected from the results presented in Figure 5. Recall stabilizes around 45% and f-measure at 60%. With a similar evolution to that of the support for the same window size, as illustrated in the second graph of Figure 6, similar levels of precision, recall and f-measure are obtained but with their values stabilizing higher and later (f-measure reaching 60% after circa 40 samples and 65% after 100). This effectively shows that by having a bigger sliding window – i.e., by having more individuals available for each query to search in – it is possible to track more nuances in the data and better classify each individual as a selective confirmation or counterexample. In further increasing w_s to 100, no significant improvements are made. Precision, recall and f-measure stabilize faster, but improvements account for little more than 1%.

If one were to trust these metrics by themselves, it could be concluded that the approach can identify if the property is not IF with a certainty of 65%. Here, necessity, possibility and ARI can provide a faster and more complete idea of the classification that must be done when giving the proper weight to the counterexamples. The following Figures illustrate the evolution of ARI using the same three window sizes as before.

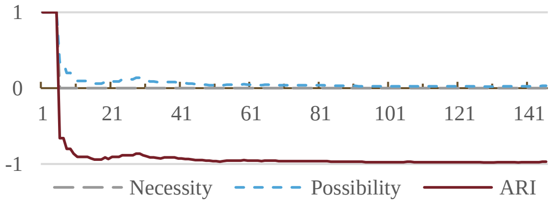


Figure 7 – Evolution of N, Π and ARI of IF for the *rejectedBy* property with w_s of 10

Figure 7 shows necessity starting at 1, as only one individual has been analysed, and it is not a counterexample. However, as the second individual provides a counterexample, it immediately drops and stays at 0. Possibility, as determined by the number of selective confirmations, suffers a gradual loss as less and less selective confirmations are found in the data, and stabilizes very close to 0 (but with a positive value, as individuals that do not explicitly contradict the axiom can still be found) after around 50 individuals. ARI, as a function of the other two metrics, shows a similar evolution from 50 individuals onwards, stabilizing very close to -1, which strongly advocates for axiom rejection, as seen in Figure 8.

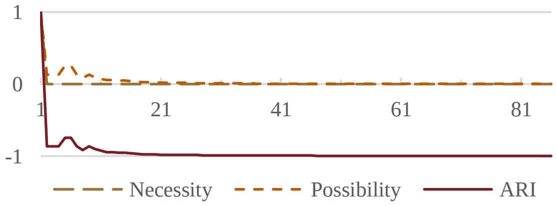


Figure 8 - Evolution of N, Π and ARI of IF for the *rejectedBy* property with w_s of 50

There are no discernible changes in the evolution of necessity, possibility and ARI when the window size is increased, as the number of selective confirmations and counterexamples is not as relevant for these metrics as the mere presence of counterexamples is. The metrics support the reasoning that a bigger window provides a better classification of each sample as either a selective confirmation or a counterexample, but only to a certain extent. Similar to support and information retrieval metrics, Figure 9 displays how ARI evolves slightly faster but is not significantly improved:

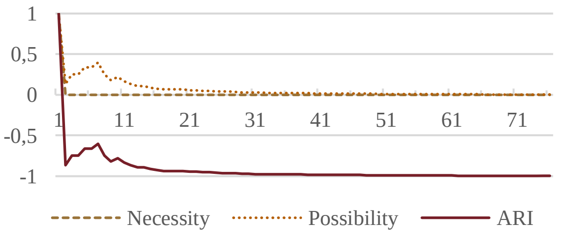


Figure 9 - Evolution of N, Π and ARI of IF for the *rejectedBy* property with w_s of 100

Unlike the *Paper* class previously explored, *Person* has even more variance in the application of its properties – e.g. the property *addedBy* is used by 734 of the 3000 individuals studied (≈25%), while *rejectPaper* is used only by 4 of them (≈0.1%). While this change in frequency affects the evolution of the metrics being studied, the results follow the same trends as before: by allowing the queries to access more individuals, metrics are improved, but only until a certain point. The speed at which they improve, however, is indeed affected by the variety in the data, as shown in Figure 10.

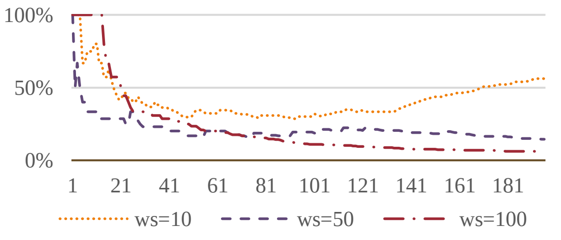


Figure 10 - Evolution of confirmations of IF for the property *addedBy* with w_s of 10, 50 and 100

The results show that if potential variations in use of the properties can be expected, increasing w_s not only allows for better classification as selective confirmation or counterexample, but also does so correctly significantly faster. The same conclusion can be drawn from the analysis of the differences in precision, recall and f-measure for the three values of w_s, shown in Figure 11. Furthermore, it is important to note that for the smaller values of w_s, the evolution of the metrics is not monotonic (see the first graph of Figure 11). This suggests a lower w_s can be detrimental when parsing a large number of individuals.

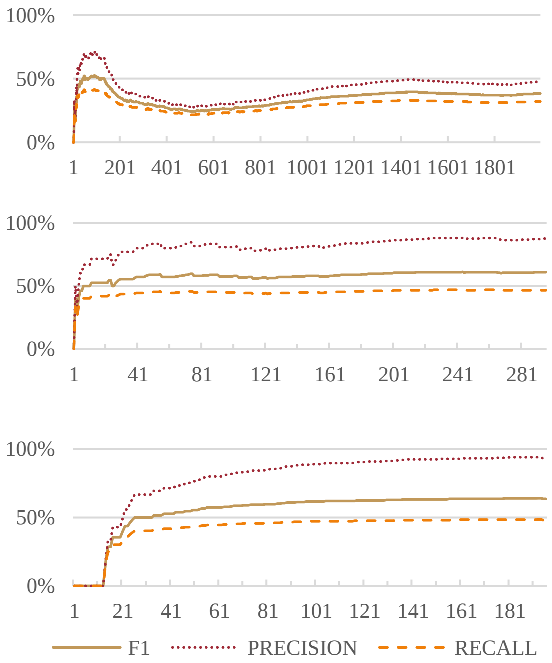


Figure 11 - Evolution of precision, recall and f-measure of IF for the property *addedBy* with w_s of 10 (first graph), 50 (second graph), and 100 (third graph)

Finally, the changes in the evolution of necessity, possibility and ARI follow the information retrieval ones, by showing how a small window size for a type with high property variety takes longer to stabilize, as seen in Figure 12.

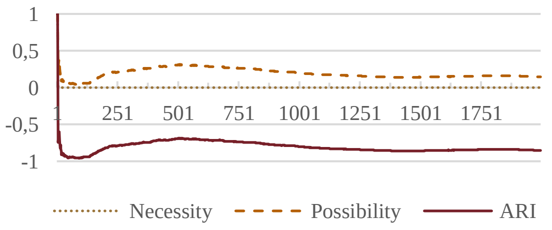


Figure 12 - Evolution of N, Π and ARI of IF for the property *addedBy* with w_s of 10

However, while there are significant improvements when w_s is increased from 10 to 50, the same cannot be said from increasing it from 50 to 100 (much like in the studies for the *Paper* class), as seen in Figure 13.

