Applying possibilistic axiom scoring to instance-guided ontology evolution in RDF streams

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 to complete and large datasets in ontology learning. This paper presented 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 a knowledge evolving 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 were recontextualized in terms of their bias towards 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.

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. If sufficient difference between the individuals of the concepts being analysed and the version of them asserted in the ontology is identified, TICO uses a 4-D Fluents [6] approach to reify new, disjoint definitions of the concept for each new timeframe – or Time Slice – in a strictly positive monotonic fashion. The general architecture of TICO has been described in [7]. 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. For the purposes of this work, the authors will focus on the discovery of a particular set of OWL axioms, namely those concerning the characteristics of properties. To do so, axiom testing analysis from both statistical and possibilistic perspectives will be executed.

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 [8] 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 different sliding windows sizes when searching for potential axioms in RDF data streams.
2. Using the results of the previous point to establish thresholds for axiom inclusion/exclusion in an ontology learning and evolution scenario, testing the approach against an ontology generated from publicly available data, for which no information about property characteristics is known.

The rest of the paper is organized as follows: (2) the Background section, which contextualizes the work in the field of ontology learning/evolution and describes existing works on axiom scoring, (3) Property Constructors, which goes in detail about the definitions of the axioms pertaining to each of the characteristics properties can use in OWL and how to evaluate their presence in an RDF data stream, (4) and (5) Experiments, which describe experiments using the framework in different contexts and with different purposes and (6) Conclusions.

2. Background

Ontology learning can be defined as the processes and techniques applied to design ontologies either automatically or semi-automatically [2]. According to [9], 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. 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. 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 [10], 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 [9].

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 [8] – i.e., whether they are confirmations or counterexamples of the axiom. The selective confirmation [11] principle can be put into effect as well: a fact selectively confirms a hypothesis when not only it favours the hypothesis but also fails to confirm its negation. Not all facts are equally relevant for axiom testing, and, as such, the number of confirmations needs to be considered not in the context of all analysed instances, but of those that do entail the relationships under scrutiny [8].

Property characteristics, from a perspective of axiom suggestion for ontology enrichment purposes, have been described in [12]. 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 confirmations 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.

The possibilistic approach has been applied in the works of Tettamanzi et al. [8,13–16] 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 possibility theory better reflects how to deal with incomplete knowledge. In [13] (and, by extension, [8]), 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. Furthermore, being designed for use in ontology validation, the possibilistic approach, as presented, is also implicitly working under the implication of the knowledge available being complete – which is not the case when information is delivered via streams – although it is an unsupervised approach that should be applicable even in cases where data may be incomplete.

3. Problem Definitions

Definition 1 (Ontology) – An ontology $\mathcal{O}$ is a model $\mathcal{O} = (\mathcal{T}Box, \mathcal{A}box)$ such that $\mathcal{T}Box$ is the set of terminological axioms and $\mathcal{A}Box$ is the set of assertional axioms. These are described through some vocabulary $\mathcal{T}Box$, which identifies concepts $\mathcal{C}$ − also known as classes − and roles $\mathcal{R}$ – also known as properties, and axioms. Both concepts and roles can partake in subsumption ($C_1 \sqsubseteq C_2$) and equivalence ($C_1 \equiv C_2$) relationships. Additionally, for the set of individuals $\mathcal{I}$, concepts and role assertions take the form of $C(a)$ or $P(b, c)$, with $C \in \mathcal{C}$, $P \in \mathcal{R}$ and $a, b, c \in \mathcal{I}$.

Definition 2 (Ontology Evolution) – Process through which a version of an ontology $\mathcal{O}_1$ – which may or may not contain assertions – is modified into a different version, $\mathcal{O}_2$. The task which modifies an existing ontology into new, evolved version of it, is a function $evolve(\mathcal{O}_1, \mathcal{I}_s,\ EA) \rightarrow \mathcal{O}_2$ such that:

- $\mathcal{I}_s$ is a stream of individuals described through RDF. Each individual is considered complete for the timeframe (T) it pertains to: all triples that constitute its description arrive simultaneously, i.e., no new information can be added to an individual during the T. As such, we can make a close world assumption for each T.
- EA is an ordered set of Evolutionary Actions $\{EA_1, \ldots, EA_n\} \in P$ where P is the set of all possible Evolutionary Actions. The execution of each action leads to the inclusion or exclusion of an axiom from the ontology.

Definition 3 (Property Constructor Axiom) - a property axiom $\mathcal{A}x(P)$ - or *characteristics* [12] - one can use to describe how the property must be employed. There are seven property axioms in OWL, namely: Functionality (F), Inverse Functionality (IF), Transitiveness (T), Reflexiveness (R), Irreflexiveness (IR), Symmetry (S) and Asymmetry (AS). The definition of Reflexiveness, however, is too strong to evaluate properly considering the constrains of this approach, and it is not in the scope of this paper.

Focusing exclusively on the application of property axioms in Object Properties, $\mathcal{A}x(P)$ denotes one of the possible OWL property axioms among:

$$\begin{aligned}\mathcal{A}x(P) \in \{\, & FunctionalObjectProperty, \\ & InverseFunctionalObjectProperty, \\ & TransitiveObjectProperty, \\ & SymmetricObjectProperty, \\ & AsymmetricObjectProperty, \\ & IrreflexiveObjectProperty \,\}\end{aligned}$$

or, in a more summarized way:

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

All property axioms can be expressed as first order logic implications in the form:

$$\begin{gathered}\mathcal{A}x(P) \colon \forall i, \forall x_1, \dots, \forall x_n \\ B(P, i, x_1, \dots, x_n) \,\rightarrow\, H(P, i, x_1, \dots, x_n)\end{gathered}$$

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 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 is the resulting implication that the ranges of said property must therefore be the same – $H(P, i, x_1, x_2) : x_1 = x_2$.

Definition 4 (Axiom Testing) – for testing a property axiom $\mathcal{A}x(P)$, we assume that $\mathcal{T}Box \cup \mathcal{A}box$ is consistent and that $\mathcal{T}Box \nvDash \mathcal{A}x(P)$ and $\mathcal{T}Box \nvDash \neg\mathcal{A}x(P)$.