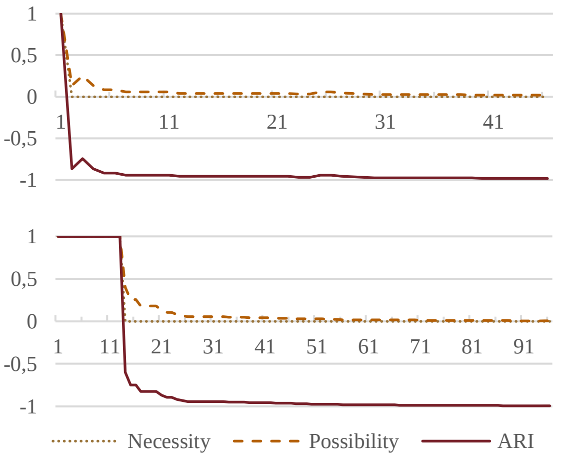


Figure 13 - Evolution of N, Π and ARI of IF for the property *addedBy* with w_s of 50 (first graph) and 100 (second graph)

In conclusion, in the cases where there is a very limited number of counterexamples, ARI may never reach its lower threshold; but remains reliably negative, even when f-measure is at its lowest – showing that the possibilistic approach is indeed robust and applicable when scoring axioms in streams and with limited data, and more so than a strictly probability-based one. It nonetheless benefits from increased number of counterexamples, implying that a w_s of 50 may provide sufficiently reliable results while also accounting for the negative effects of variations in property use.

Depending on the completion of the data, the incompleteness of the ontology, or simply because no such cases have ever been documented – the same sample can easily selectively confirm multiple property axioms. Consider, for example, that falsifying both F and IF axioms rely on the existence of more than one sample, or at least that the one sample uses the same property more than once. This is an unreasonable expectation to have about the data, and any decision to include the axioms needs to consider the fact that absence of evidence is not evidence of absence.

Since a bigger window size allows for better categorization of samples, it is possible to see the effect in evolution of IF's ARI for the functional property *hasAuthor*. Consider the difference between the graphs in Figure 14.

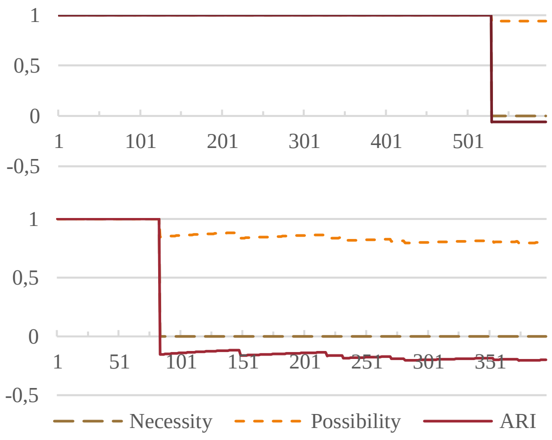


Figure 14 - Evolution of N, Π and ARI of IF for *hasAuthor* with w_s of 10 (first graph) and 50 (second graph)

Not only is the number of counterexamples very low, but they also take some time to occur in the stream. This means that while there are counterexamples, since there are so many sequential selective confirmations – and they continue even after a few counterexamples are encountered – N may drop to 0, and Π remains high to accommodate for potential errors in data. Therefore, while ARI skews negative, it remains relatively close to 0 (full uncertainty). However, it is once again interesting to note the benefits of the bigger window in the correct categorization of samples, as it allows for the negative ARI to be reached sooner (and more negative) as the window size increases.

Again, it is important to reiterate that this is a fortunate case: while the counterexamples may have taken longer to arrive and be few, they were still identifiable. When such is not the case, it has to be considered that just because a property is only used once per sample, it does not necessarily mean that it must certainly be *both* functional and inverse functional (although it is possible to be both at once) – even if their Π, N and ARI are always at 1.

4.2 Section II – Using the Weak Forms of Possibility and Necessity

Transitivity was studied using the Wine and Plant ontologies, with similar results. Since the Wine ontology had more available individuals with transitive properties (a total of 82), the following results reflect those exclusively.

Figure 15 shows how the lack of explicit confirmations affects the support for an axiom. It was necessary to analyse at least 30 individuals until a confirmation could be found, with the number increasing steadily until it reaches around 15%. Information retrieval metrics, as seen in Figure 16, also illustrate the clear lack of selective confirmations, which informed the decision to apply the weak forms of Π and N.

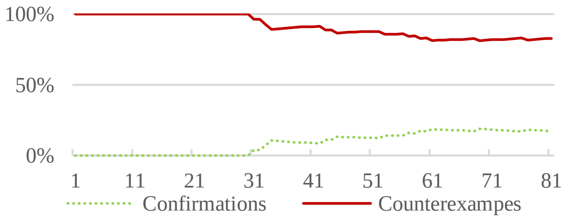


Figure 15 - Evolution of confirmations and counterexamples of T for the property *locatedIn* with w_s of 50

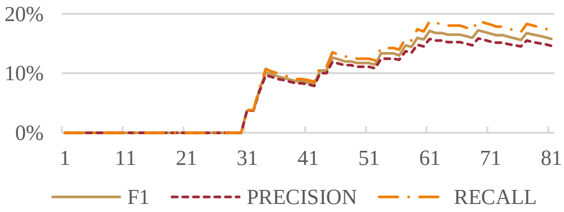


Figure 16 - Evolution of information retrieval metrics of T for the property *locatedIn* with w_s of 50

By giving more weight to the very hard to find selective confirmations than to the very easily incorrectly identified counterexamples, Figure 17 shows it is possible to obtain a positive, albeit conservative, ARI.

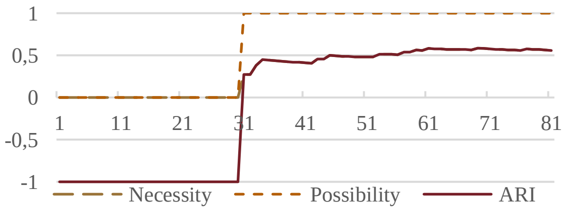


Figure 17 - Evolution of the weak forms of Π, N and ARI of T for the property *locatedIn* with w_s of 50

Should the strong form of possibility and necessity been applied instead, ARI would tend towards extremely negative (circa -1), even if the possibility was seen increasing (very) slowly.

4.3 Discussion

Of the three options in window size studied – of 10, 50 and 100 individuals – the results show that there is a significant improvement between going from 10 to 50, but hardly any from 50 to 100. We find 50 individuals to be a good middle ground; while the results can be improved by employing bigger sliding windows, they also validate our assumption that it is possible to efficiently learn property axioms from a relatively small number of samples.

The observation of the previous results show that the computation of ARI contains more information than that just the percentage of selective confirmations and could potentially be used in its place. Additionally, ARI shows better performance than traditional information retrieval metrics, achieving sharper results and considerably earlier. However, since the existence of a single counterexample immediately skews ARI towards negative, $\mathrm{E}^{+}_{\mathrm{Ax(P)}}$ cannot be altogether excluded, especially considering that potential errors in data could be classified as counterexamples. Furthermore, the results suggest that some axioms benefit from the application of a higher threshold for ARI than others.

For axioms that are relatively easy to falsify, but not so easy to prove, like F and IF, a higher threshold for ARI should allow for some errors in the data while still strongly advocating for the inclusion of said axioms. On a window size of 50 individuals, we argue that ARI should be of 1 for an axiom to be considered for inclusion, but have a proportion of $\mathrm{E}^{+}_{\mathrm{Ax(P)}}$ of at least 95%. Alternatively, for axioms for which there may be an overabundance of negatively-identified counterexamples and therefore make use of the weak forms of N and Π, a lower threshold for ARI should be considered: allowing an axiom to be considered for inclusion even though there is relatively little evidence for it. For these, we propose a threshold for ARI of around 0,5.

There is a second part to the discussion of the existence of counterexamples: axioms that are the opposite of one another. While a property can be both functional and inverse functional at the same time, it cannot be both symmetrical and asymmetrical. Furthermore, while a selective confirmation of S is a counterexample of AS, a selective confirmation of either does not necessarily mean that the property axiom must to be present; any decision-making processes need to take this in consideration. This is especially evident in the case of property *hasDecision* of the CMT ontology, which is functional, but shows N, Π and ARI of 1 for F, IF, AS and IR.

5. Experiment II: Accommodating for errors in data and the effects of previous knowledge in axiom-inclusion decisions

The following experiment pertains to show the applicability of the solution for the purposes of ontology enrichment/evolution. This is done by analysing a stream of individuals in which not only new properties are added over time, but the known properties are also applied in different ways to describe the data. As such, a suitable dataset needs to support the following: (1) have several properties that connect the individuals in diverse ways, such that all different property axioms being studied could be tentatively discovered, (2) the application of said properties should change over time, as the domain naturally evolves, (3) the property axioms are not previously known.

The goal of this experiment is to answer the following questions:

1. When combining ARI and Support, are any inconsistencies introduced? To which extent? Are the inconsistencies the result of incorrect data or marginal, but valid uses of the properties?
2. If previous knowledge is considered in the decision to include/exclude a property axiom, does it prevent or increase the amount of inconsistencies introduced?

The ontologies in use pertain to the Pokémon domain as it is described in Wikidata [33]. Pokémon is a series of games about cataloguing fictional wild creatures of the same name. Over the years, several games have been released, and each generation of games – a total of nine, at the time of writing, – introduces (and refines) mechanics, regions and creatures. Much of this information is publicly available on Wikidata. New subproperties of those present in Wikidata were created (but not characterized) that pertain to specific relationships in the Pokémon domain – Wikidata's properties, by design, are high level and lack the nuance necessary to describe specific domains beyond very simple connections between its individuals (e.g. part of, instance of).

The experiments detailed below show how axiom scoring can be used to determine which axioms could be associated with each property, and how their usage changes with each new generation of games. Table 10 shows the values and thresholds employed, and the application of both ARI and the percentage of selective confirmations (cf_t) for axiom scoring are described below. Each generation will be analysed as a different version of the ontology, provided by its own timeframe, and introductions and changes to its properties are documented. As previously established, an acceptation threshold of 0.5 will be used for both strong and weak forms of ARI (sf_t and wf_t, respectively) and a window size (w_s) of 50.

Table 10 - Values and thresholds employed in the experiments

Variable	Description	Value
cf_t	Percentage of selective confirmations w.r.t. support	95%
sf_t	Threshold for the Strong form of ARI	0.5
wf_t	Threshold for the Weak form of ARI	0.5
w_s	Sliding window size	50

5.1 Section I – ARI and Support

5.1.1 Generation I

Generation I (Gen I) features games with relatively simple mechanics. It introduces one region, one Pokédex (the Pokémon encyclopaedia) and 151 Pokémon. Using the data available on Wikidata, enough information was extracted to ascertain the properties described below. Table 11 shows the names, percentage of selective confirmations, proposed property axioms and ARIs for each.

Table 11 – Gen I properties

Property Name	Axiom	%Cf	ARI
bordersWith	T	51%	0.87
hasEvolutionGroup	AS	100%	1
	F	100%	1
hasMoveType	F	100%	1
	AS	100%	1
	IR	100%	1
hasPart	IF	100%	1
	AS	100%	1
hasPokedexEntry	F	100%	1
	IF	100%	1
	AS	100%	1
	IR	100%	1
introducedIn	F	100%	1
	AS	100%	1
	IR	100%	1
locatedIn	F	100%	1
	AS	100%	1
	IR	100%	1
partOf	F	100%	1
	AS	100%	1
	IR	100%	1
presentIn	F	100%	1
	AS	100%	1
	IR	100%	1
hasValue	F	100%	1
hasHeight	F	99%	-0.11

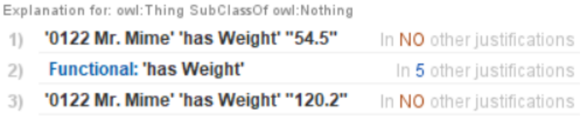
Property Name	Axiom	%Cf	ARI
hasWeight	F	96%	-0.28
hasFacet	F	100%	1
hasName	F	100%	1
hasPokedexNumber	F	100%	1
hasColor	F	99%	-0.16

bordersWith, a property that establishes a connection between two locations (e.g. cities or roads) that share a border, seems a target candidate for S, but the results do not support it (selective confirmations amounted to 1%). Interestingly, the results show there is sufficient positive evidence for T, and adding this axiom to the ontology does not make it inconsistent – although including it would allow for entailing incorrect conclusions about the data. Counterexamples were found for AS and IR. *locatedIn* is another property related to the geography of the region. However, since only one region has been introduced as of Gen I, it is classified as F, with no contradictions on the data.

With each Pokémon belonging to a single evolution group (described by the *hasEvolutionGroup* property) but each group having more than one creature, the expected axiom for this property would be F, which the results support.

Of the datatype properties, three of them were classified as F even though counterexamples were found – since the percentage of selective confirmations was considered sufficient, and the deviations should pertain to possible errors in the data. In this case, we can verify if this was the cause, by using the reasoners provided by Protégé (in this case, HermiT[1]) and adding said axioms to the ontology and analysing the explanations provided in case inconsistencies are found.

Figure 18 shows the inconsistency explanations for the *hasWeight* property, in which it is possible to see there are 6 individuals with more than one entry, and the duplicates' values seem to correspond to the same value under different representation systems (metric vs imperial), which suggest that using the same property to represent both may not be adequate.

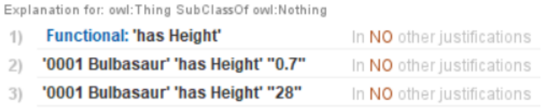

Explanation for: owl:Thing SubClassOf owl:Nothing

1) '0122 Mr. Mime' 'has Weight' "54.5" In NO other justifications

2) Functional: 'has Weight' In 5 other justifications

3) '0122 Mr. Mime' 'has Weight' "120.2" In NO other justifications

Figure 18 - Reasoner's explanation for inconsistency for property *hasWeight*


Explanation for: owl:Thing SubClassOf owl:Nothing

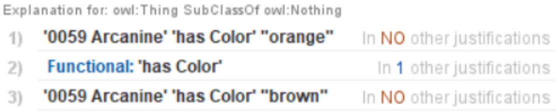
1) Functional: 'has Height' In NO other justifications

2) '0001 Bulbasaur' 'has Height' "0.7" In NO other justifications

3) '0001 Bulbasaur' 'has Height' "28" In NO other justifications

Figure 19 - Reasoner's explanation for inconsistency in property *hasHeight*

Figure 19 shows how there is one single individual that contradicts the functionality of the *hasHeight* property, with the same apparent justification as the *hasWeight*.