Given $\mathcal{A}box$ and any $\mathcal{A}x(P)$, examples of $\mathcal{A}x(P)$ in $\mathcal{A}box$ are defined by the set of instances for which there is at least one substitution $i/a, x_1/b_1, \dots, x_n/b_n$ such that $\mathcal{T}Box, \mathcal{A}box \vDash \neg B(P, a, b_1, \dots, b_n) \vee H(P, a, b_1, \dots, b_n)$. Scenarios that do not rely on $B(P, a, b_1, \dots, b_n)$ being true are not counted for support.

Support for $\mathcal{A}x(P)$ is defined as the set of instances for which there is at least one substitution $i/a, x_1/b_1, \dots, x_n/b_n$ such that $\mathcal{T}Box, \mathcal{A}box \vDash B(P, a, b_1, \dots, b_n) \wedge H(P, a, b_1, \dots, b_n)$; i.e., only the individuals for which there is at least one valid substitution are considered, instead of all possible valid substitutions.

Definition 5 (Confirmation and Counterexample) – In axiom testing, an individual from the stream can either confirm or deny the hypothesis $\mathcal{A}x(P)$. With $Fi(i)$ as the set of facts associated with an individual i from the stream $\mathcal{I}_s$, a confirmation can be described as any i for which $\mathcal{T}Box, \mathcal{A}box, \mathcal{A}x(P), Fi(i)$ is consistent and counterexamples those for which it is not. If i does not use the property P, it is not considered as support and is not considered either a confirmation or counterexample.

Definition 6 (Sliding Window) – with the exception of Irreflexiveness, all property axioms pertain to the relation between one individual and others than itself. Since the individuals are provided by a stream $\mathcal{I}_s$, the analysis must be executed over a sliding window (w_s) constituted of a fixed number of individuals. In a first-in-first-out fashion, every time $\mathcal{I}_s$ delivers a new individual, it is added to w_s, until its maximum size is reached. Afterwards, for each new arrival, the oldest individual is removed from w_s.

Finally, some considerations about open vs closed world assumptions need to be made:

- Unique Name Assumption (UNA): in an open world assumption, the same entity can have different names. However, and since this would make the identification some axioms nearly impossible, operating under a Unique Name Assumption makes more sense. Any two individuals with different identifier URIs must be distinct unless there is evidence to the contrary (such as a *sameAs* assertion).
- Timeframe: while we do not wish to work under a strictly closed world assumption, we consider that the conclusions reached under a certain period are valid for that period exclusively and may be different than future or previous assertions. Time here can be defined in a number of ways: (1) through the timestamps of the individuals themselves, should they have them, (2) the number of individuals seen, or (3) actual observational period.

3.1 Property Constructor Definitions

The analysis of real-time data can unravel patterns regarding a role/property usage that may hint at its formal definition. Each property axiom, per virtue of its definition, corresponds to a different pattern in the data. These patterns can be searched for among the individuals of a dataset, to ascertain how many of them support, deny, or otherwise do not affect the probability of a certain axiom being true. It is therefore important to define each of the axioms and their corresponding patterns. As such, we can define the generic task $computeAxiom$, which has the objective of gathering statistics concerning the property axioms mentioned before:

$$computeAxiom(\mathcal{T}Box, \mathcal{A}box, \mathcal{I}_s, T_s, \mathcal{A}x(P)) \rightarrow E^+, E^- \in \mathbb{N}^+ \times \mathbb{N}^+$$

in which:

- $\mathcal{I}_s$ is the given dataset, which is constituted by a stream of $Fi(i)$ arriving sequentially;
- $\mathcal{A}x(P)$ is the property axiom of P under scrutiny;
- E^+ and E^- reflect, respectively, how many individuals of $\mathcal{I}_s$ were categorized as confirmations or counterexamples of $\mathcal{A}x(P)$ during the timeframe T_s. Support (E^0) is provided by the sum of E^+ and E^-.

The task is guided by the individuals in the domain of the properties, and each individual i is a sample which may or may not support an axiom. Contrary to Tettamanzi's approach, instead of searching for all possible confirmations of the implication, we look for the conjunction of $\mathcal{A}x(P)$ with $Fi(i)$ in $\mathcal{I}_s$ that is in the domain of P: i.e., the substitution occurs at the individual level instead of for each triple. Each substitution can be enforced, for example, through means of a SPARQL query executed over w_s, for which the subject and predicates are known, and objects must follow the pattern that makes them compatible with the axiom. Code snippet 1 shows one of such possible queries, in which `#iURI` and `#pURI` are known and correspond, respectively, to i and P, and the existence of a single `?obj` is a confirmation of $S(P)$.

Code snippet 1 - SPARQL query representing Symmetry

```
SELECT ?obj WHERE {
       <$iURI> <$pURI> ?obj.
       ?obj <$iURI> <$pURI>.
}
```

The details of how each property axiom is defined and the implication to search for in the data are described next, in Table 1.

Table 1 – Definitions, implications, confirmations and counterexamples for each property axiom

$\mathcal{A}x$ (P)	Definition	Formula	Confirmations (E^+)	Counterexamples (E^-)
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.	$\forall\, i, y_1, y_2 :$ $P(i, y_1) \wedge P(i, y_2) \rightarrow y_1 = y_2$	i for which P has either only have one value or the values were equivalent	i for which P is used more than once and the ranges of P are distinct
IF	P cannot have the same entity on its range for more than one i (e.g. a unique ID cannot be shared by more than one individual).	$\forall\, i_1, i_2, y$: $P(i_1, y) \wedge P(i_2, y) \rightarrow\ i_1 = i_2$	i for which the value of y has not been the range of P in any previously analyzed individual in $\mathcal{I}_s$	i for which the value of y was found in the range of P of other previously seen individuals
T	P holds between any two individuals of in a sequence and holds between all individuals of that sequence. A transitive property propagates as a chain, in which each individual in the chain is related to all others ones through P	$\forall\ i, y, \mathrm{z}$ $: P(i, y) \wedge\ P(y, z) \rightarrow P(i, z)$	i for which P has propagated between any set of three individuals i, y, z, for which a $P(i, z)$ exists or can be entailed.	i for which P does not propagate between any set of three individuals i, y, z, or, if such propagation exists, $P(i, z)$ cannot be entailed.
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
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
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

The definitions presented above may become clearer with the following example. Consider a stream $\mathcal{I}_s$ comprised of an unknown number of individuals. Of these, a sliding window of 8 is available for analysis, and its current individuals are described in Table 2. Each individual is complete upon arrival and features a set of properties and values, but may contain references to individuals that are not in the window at the time of analysis.

Table 2 - Example individuals in a stream and respective properties

ID	hasName	hasEvolutionGroup	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	Monster EG, Grass 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:

- Support: i_1, i_3 and i_4, as they use the property being analysed. E^0 = 3.
- Confirmations: i_1, as the domain of the property is only used once per value. E^+= 1.
- Counterexamples: i_3 and i_4, as they have the same property value for different individuals under a unique name assumption. E^-= 2.

Individuals i_2 and i_5 do not count for support and are not used in the analysis of $IF(hasEvolutionGroup)$. They may be used in the analysis of axioms for different properties. For the $F(hasType)$, the classification is as follows:

- Support: i_1, i_2, i_3, i_4, i_5 and i_8. E^0 = 6.
- Confirmations: i_3, i_4, i_5, i_8 as they only have one use of the property and therefore fit the definition of functionality. E^+= 4.
- Counterexamples: i_1 and i_5, as they have multiple different values for the property at the same time. Since the granularity of the search is at the individual level, each counts as a single counterexample. E^-= 2.

3.2 The Possibilistic Approach to Axiom Scoring

The work described in [8] 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 axiom $Ax(P)$, 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 [13]:

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

N and Π are calculated differently depending on the approach to the data being used, with distinct conjunctive and disjunctive formulas. However, these formulas can also be interpreted by how they value confirmations and counterexamples, i.e., their relative weights. The conjunctive normal form, on one hand, gives more weight to counterexamples – a single counterexample is sufficient to quash N, regardless of how many confirmations are found in the data. On the other hand, there is the disjunctive normal form, which gives more value to confirmations – with Π being at its highest whenever confirmations are present.

With that in mind, in the cases where the counterexamples have more weight in the decision-making process than confirmations, the conjunctive normal form of N and Π is applied, as follows:

$$N\big(Ax(P)\big) = \begin{cases} \sqrt{1 - \left(\frac{\mathrm{E}^-_{Ax(P)}}{\mathrm{E}^0_{Ax(P)}}\right)^2} & ,\text{if } \mathrm{E}^-_{Ax(P)} = 0 \\ 0 & ,\text{if } \mathrm{E}^-_{Ax(P)} > 0 \end{cases}$$

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 samples considered, i.e., the support, $E^+_{Ax(P)}$ the number of confirmations and $E^-_{Ax(P)}$ the number of counterexamples.

In the cases in which counterexamples are not sufficient to exclude a hypothesis, and confirmations have more relative weight, of N and Π are calculated according to:

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

and

$$\Pi(Ax(P)) = \begin{cases} 1 - \sqrt{1 - \left(\frac{\mathrm{E}^+_{Ax(P)}}{E^0}\right)^2} & ,if\ \mathrm{E}^+_{\mathrm{Ax(P)}} = 0 \\ 1 & ,if\ E^+_{Ax(P)} > 0 \end{cases}$$

The inherent incompleteness of data available in the sliding window (and potentially in the entire stream) influences the decision upon which form to apply. This incompleteness is directly related to the ability to *correctly* classify each individual as a confirmation or a counterexample. Functionality, for example, by virtue of its falsifying process, is bound to be erroneously confirmed by a lot of individuals, generating an abundance of false positives. Returning to the example provided in Table 2, all individuals are confirmations of the functionality of the property *hasName.* The only way to prove that a property is not functional is if it is shown having more than one distinct value. By making use of the property only once per individual, it is not possible to falsify functionality and therefore all individuals count for support. As such, individuals classified as counterexamples should be given more weight as they are synonymous with more completeness of the data; incomplete data in this case may generate a false positive, but never a false negative.

On the other hand, incompleteness of data in the case of, for example, transitivity, can generate false negatives, but not false positives. As such, it can be argued that it is more likely to find individuals incorrectly characterized as counterexamples, and therefore more weight should be given to individuals that do confirm the axiom. Transitivity, like symmetry, depends not only on the completion of data – how many individuals are in the sliding window at a given time – but also, in many cases, the relationships that would be positive evidence for them are not explicit.

Returning to the example in Table 2, not all individuals in the stream are equally available to be analysed – e.g. there is mention of an individual *Monster EG* that is not among the individuals in the sliding window. In the case of the *hatches* property, for example, one can see that it relates several individuals in a symmetrical fashion: i_5 relates to i_7 using this property and vice-versa, and the same happens between i_7 and i_8. These would, indeed, count as 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 3:

Table 3 - Axiom scoring for the *hatches* property using the conjunctive forms of N and Π

$Ax(hatches)$	E^+	E^-	N	Π	ARI
F	6	2	0	0,34	-0,66
IF	5	3	0	0,22	-0,78
T	5	3	0	0,22	-0,78
S	5	3	0	0,22	-0,78
IS	3	5	0	0,07	-0,93
IR	8	0	1	1	1

In not having information about individuals such as *Mineral EG* and *Ditto EG,* it is not possible to have all the data that needed to be taken into account for the remaining individuals to count as confirmations. As such, in the cases of T and S, incomplete data may lead to false negatives, and confirmations are much harder to find as they require more individuals to be on the same sliding window. When using the disjunctive form of ARI for T and S, the results better reflect the potential for those axioms to be present (see Table 4):

Table 4 - Axiom scoring for the *hatches* property using disjunctive forms of N and Π

$Ax(hatches)$	E^+	E^-	N	Π	ARI
T	5	3	0,93	1	0,93
S	5	3	0,93	1	0,93

Irreflexiveness necessitates only the analysis of a single individual – since they refer to the relationship the individual has with itself exclusively – and should not be influenced by the completeness of data or the size of the sliding window.