Explanation for: owl:Thing SubClassOf owl:Nothing

1) '0059 Arcanine' 'has Color' "orange" In NO other justifications

2) Functional: 'has Color' In 1 other justifications

3) '0059 Arcanine' 'has Color' "brown" In NO other justifications

Figure 20 - Reasoner's explanation for inconsistency in property *hasColor*

Figure 20, on the other hand, shows there are 2 individuals with more than one colour, both belonging to the same evolutionary group. This may be an oversight in the data acquisition from Wikidata, which often mixes the information of all generations.

5.1.2 Generation II

Generation II (Gen II) improves on Gen I by introducing a new region (adjacent to the first one), while still allowing the player to visit the one introduced in Gen I. The Pokédex is expanded to accommodate for the new region: the player now effectively can access not one, but two Pokédexes, one at the national level (with all creatures) and a regional one (with the creatures inhabiting the new region exclusively, which may or may not be new). Therefore, the same creature may now be associated with more than one Pokédex entry, but each entry will be associated to its own numbering system (and potentially, description). Information about the creature's shape is also added, and the number of Pokémon grows from 151 to 251. Additionally, some new game mechanics are included: creatures can now have one of three genders (female, male, and unknown), and can reproduce within a given group (not necessarily only with members of the same species).

Information about the properties present in Gen II is displayed in Table 12. For the sake of brevity, only changes in axioms are shown. If an axiom is removed, it is preceded by a – symbol, and by a + otherwise.

[1] http://www.hermit-reasoner.com/

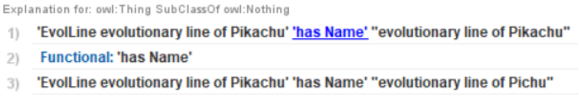
Table 12 – Gen II properties

Property	Axiom	%Cf	ARI
bordersWith	+T	65%	0.94
hasEvolutionGroup	+F	100%	1
hasMoveType	+F	100%	1
hasPokedexEntry	–F	0%	-1
presentIn	–F	37%	-0.93
alternateDexEntry	+F	50%	0.86
	+IF	100%	1
	+T	50%	0.87
	+S	50%	0.87
hasGenderRatio	+F	100%	1
	+AS	100%	1
	+IR	100%	1
hatches	+T	25%	0.66
hasShape	N/A	N/A	N/A

hasPokedexEntry, which relates a Pokémon to its corresponding Pokédex information, loses the F axiom – as the new region introduced a new Pokédex, and therefore a Pokémon may have more than one entry. However, it retains the IF axiom, as each entry relates to a single creature. *presentIn*, a property which describes in which generation a given entity or mechanic is featured, can now point to more than one option and, therefore, is no longer F. Of the new properties, *alternateDexEntry* connects any two entries in different Pokédexes that describe the same creature and therefore should be classified at least as S. The results show not only this happens, but it also does not contradict the T, F and IF property axioms.

With the introduction of genders, each species displays one of several possible gender ratios (via the *hasGenderRatio* property), and as such has been classified as F. As no evidence was found against it, it is also classified as AS and IR. *hatches*, which relates a creature with its reproductive group, and each reproductive group with the Pokémon in it, has sufficient ARI to be classified as T, but not S. Finally, the only datatype property added in Gen II, *hasShape*, does not meet the criteria for any of the property axioms.

When adding the proposed axioms to the object properties in the ontology of Gen II, no inconsistencies are generated. However, the same cannot be said for the datatype properties. According to Figure 21, there are some inconsistencies regarding the use of the *hasName* property, namely when describing the evolutionary groups. Because new Pokémon were introduced in between generations and added to existing evolutionary groups, some of their names to have undergone changes that have not been corrected.

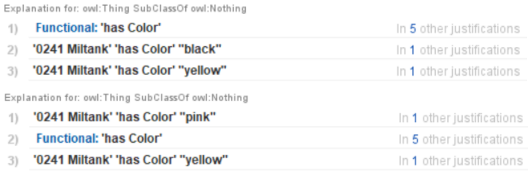


Explanation for: owl:Thing SubClassOf owl:Nothing

1) 'EvolLine evolutionary line of Pikachu' 'has Name' "evolutionary line of Pikachu"
2) Functional: 'has Name'
3) 'EvolLine evolutionary line of Pikachu' 'has Name' "evolutionary line of Pichu"

Figure 21 - Inconsistencies with the use of the *hasName* property

As seen in Figure 22, in addition and similar to the inconsistencies present in Gen I, there are a few (six) entities with two or more uses of the property *hasColor*.



Explanation for: owl:Thing SubClassOf owl:Nothing

1) Functional: 'has Color' — In 5 other justifications
2) '0241 Miltank' 'has Color' "black" — In 1 other justifications
3) '0241 Miltank' 'has Color' "yellow" — In 1 other justifications

Explanation for: owl:Thing SubClassOf owl:Nothing

1) '0241 Miltank' 'has Color' "pink" — In 1 other justifications
2) Functional: 'has Color' — In 5 other justifications
3) '0241 Miltank' 'has Color' "yellow" — In 1 other justifications

Figure 22 - Inconsistencies with the use of the *hasColor* property

5.1.3 Generation III

Generation III (Gen III) introduces two more regions, and no longer features those of Gens I and II. It also introduces two new mechanics: abilities and contests. Each Pokémon can now have one or two of the 77 possible abilities – some of which are considered “signature abilities”, as they are only shown for specific Pokémon or specific evolutionary groups. Furthermore, 135 new Pokémon are added (to a total of 386). The changes and additions to the properties and their axioms are shown in Table 13.

Table 13 – Gen III properties

Property	Axiom	%Cf	ARI
hasMoveType	–F	0%	-1
alternateDexEntry	–F	65%	-0.76
	–IF	53%	-0.9
	–T	35%	0.76
hasSignatureAbility	+F	100%	1
	+IF	100%	1
	+AS	100%	1
	+IR	100%	1
isSignatureAbilityOf	+F	100%	1
	+IF	100%	1
	+AS	100%	1
	+IR	100%	1

With the introduction of contests, moves now have no longer only a type in battle, but a type that shows in contest (the two are not related). As such, the property *hasMoveType* is no longer compatible with functionality. Once again, with the introduction of a new region and a new Pokédex, the same Pokémon can have multiple entries – and although these are still alternatives to each other, and therefore symmetrical, there is now more than one possible alternative and the property can no longer be classified as either F or IF.

Finally, *hatches* maintains its T. Evidence against IR was below the minimum threshold (selective confirmations amounting to 95% of the data) – but as it was already classified as such, the axiom is not added. Following the definition of a signature ability discussed above, the property is properly classified as F (each Pokémon/evolutionary group has only one signature ability) and IF (each signature ability is used by only one Pokémon/evolutionary group). Since no counterexamples for IR and AS are found, these axioms are also added to the property. As the percentage of counterexamples found for F in *hasWeight* has once again lowered below the minimum threshold, the property axiom is reinstated.

As with Gen II, when the axioms discovered are added to object properties in the ontology for Gen III, they do not produce inconsistencies. However, with support for *hasColor* at 98% and *hasHeight* as 99%, counterexamples are rare but produce some inconsistencies.