With this reasoning in mind, Table 5 summarizes which form of ARI is used per each property axiom:

Table 5 – ARI form to apply for each property axiom

Property Axiom	ARI form
Functionality	Conjunctive
Inverse Functionality	Conjunctive
Transitiveness	Disjunctive
Symmetry	Disjunctive
Asymmetry	Conjunctive
Irreflexiveness	Conjunctive

Finally, there are some considerations to keep in mind when using the definitions above to identify the potential axioms associated with a property, especially when working under an open world assumption:

- Just because there is no negative evidence for an axiom, it does not necessarily mean the axiom must be true. As per the Possibilistic approach to axiom score, it is possible nonetheless to ascertain if an axiom is at least compatible with the known data;
- Similarly, when analysing a dataset, it is possible that not all possible triples are asserted – particularly triples that could potentially be left for a reasoner to materialize, were the property axioms known (but may, for some reason, be missing). This can lead to an abundance of triples being miscategorized as negative evidence.
- Symmetry and asymmetry are directly opposite to one another, meaning the negative evidence for one will be positive evidence for the other, and therefore evidence for at least one of them will

almost always be found, which could result in over-characterization of properties;

– Some pairs of axioms are incompatible with each other: e.g., a property cannot be both functional and transitive, but it is possible that no negative evidence against both can be found simultaneously (in this case, it can happen because the algorithms that search for functionality and transitiveness are independent from each other and use different forms of ARI).

4. Experiment I

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 samples should be included in the sliding window and how does the support 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 axiom support exclusively?
2. Can the disjunctive form of necessity and possibility be applied to compensate for lack of confirmations?

The experiments described below were done using the set of ontologies in Table 6, which were selected for their property axioms and population sizes. Each populated ontology was split into samples of 3000 random individuals, which are then sequentially subjected to the analysing queries. Each query is then executed over a sliding window of individuals of the same class (to reduce the number of possible neutral individuals in the sample). Furthermore, and 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.

Table 6 – Properties and their respective axioms by class

	Type	Property	Property Axioms
CMT [17]	Paper	has Author	F
		has Decision	F
		readByMetaReviewer	F
		rejectedBy	F
	Person	addedBy	F
		addProgram CommitteeMember	IF
		assignedByReviewer	F
		assignExternalReviewer	IF
		rejectPaper	IF
		writePaper	IF
		writeReview	IF
WINE [18]	Region	adjacentRegion	S
	Region (as range)	locatedIn	T
Plant [19,20]	Thing	part_of/has_part	T

The approach followed in this work differs from that described in [12] by complementing the queries used for the identification of support with queries meant to specifically identify counterexamples. Two different types of queries were used for each property axiom: the confirmation queries (CQ), which ascertain if a property axiom could present, and the negation queries (NQ), which ascertain if a property axiom cannot be present – i.e., confirmation queries look for results that do not contradict the axiom, while the negation queries look for explicit contradictions. For the confirmation queries, we use a modified version of those described in [12], 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).

In classifying each individual as a confirmation or counterexample, each of the queries provides half of a 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 7.

Table 7 – 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)

Code snippet 2 and Code snippet 3 show one possible difference between confirmation and negation queries, using the functionality axiom as an example:

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 )
}
```

Each query is executed for each individual being tested and generates a set with a size of either 0 or 1 query solutions.

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 confirmation of the functionality axiom. If the property is indeed functional – i.e., if the true class in Table 7 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 7 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.

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

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

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

Experiments were conducted using three differently sized sliding windows, as to ascertain how “quickly” and effectively the queries can identify the possibility/probability of the presence of an axiom for a given individual: using 10, 50 and 100 individuals of the same type/class. 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 individuals always all 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), 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 to not be present. The values presented in Table 8 are applied on the following experiments, accessing how the relevance of each individual for support affects the evolution of the metrics and the potential results:

Table 8 – 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)	10 / 50 / 100
Property	Support
readByMetaReviewer	100%
rejectedBy	≈50%
addedBy	≈25%

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 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 1, with most individuals (around 94%) being incorrectly categorized as confirmations due to lack of information to the contrary.

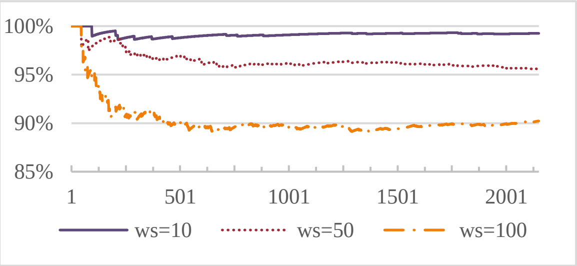


Figure 1 – Evolution of the confirmations of IF for the *readByMetaReviewer* property with w_s of 10, 50 and 100

However, by increasing w_s to 50, the categorization improves significantly: the number of 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 2 compares the evolution of precision, showing similar improvements although with a high propensity for misclassification (stabilizing around 10%).

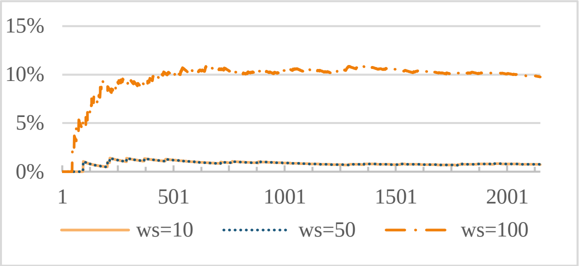


Figure 2 – Evolution of the precision of IF for the *readByMetaReviewer* property with w_s of 10, 50 and 100

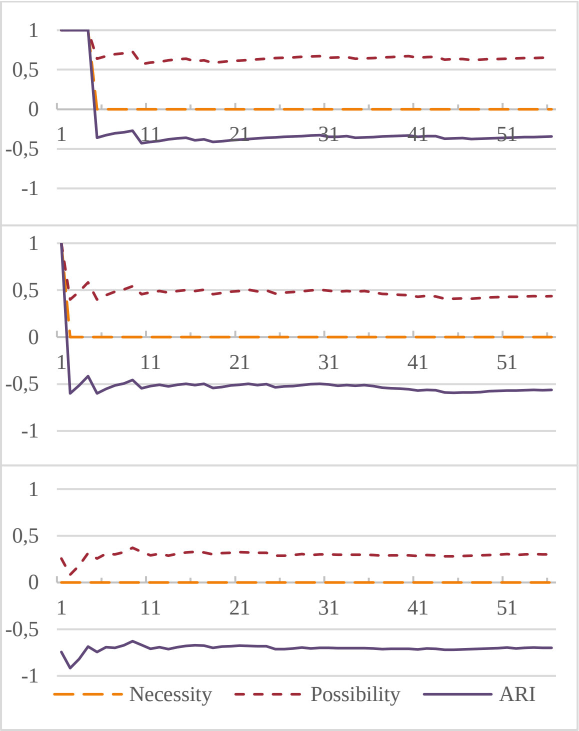


Figure 3 – Evolution of necessity, possibility and ARI of IF for the *readByMetaReviewer* property with w_s of 10 (first graph), 50 (second graph) and 100 (third graph)

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 3.

The metrics evolve differently when not all individuals in the stream are considered support.

Consider the results in Figure 4, which were obtained when searching for inverse functionality of the *rejectedBy* property, which is used by 1279 individuals of the *Paper* class (≈50%):

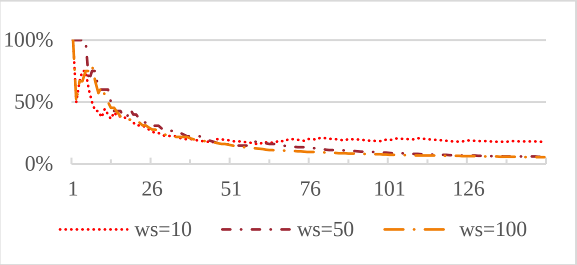


Figure 4 – Evolution of confirmations of IF for the *rejectedBy* property with w_s of 10, 50 and 100

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%.

Precision, recall and f-measure show similar evolution patterns as those of support, as seen in the first graph of Figure 5. 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 4. Recall stabilizes around 45% and f-measure at 60%.

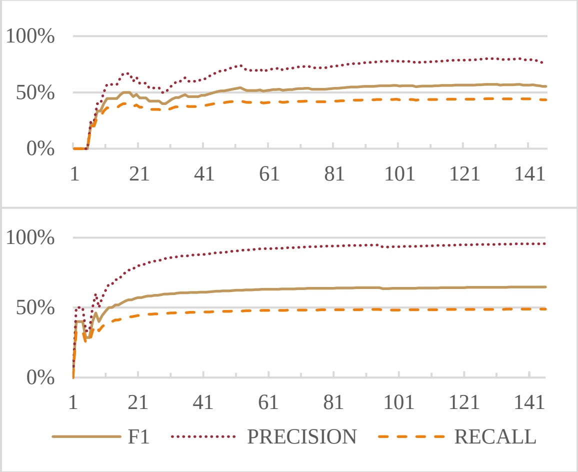


Figure 5 - Evolution of precision, recall and f-measure of IF for the *rejectedBy* property on with w_s of 10 (first graph) and 50 (second graph)

With a similar evolution to that of the support for the same window size, as illustrated in the second graph of Figure 5, 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 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.

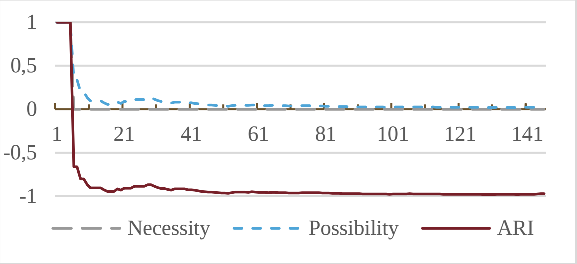


Figure 6 – Evolution of necessity, possibility and ARI of IF for the *rejectedBy* property with w_s of 10

Figure 6 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 confirmations, suffers a gradual loss as less and less 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 7.

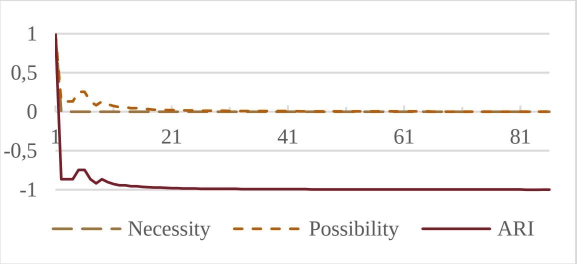


Figure 7 - Evolution of Necessity, Possibility and ARI of IF for the *rejectedBy* property with w_s of 50

There are no discernible changes in the evolution of necessity, possibility and ARI when the window size is increased, as the number of 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 confirmation or a counterexample, but only to a certain extent. Similar to support and information retrieval metrics, Figure 8 displays how ARI evolves slightly faster but is not significantly improved:

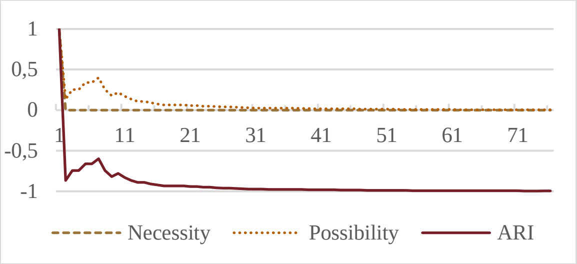


Figure 8 - Evolution of necessity, possibility and ARI of IF for the *rejectedBy* property with w_s of 100

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 shows Figure 9:

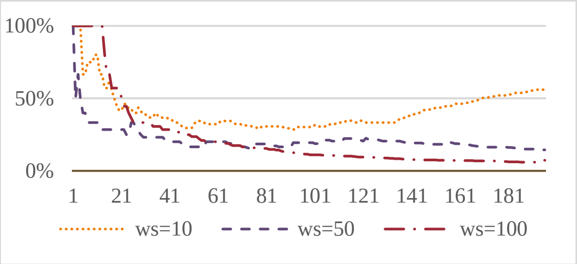


Figure 9 - Evolution of confirmations of IF for the property *addedBy* with w_s of 10, 50 and 100

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 confirmation or counterexample, but also does so correctly significantly faster. The same conclusion can be withdrawn from the analysis of the differences in precision, recall and f-measure for the three values of w_s, shown in Figure 10. Furthermore, it is important to note that for the smaller values of w_s, the evolution of the metrics is not positively monotonic (see the first graph of Figure 10). This suggests a lower w_s can be detrimental when parsing a large number of individuals.

Figure 10 - 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 11.

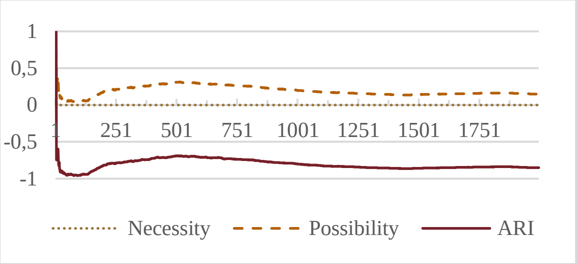


Figure 11 - Evolution of Necessity, Possibility and ARI of IF for the property *addedBy* with w_s of 10

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 12. In conclusion, in the cases where there is a very limited number of counterexamples, ARI may not ever 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.

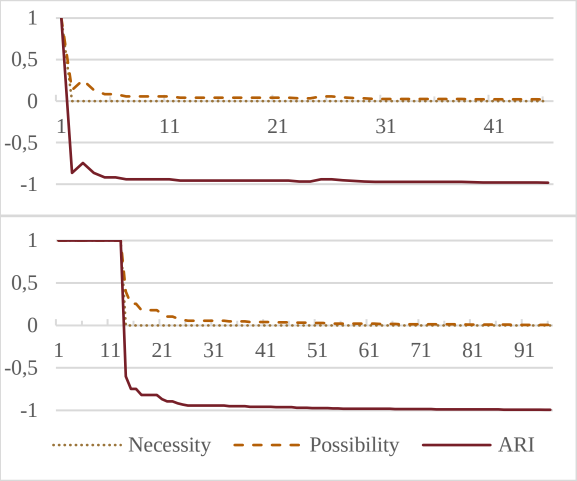


Figure 12 - Evolution of Necessity, Possibility and ARI of IF for the property *addedBy* with w_s of 50 (first graph) and 100 (second graph)

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 provide confirmation for multiple property axioms. Consider that, for example, 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 in for the functional property *hasAuthor*. Consider the difference between the graphs in Figure 13. 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 confirmations – and they continue even after a few counterexamples are encountered – the Necessity may drop to 0, and Possibility 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.

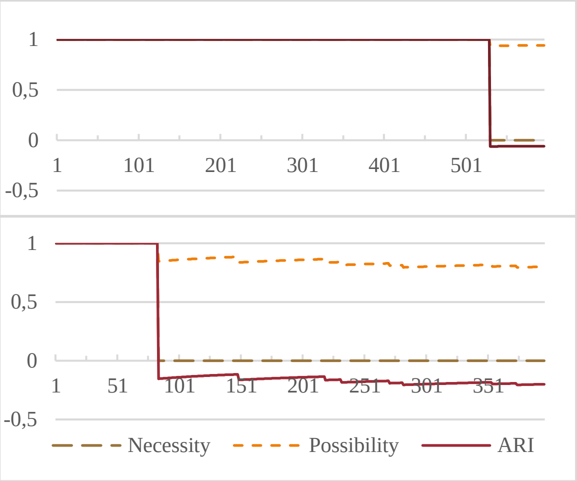


Figure 13 - Evolution of necessity, possibility and ARI of IF for *hasAuthor* with w_s of 10 (first graph) and 50 (second graph)

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 precision, necessity and ARI are always at 1.

4.2 *Section II – Using the Disjunctive Form of Possibility and Necessity*

Transitiveness 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 14 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 around 15%. Information retrieval metrics, as seen in Figure 15, also illustrate the clear lack of confirmations, which is the justification for applying the disjunctive forms of possibility and necessity.