5.1.4 *Generations IV and V*

Generation IV (Gen IV) introduces once again a new region and, with it, comes a new catalogue and new Pokémon (totalling 493). There is also the introduction of 47 new abilities and 113 new moves. Several evolutionary groups from Gen I were expanded with new elements, and moves are now classified not only according to their previously known contest and type categories, but with an additional damage category that is fully independent from its type. It is also in this generation that the games make use of alternate forms of the same Pokémon (including, but not limited to, differences by gender). Generation V (Gen V) raises the total number of Pokémon to 649, and once again introduces a new region while limiting access to the previous ones, but does not introduce any new properties.

Changes and additions to the properties of the ontology for Gen IV are displayed in Table 14:

Table 14 – Gen IV properties

Property	Axiom	%Cf	ARI
alternateDexEntry	+T	47%	0.85
hasAlternateForm	+T	62%	0.92
	+S	27%	0.69

The inclusion of these axioms in both Gen IV and Gen V ontologies does not cause inconsistencies beyond those already discovered in previous generations.

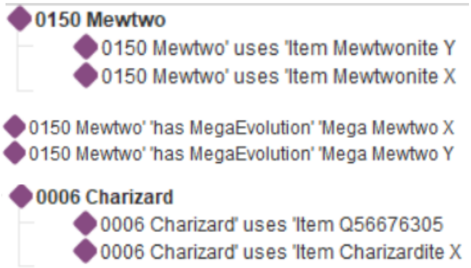
5.1.5 *Generation VI*

Generation VI (Gen VI) introduces 72 new Pokémon (to a total of 721), 58 new moves and 24 new abilities (to a total of 617 and 188, respectively). It also adds a new battle mechanic, the Mega Evolution, which is a type of alternative form that can be (temporarily) triggered in battle. Like previous generations, Gen VI introduces a new region and a new Pokédex, while also revisiting the region first introduced in Gen III. Table 15 shows the changes in the property axioms in Gen IV:

Table 15 – Gen VI properties

Property	Axiom	%Cf	ARI
hasMegaEvolution	+F	98%	-0.21
	+IF	100%	1
	+AS	100%	1
	+IR	100%	1
uses	+F	96%	-0.23
	+IF	100%	1
	+AS	100%	1
	+IR	100%	1

hasMegaEvolution, a relationship between a Pokémon and its Mega Evolution alternate form, is both F and IF (theoretically meaning that a Pokémon can only have one mega evolution and that evolution belongs to only one Pokémon). The mechanic is triggered by the usage (with the *uses* property) of different battle items, and each of them causes a specific evolution to occur.



0150 Mewtwo
0150 Mewtwo' uses 'Item Mewtwonite Y
0150 Mewtwo' uses 'Item Mewtwonite X

0150 Mewtwo' 'has MegaEvolution' 'Mega Mewtwo X
0150 Mewtwo' 'has MegaEvolution' 'Mega Mewtwo Y

0006 Charizard
0006 Charizard' uses 'Item Q56676305
0006 Charizard' uses 'Item Charizardite X

Figure 23 – Rare, but valid individuals which do not support F for the *hasMegaEvolution* and *uses* properties

When the axioms are added to the Gen VI ontology, they do not produce inconsistencies – contrary to what would be expected when the percentage of confirmations is below 100%. However, upon closer analysis, it is possible to see that there are indeed few, but valid, cases in which a Pokémon can have more than one mega evolution, caused by the application of more than one item. In the dataset for Gen VI, there are two such occurrences, shown in Figure 23. The reasoner fails to flag these as inconsistencies unless the individuals are explicitly stated as being different (which would always be the case under the UNA).

5.1.6 Generation VII

Generation VII (Gen VII) sees the introduction of regional forms, as the same creature adapts to different habitats: effectively another specific type of alternate form. It also increases the number of Pokémon to 802 and introduces a new region and its respective Pokédex. It revisits the region introduced in Gen I, adding new alternate forms to some previously known Pokémon.

Table 16 – Gen VII properties

Property	**Axiom**	**%Cf**	**ARI**
hasRegionalForm	+F	97%	-0.24
	+IF	98%	-0.19
hasSignatureAbility	–F	94%	-0.33
isSignatureAbilityOf	–IF	95%	-0.32
uses	–IF	71%	-0.7

With the introduction of new abilities, Gen VII alters how signature abilities work, and a few Pokémon can now have more than one signature ability – as such, *hasSignatureAbility* can no longer be F, and *isSignatureAbilityOf* can no longer be IF. Thankfully, in this case, the number of counterexamples is above the threshold and the axioms are correctly removed. If the axioms are added to the ontology, by UNA they generate some inconsistencies. Figure 24 shows one such case, in which a valid selective confirmation of a creature has more than one known regional variant.

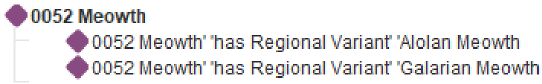

0052 Meowth
0052 Meowth' 'has Regional Variant' 'Alolan Meowth
0052 Meowth' 'has Regional Variant' 'Galarian Meowth

Figure 24 – Rare, but valid individuals which do not support the F and IF for the *hasRegionalForm* property

5.1.7 Generations VIII and IX

Generation VIII (Gen VIII) introduces two new regions, each with its individual Pokédex. The national one, which catalogues all creatures, now goes up to 890, with 19 new regional forms. The mega evolution mechanic is removed from the games. Generation IX (Gen IX) introduces another region and its respective catalogue, and 103 new creatures (raising the total to 1008). While it introduces the concept of convergent evolution – creatures that fill the same ecological niches also sharing physical similarities – information about this was not present in Wikidata at this time. Finally, this generation introduces a few more regional forms and a new type of alternative form that is not well described in Wikidata.

Because of alterations on how signature abilities are assigned to evolutionary groups between games, Table 17 shows the changes in the associated properties.

Table 17 – Gen VIII and Gen IX properties

Property	**Axiom**	**%Cf**	**ARI**
hasSignatureAbility	+F	96%	-0.28
isSignatureAbilityOf	+IF	96%	-0.27

Once again, this is a case in which the inconsistencies refer to valid uses of the property, meaning the application of the axioms is in the wrong, as shown in Figure 25:

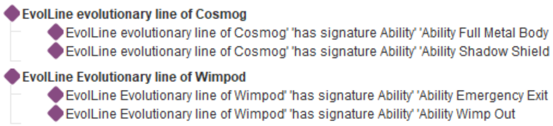

EvolLine evolutionary line of Cosmog
EvolLine evolutionary line of Cosmog' 'has signature Ability' 'Ability Full Metal Body
EvolLine evolutionary line of Cosmog' 'has signature Ability' 'Ability Shadow Shield
EvolLine Evolutionary line of Wimpod
EvolLine Evolutionary line of Wimpod' 'has signature Ability' 'Ability Emergency Exit
EvolLine Evolutionary line of Wimpod' 'has signature Ability' 'Ability Wimp Out

Figure 25 – Evolutionary line with more than one signature ability, a counterexample to the F of *hasSignatureAbility*

5.2 Section II - Evolving ARI

Experiments show that allowing for some leeway in terms of inconsistency can result in the discovery of errors in the data, such as duplicates, or potential mistakes in modelling that do not account for the possible alternatives. It also shows that not all inconsistencies are caused by said incorrections, and some valid, but outlier information can get flagged as inconsistent.

Evolving ARI depends not on a minimum threshold on the percentage of selective confirmations, cf_t, but on the evidence found in the current timeframe and the conclusions obtained on the previous one (which may be a state of total ignorance, if the property was not present).

The following experiments are divided in two different sections, namely:

1. Applicability and robustness of ARI for axiom scoring. Considering the dataset was obtained from Wikidata, which is often incomplete and sometimes offers incorrect or even contradictory information, we consider the percentage of selective confirmations does not need to equal 100% for the axiom to be accepted. Furthermore,

in this case, the analysis is done from a point of complete ignorance, in which no previous versions of any axiom are considered. This should allow for a more informed decision about which thresholds to consider for axiom inclusion in the following experiments.

2. Analysis of ARI_e over several different timeframes, in which previous versions of the ontology affect the scoring of the new axioms. Using the previously defined thresholds for axiom inclusion and window size, it is possible to measure the effect of more conservative vs progressive approaches to knowledge evolution.

First, we must consider how conservative the approach will be by defining the relative weight of previous knowledge, w_p. Then, we must establish a threshold for inclusion or rejection of the hypothesis considering the computed ARI_e (sf_t or wf_t, depending on which form was applied). These two values cannot be completely independent of one another, since we are considering only scalar values for the previous state and it is possible for new data to directly contradict said state. Should the threshold for inclusion be too high, changing the axiom between timeframes could become impossible. For such changes to be a possibility, sf_t , wf_t and w_p should be inversely proportional. Since ARI cannot go over 1, for ARI_e to allow for evolution, sj_t should be below or equal to 0.5. Using ARI_e also allows for some data in a timeframe to contradict the axiom that is asserted in that period.

Table 18 shows the thresholds employed for the following experiments – maintaining the acceptance thresholds for the strong and weak forms of ARI (sf_t and wf_t, respectively) and a window size (w_s) of 50 –, in which different weights of previous knowledge for ARI_e (w_p) will be compared.

Table 18 - Values and thresholds employed in the experiments

Variable	Value
sf_t, wf_t	0.5
w_p	0.8 / 0.7 / 0.4
w_s	50

ARI_e depends on the results obtained in previous iterations, so its application can be measured from Gen II onwards. When previous decisions (valued 0 if not present, and 1 for present) about the property axioms are not known, an ARI_e of 0 is assumed, and the decision for each previous non-existent axiom is scored at 0.5, both to reflect a state of ignorance. The rest of this section presents and discusses the cases in which the decision supported by ARI_e is different than when using the combination of cf_t and ARI.

First of all, it is interesting to see the effects of the different weights affect the evolution of ARI_e. Figure 26 shows how a more conservative approach tends to favour whichever initial decisions were taken, while lower values require less evidence for a decision to be revoked. In this case, the conservative approach would be wrong, as classifying *hasAlternateDexEntry* as F would generate a large number of inconsistencies that would only grow with each generation.

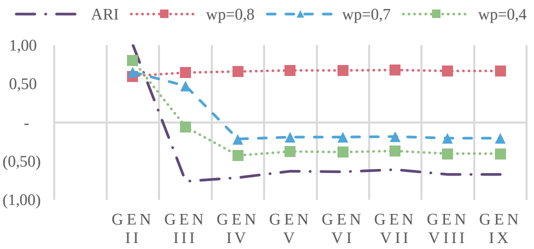


Figure 26 – Evolution of ARI_e for *hasAlternateDexEntry*'s F with different relative weights

Starting with the datatype properties, which were the ones creating the most inconsistencies since early generations, we can see that without taking cf_t in consideration, the duplicates are taken in consideration and the decision, across all generations, is not to include the F axiom. Figure 27 shows the results for the *hasColor* property, although they are similar for the others previously mentioned (*hasWeight* and *hasHeight*). This effectively means that F can no longer be employed to identify errors in data that resulted in the inconsistencies described above.

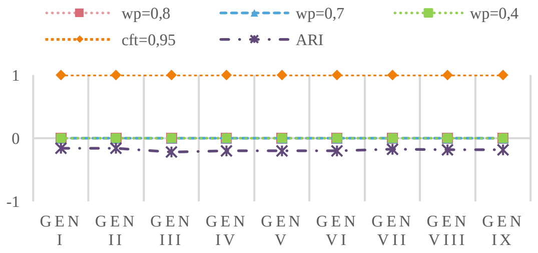


Figure 27 - Comparison of the decisions made using ARI_e and ARI+cf_t for F in *hasColor*

Differences in the decisions reached ARI, ARI+cf_t and ARI_e are shown by the *hatches* property, as seen in Figure 28 for T.

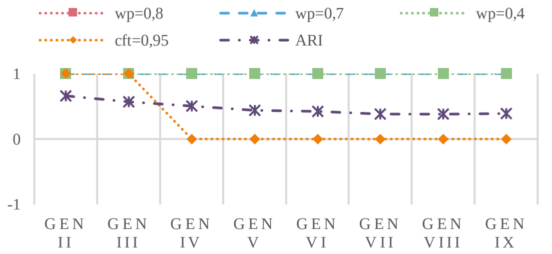


Figure 28 – Decisions regarding the T of the *hatches* property

With the more conservative approach to change provided by ARI_e, the property is equally classified as T for the first two generations, but manages to maintain said status afterwards, despite the gradual decrease in ARI. In this case, adding the axioms to any of the versions of the ontology does not generate any inconsistencies.

In some generations, *isSignatureAbilityOf* and *hasSignatureAbility* were shown to have valid counterexamples that were not considered because of the chosen value for cf_t. Figure 29 shows the effect of the different w_p in the decision-making process. The results show that employing lower w_p results consistently in a Functional *hasSignatureAbility*, which is known to produce some inconsistencies from Gen VIII onwards. The opposite happens with heavier w_p: by making it harder to change opinion, even though ARI_e gets slightly higher – and in some cases, even positive – values than ARI, the evidence is not sufficient for a change. Figure 30 shows similar trends for IF of *isSignatureAbility* property.

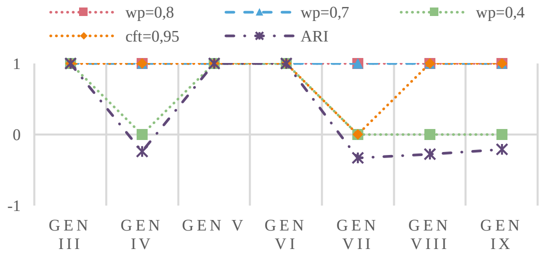


Figure 29 – Comparison of the decisions to include/exclude the F axiom for the *hasSignatureAbility* property

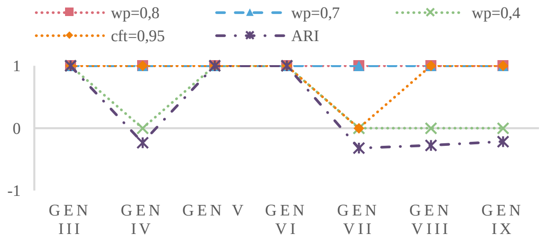


Figure 30 - Comparison of the decisions to include/exclude the IF axiom for the *isSignatureAbilityOf* property

hasRegionalForm, which was considered F, IF and S, is now instead classified as T and S for all generations in which it is featured. This happens because ARI_e cannot sustain F, as seen in Figure 31.