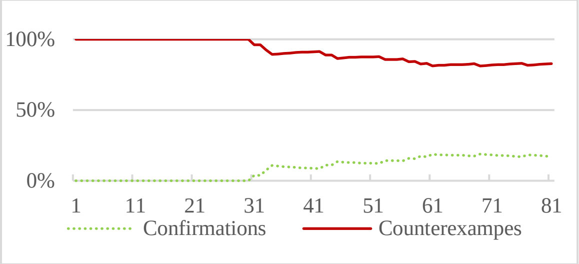


Figure 14 - Evolution of confirmations and counterexamples of T for the property *locatedIn* with w_s of 50

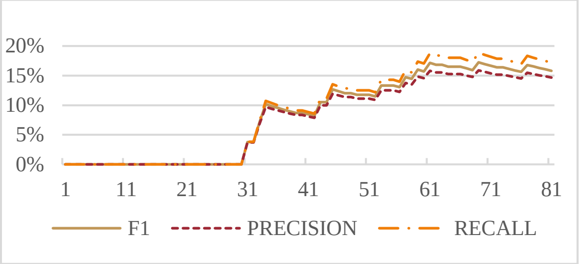


Figure 15 - Evolution of information retrieval metrics of T for the property *locatedIn* with w_s of 50

By giving more weight to the very hard to find confirmations than to the very easily incorrectly identified counterexamples, Figure 16 shows it is possible to obtain a positive, albeit conservative, ARI:

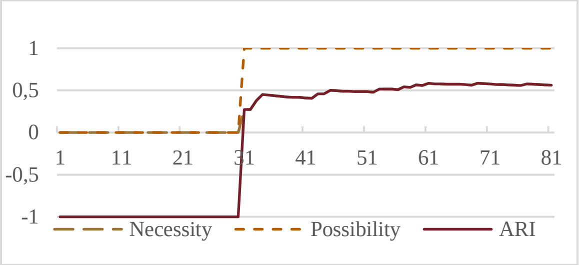


Figure 16 - Evolution of the disjunctive forms of possibility, necessity and ARI of T for the property *locatedIn* with w_s of 50

Should the conjunctive 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 *Insights*

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. As such, we consider that 50 is a good middle ground to allow for sufficient proper classification of individuals as confirmations or counterexamples, even if these results could be slightly improved with a bigger window.

The observation of the previous results show that the computation of ARI contains more information than that just the percentage of confirmations and could potentially be used in its place. Additionally, it shows better performance than traditional information retrieval metrics, achieving stronger 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 $\mathrm{E}^{+}_{\mathrm{Ax(P)}}$ of at least 95%. Alternatively, for axioms which there may be an overabundance of negatively-identified counterexamples and therefore make use of the disjunctive forms of possibility and necessity, 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, as mentioned earlier: 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, a confirmation of S is a counterexample of AS – however, a confirmation of either does not necessarily mean that the property axiom needs to be present; and therefore, any decision-making processes that come further down the line 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 necessity, possibility and ARI of 1 for F, IF, AS and IR, regardless of the forms of necessity and possibility applied.

5. Experiment II

The following experiments show the application of the axiom scoring process and how the decisions about which axiom to add to an evolving ontology were made in a dataset for which neither the axioms nor the properties were previously known.

The ontologies in use pertain to the Pokémon domain as it is described in Wikidata [21]. 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 new mechanics, regions and creatures. Much of this information is publicly available on Wikidata. However, while each concept is present and described with its own class, the same cannot be said about its properties: the properties in use in Wikidata, by virtue of how it is modelled, are very high level and lack the nuance necessary to describe specific domains beyond very simple connections between the data (e.g. part of, instance of). To overcome this problem, new subproperties were created (but not characterized) that pertain to specific relationships in the Pokémon domain. 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.

Considering this experiment pertains to an evolving ontology, with a potential new version (or at least for of some of its axioms) being created at the end of each timeframe, 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):

$$ARI_e = ARI * (1 - w_p) + Ax(P)_{Prev} * w_p$$

in which:

1. ARI is the Acception/Rejection Index calculated during the analysed timeframe;
2. $Ax(P)_{Prev}$ 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.

An 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.

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 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.

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, we do not find this 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 conjunctive-form using properties, axiom acceptance is informed by:

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

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

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

Finally, as F and T axioms are incompatible, and F was computed on stricter terms (using the conjunctive-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 experiments described next apply the values and thresholds in Table 9:

Table 9 - Values and thresholds employed in the experiments

Variable	Value
cf_t	95%
cj_t	0.5
dj_t	0.5
w_s (size)	50

The experiments below describe the application of both ARI and cf_t for axiom scoring. Each generation will be analysed as a different version of the ontology, and introductions and changes to its properties are documented.

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 catalogue) and 151 Pokémon. Using the data available on Wikidata, enough information was extracted to ascertain the properties described below. Table 10 shows the names, percentage of confirmations, proposed property axioms and ARIs for each.

bordersWith, a property that establishes a connection between two locations that share a border, seems a target candidate for S. Interestingly, the results also 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.

Table 10 – Gen I properties

Property Name	Axiom	%Cf	ARI
bordersWith	T	51%	0.87
	S	49%	0.86
hasEvolutionGroup	AS	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
hasWeight	F	96%	-0.28
hasFacet	F	100%	1
hasName	F	100%	1
hasPokedexNumber	F	100%	1
hasColor	F	99%	-0.16

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 IF, which the results support.

Of the datatype properties, three of them were classified as F even though counterexamples were found – since the percentage of 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 17 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 17 - Reasoner's explanation for inconsistency for property *hasWeight*


Explanation for: owl:Thing SubClassOf owl:Nothing

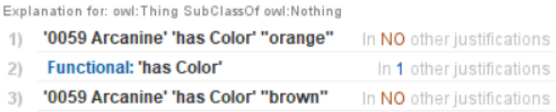
1) Functional: 'has Height' In NO other justifications

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

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

Figure 18 - Reasoner's explanation for inconsistency in property *hasHeight*

Figure 18 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 19 - Reasoner's explanation for inconsistency in property *hasColor*

Figure 19, 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

---

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

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 11. For the sake of brevity, only changes in axioms are shown. If an axiom is removed, it is preceded by a negation (¬) symbol.

Table 11 – Gen II properties

Property	Axiom	%Cf	ARI
bordersWith	T	65%	0.94
	S	63%	0.93
hasEvolutionGroup	F	100%	1
hasMoveType	F	100%	1
hasPokedexEntry	¬F	0%	-1
presentIn	¬F	37%	-0.93
alternateDexEntry	F	100%	1
	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
	S	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 can 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 IF. 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 both T and 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 20, there are some inconsistencies regarding the use of the *hasName* property, namely when describing the evolutionary groups:

```
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 20 - Inconsistencies with the use of the *hasName* property

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. Furthermore, when *hasName* is F while *hasPart* is T, the two become incompatible.

As seen in Figure 21, 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 21 - 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 12.

With of 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.

Table 12 – 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

Finally, *hatches* maintains its T and S. Evidence against IR was below the minimum threshold (confirmations amounting to 95% of the data) – but as it was already classified as T and S, 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 that replaces the previous one. 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 13:

Table 13 – Gen IV properties

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

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.

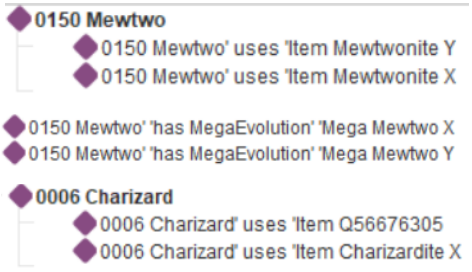
Table 14 – Gen VI properties

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.

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 22.



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

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).

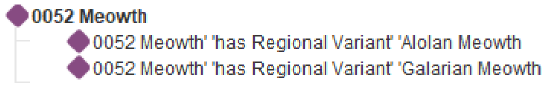
5.1.6 Generation VII

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 also revisits the region introduced in Gen I, adding new alternate forms to some previously known Pokémon.