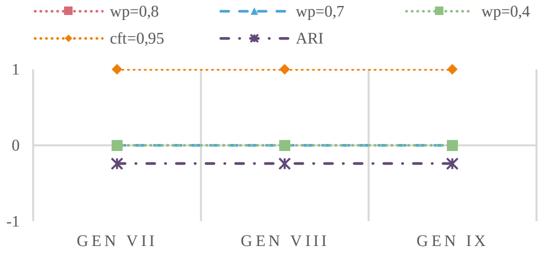


Figure 31 - Comparison of the decisions to include/exclude the F axiom for the *hasRegionalForm* property

Since F is no longer associated with the property, it can assume the T axiom, for which it did have sufficient ARI and ARI_e, as Figure 32 shows:

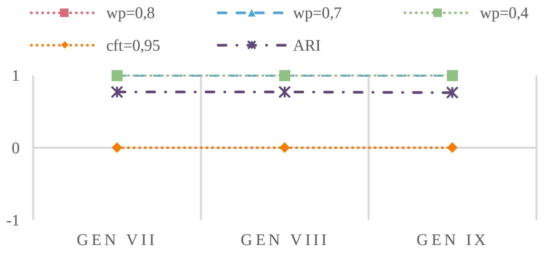


Figure 32 - Comparison of the decisions to include/exclude the T axiom for the *hasRegionalForm* property

The inclusion of the T and S axioms to the ontologies of Gens VIII and IX does not produce inconsistencies.

5.3 Discussion

Using ARI by itself does not allow for inconsistencies to arise, which in turn means that in spite of its utility for axiom scoring, it does not allow for the very likely possibility of errors and inconsistencies in the data – which is especially relevant when developing an ontology from publicly-available information from sources such as Wikidata. By combining ARI and cf_t, we allow for some level of counterexamples to be possible while still accepting an axiom hypothesis. In these cases, only the analysis of the data can show if the decision is correct or incorrect – for example, by investigating any generated inconsistencies to ascertain if they effectively correspond to errors, mistakes, or valid information. Nonetheless, we find the approach helpful for deciding on the inclusion of property axioms on evolving ontologies.

The results show that is still possible to counter some errors in data when using ARI_e, but not as effectively as the combination of ARI and cf_t. This is because, by favouring previous knowledge, ARI_e may have difficulty changing its opinion even in the face of overwhelming evidence. On some occasions, this may be beneficial: the experiments show that, when applied for T and S – the two property axioms that use the weak-form of ARI because of their inherent missing information – their inclusion is favoured earlier, and is harder to dismiss, with no inconsistencies resulting from it. However, for the properties using the strong form, the effect is diametrically opposite: axioms are maintained, even though they may introduce an increasing number of inconsistencies over time. Finding a balance between these two outcomes may be possible by tinkering with different values for sf_t and wf_t.

6. Conclusions

This paper presented an adaptation of the possibilistic approach to axiom scoring to the context of RDF data streams for ontology evolution. The different approaches to possibility and necessity proposed in literature were recontextualized in terms of their bias towards selective confirmations or counterexamples, and the assumptions regarding the openness of the world under which they operate. Some axioms, namely transitivity and symmetry, benefit from a more lenient approach, relying more on selective confirmations than on counterexamples – while the others benefit from stricter acceptance conditions to prevent the proliferation of inconsistencies.

To test the applicability of the solution, it was applied in two distinct scenarios: (1) a first one, in which the property axioms were previously known, and which allowed for the exploration of the effectiveness of the approach for their discovery in a scenario where no incorrect data was present; and (2) a second one, in which the neither the properties nor their axioms were known, and the dataset was obtained from publicly available sources, possibly both incomplete and with errors.

Regarding Experiment I, results show that possibilistic approach is well suited to suggest potential axioms for ontology properties in an instance-guided ontology evolution scenario, achieving conclusions about inclusion/exclusion of axioms from a relatively small sample of individuals (roughly 2,3%), and substantially faster than using traditional information retrieval metrics – making it particularly suitable for quickly learning new axioms from streams of RDF data. This, however, is not achieved without some caveats: the approach is not sufficiently robust to cases in which there are inconsistences or errors in data, with the strong approach being particularly unable to recover from counterexamples that may reflect said errors. No approach can effectively deal with all potential negative side effects of dealing with an open world and with incomplete knowledge; if ARI is used independently of support, it cannot account for missing or incorrect information.

Additionally, this experiment shows that the size and variety of the individuals in the dataset affect the speed and accuracy of the identification of axioms, suggesting that some finetuning of parameters may be necessary to achieve the best results depending on the application scenario.

Experiment II aimed to overcome the issue with ARI identified in Experiment I in two ways: (1) by combining it with a minimum percentage of selective confirmations and (2) by attributing weight to previous knowledge between timeframes. Axioms that are not 100% supported by ARI may therefore be accepted: this allows for the identification of potential errors in data but may also erroneously suggest discarding valid information. Acknowledging the information provided by previous versions of the ontology when deciding for or against an axiom is helpful in identifying and maintaining property axioms for which positive evidence is inherently harder to find – however, it also allows for the continued integration of incorrect axioms. Here, the combination of support and ARI seems to achieve the better results when it comes to introducing the least amount of inconsistencies over time; the downside being that it may erroneously consider valid, but marginal uses of properties as irrelevant, and therefore the results still need to be analysed on a case-by-case basis. Ultimately, the experimental results show that ARI can be made more resistant to errors in data, benefiting from being combined with other metrics – but more work is needed in this regard to further explore which metrics – and in which way – can be combined in a more systematic fashion that is less dependent of empiric analysis of specific use-cases.

Finally, and although it was out of the scope of the experiments performed for this work, it is also important to consider the possibility of over-characterization of the properties in an ontology. The results show that the current solution will almost always propose one or more axioms for each property – which may or may not make sense, depending on the application context and purpose of the ontology; upper-level ontologies, by definition and application, benefit from excluding superfluous axioms. Even for lower-level ontologies, there is always an argument to be done in favour of simplicity of design, which should not be more complex than necessary to achieve its goals. The suitability of the suggested axioms, therefore, should not rely solely on the fact that their addition to the ontology does not introduce inconsistencies, but be motived too by their capacity to allow for richer inference processes to happen in order to unlock the data's true potential.

Acknowledgements

This work has been supported by the European Union under the Next Generation EU, through a grant of the Portuguese Republic's Recovery and Resilience Plan Partnership Agreement under the project PRODUTECH R3. It has received Portuguese National Funds through Portuguese Foundation for Science and Technology under the project UIDP/00760/2020 and the Ph.D scholarship with reference SFRH/BD/147386/2019. Additionally, this work benefited from the informed and pertinent insights by INRIA's mOeX team and the efforts made by Wikidata's Project Pokémon[2].

References

[1] A. Polleres, R. Pernisch, A. Bonifati, D. Dell'Aglio, D. Dobriy, S. Dumbrava, L. Etcheverry, N. Ferranti, K. Hose, E. Jiménez-Ruiz, M. Lissandrini, A. Scherp, R. Tommasini, and J. Wachs, How Does Knowledge Evolve in Open Knowledge Graphs?, *Transactions on Graph Data and Knowledge (TGDK)*. **1** (2023) 1–59. doi:10.4230/TGDK.1.1.11.

[2] A.C. Khadir, H. Aliane, and A. Guessoum, Ontology learning: Grand tour and challenges, *Comput Sci Rev*. **39** (2021). doi:10.1016/J.COSREV.2020.100339.

[3] X. Su, E. Gilman, P. Wetz, J. Riekki, Y. Zuo, and T. Leppänen, Stream reasoning for the internet of things: Challenges and gap analysis, *ACM International Conference Proceeding Series*. **13-15-June-2016** (2016). doi:10.1145/2912845.2912853.

[4] P. Esling, and C. Agon, Time-series data mining, *ACM Comput Surv*. **45** (2012) 12. doi:10.1145/2379776.2379788.

[5] J.D. Hamilton, Time series analysis, Princeton University Press, n.d.

[6] A. Canito, J. Corchado, and G. Marreiros, A systematic review on time-constrained ontology evolution in predictive maintenance, *Artif Intell Rev*. **55** (2022) 3183–3211. doi:10.1007/S10462-021-10079-Z.

[7] A. Canito, A. Nobre, J. Neves, J. Corchado, and G. Marreiros, Using sensor data to detect time-constraints in ontology evolution, *Integr Comput Aided Eng*. **30** (2023) 169–184. doi:10.3233/ICA-230703.

[8] P. Burek, N. Scherf, and H. Herre, Ontology patterns for the representation of quality changes of cells in time, *J Biomed Semantics*. **10** (2019). doi:10.1186/s13326-019-0206-4.

[9] A.G.B. Tettamanzi, C. Faron-Zucker, and F. Gandon, Possibilistic testing of OWL axioms against RDF data, *International Journal of Approximate Reasoning*. **91** (2017) 114–130. doi:10.1016/j.ijar.2017.08.012.

[10] R. Peixoto, C. Cruz, and N. Silva, Semantic HMC: Ontology-Described Hierarchy Maintenance in Big Data Context, in: On the Move to Meaningful Internet Systems: OTM 2015 Workshops, Springer, 2015: pp. 492–501. doi:10.1007/978-3-319-26138-6_53.

[11] H. Kondylakis, and N. Papadakis, EvoRDF: evolving the exploration of ontology evolution, *Knowl Eng Rev*. **33** (2018). doi:10.1017/S0269888918000140.

[12] S. Benomrane, Z. Sellami, and M. Ben Ayed, Evolving ontologies using an adaptive multi-agent system based on ontologist-feedback, in: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), 2016: pp. 1–10. doi:10.1109/RCIS.2016.7549292.

[13] S.D. Cardoso, C. Pruski, and M. Da Silveira, Supporting biomedical ontology evolution by identifying outdated concepts and the required type of change, *J Biomed Inform*. **87** (2018) 1–11. doi:10.1016/j.jbi.2018.08.013.

[14] S. Benomrane, Z. Sellami, and M. Ben Ayed, An ontologist feedback driven ontology evolution with an adaptive multi-agent system, *Advanced Engineering Informatics*. **30** (2016) 337–353. doi:10.1016/j.aei.2016.05.002.

[15] A.E. Cano-Basave, F. Osborne, and A.A. Salatino, Ontology Forecasting in Scientific Literature: Semantic Concepts Prediction Based on Innovation-Adoption Priors, in: Knowledge Engineering and Knowledge Management (EKAW 2016), Springer, 2016: pp. 51–67. doi:10.1007/978-3-319-49004-5_4.

[16] S. Ghidalia, O.L. Narsis, A. Bertaux, and C. Nicolle, Combining Machine Learning and Ontology: A Systematic Literature Review, (2024). doi:10.48550/arXiv.2401.07744.

[17] T.H. Nguyen, Mining the semantic Web for OWL axioms, Université Côte d'Azur, 2021.

[18] F.N. Al-Aswadi, H.Y. Chan, and K.H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, *Artif Intell Rev*. **53** (2020) 3901–3928. doi:10.1007/S10462-019-09782-9.

[2] https://www.wikidata.org/wiki/Wikidata: WikiProject_ Pokémon

[19] S.H. Venu, V. Mohan, K. Urkalan, and T. Geetha V, Unsupervised Domain Ontology Learning from Text, in: Mining Intelligence and Knowledge Exploration (MIKE 2016), Springer, 2017: pp. 132–143. doi:10.1007/978-3-319-58130-9_13.

[20] K. Belhoucine, and M. Mourchid, A Survey of Ontology Learning from Text, in: Twelfth International Conference on Advances in Semantic Processing (SEMAPRO 2018), 2018: pp. 14–21.

[21] P. Buitelaar, P. Cimiano, and B. Magnini, Ontology Learning from Text: An Overview, (2005). https://api.semanticscholar.org/CorpusID:18638602 (accessed February 26, 2024).

[22] I. Scheffler, and N. Goodman, Selective confirmation and the ravens: A reply to Foster, *J Philos*. **69** (1972) 78. doi:10.2307/2024647.

[23] L. Bühmann, and J. Lehmann, Universal OWL axiom enrichment for large knowledge bases, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. **7603 LNAI** (2012) 57–71. doi:10.1007/978-3-642-33876-2_8.

[24] F. Sais, C. Pruski, and M. Da Silveira, Inferring the evolution of ontology axioms from RDF data dynamics, *International Conference on Knowledge Capture (K-CAP 2017)*. (2017). doi:10.1145/3148011.3154472.

[25] D. Malchiodi, and A.G.B. Tettamanzi, Predicting the possibilistic score of OWL axioms through modified support vector clustering, *Proceedings of the ACM Symposium on Applied Computing*. (2018) 1984–1991. doi:10.1145/3167132.3167345.

[26] T.H. Nguyen, and A.G.B. Tettamanzi, An evolutionary approach to class disjointness axiom discovery, *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2019)*. (2019) 68–75. doi:10.1145/3350546.3352502.

[27] A.G.B. Tettamanzi, C.F. Zucker, and F. Gandon, Dynamically time-capped possibilistic testing of SubClassOf axioms against RDF data to enrich schemas, *Proceedings of the 8th International Conference on Knowledge Capture (K-CAP 2015)*. (2015). doi:10.1145/2815833.2815835.

[28] A.G.B. Tettamanzi, C. Faron-zucker, and F. Gandon, Testing owl axioms against RDF facts: A possibilistic approach, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. **8876** (2014) 519–530. doi:10.1007/978-3-319-13704-9_39.

[29] Ontology Alignment Evaluation Initiative 2023, (2023). http://oaei.ontologymatching.org/2023/ (accessed February 26, 2024).

[30] How does it work? (OWL Example Wine Agent), (n.d.). http://ksl.stanford.edu/projects/wine/explanation.html#ontology (accessed March 8, 2024).

[31] Oregon State University, Plant Ontology, (n.d.). https://planteome.org/ (accessed March 8, 2024).

[32] R. Bruskiewich, E.H. Coe, P. Jaiswal, S. McCouch, M. Polacco, L. Stein, L. Vincent, and D. Ware, The Plant Ontology^{TM} Consortium and Plant Ontologies, *Comp Funct Genomics*. **3** (2002) 137–142. doi:10.1002/CFG.154.

[33] Wikidata. https://www.wikidata.org/wiki/Wikidata (accessed April 8, 2024).