Table 15 – Gen VII properties

Property	Axiom	%Cf	ARI
hasRegionalForm	F	97%	-0.24
	IF	98%	-0.19
	S	36%	0.77
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 23 shows one such case, in which a valid 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 23 – 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 16 shows the changes in the associated properties:

Table 16 – 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 24:

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 24 – 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 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, in the property was not present).

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 (cj_t). 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, cj_t and w_p should be inversely proportional. Since ARI cannot go over 1, for ARI_e to allow for evolution, cj_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 17 shows the thresholds employed for the following experiments, in which different weights for ARI_e will be compared.

Table 17 - Values and thresholds employed in the experiments

Variable	Value
cj_t	0.5
dj_t	0.5
w_p	0.8/ 0.7/0.4
w_s (size)	50

ARI_e depends on the results obtained in previous iterations, so its application can be measured from Gen II onwards. When previous decisions ([0-1] for not present or present, respectively) 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.

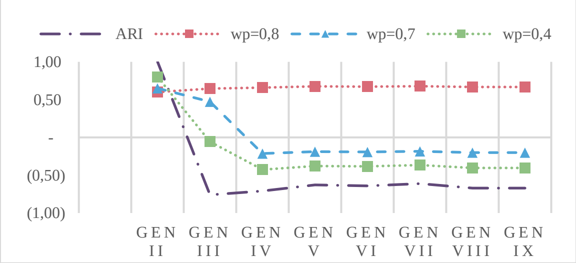


Figure 25 – Evolution of ARI_e for *hasAlternateDexEntry*'s F with different relative weights

Figure 25 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.

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 to not include the F axiom. Figure 26 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.

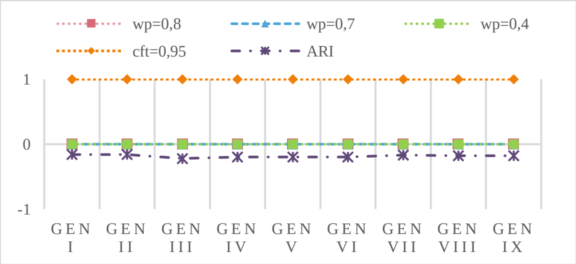


Figure 26 - Comparison of the decisions made using ARI_e and ARI+cf_t for F in *hasColor*

Differences in the decisions reached ARI, ARI+cf_t and ARI_e are shown by the *hatches* property, as seen in Figure 27 for T (results are the same for S). With the more conservative approach to change provided by ARI_e, the property is equally classified as T and S 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.

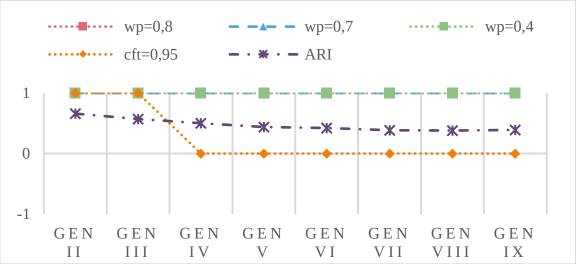


Figure 27 – Decisions regarding the T of the *hatches* property

In some generations, *isSignatureAbilityOf* and *hasSignatureAbility* were shown to have valid counterexamples that were not considered because of the chosen value for cj_t. Figure 28 shows the effect of the different w_p in the decision-making process:

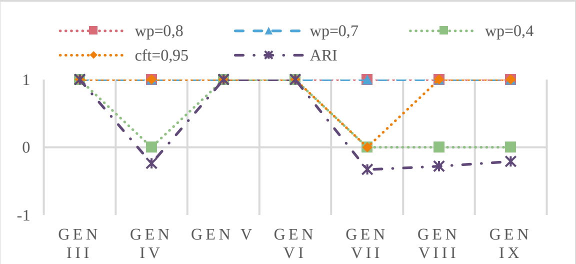


Figure 28 – Comparison of the decisions to include/exclude the F axiom for the *hasSignatureAbility* property

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 29 shows similar trends for IF of *isSignatureAbility* property:

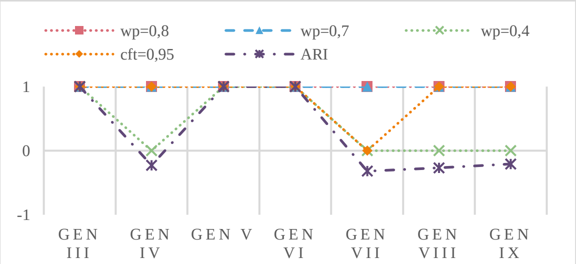


Figure 29 - Comparison of the decisions to include/exclude the IF axiom for the *isSignatureAbilityOf* property

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 30.

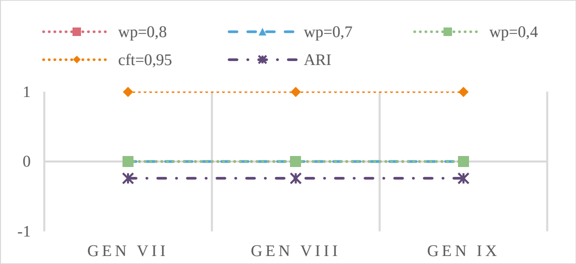


Figure 30 - Comparison of the decisions to include/exclude the F axiom for the *hasRegionalForm* property

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 31 shows:

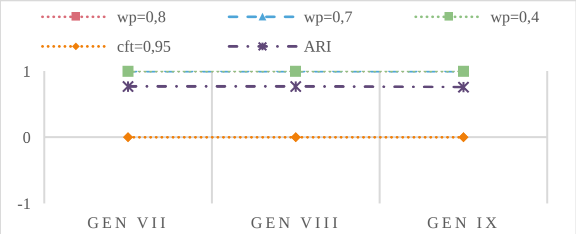


Figure 31 - Comparison of the decisions to include/exclude the T axiom for the *hasRegionalForm* property

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

5.3 *Insights*

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 – 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, it may have difficulty changing its opinion even in the face of overwhelming evidence – some tinkering with the cj_t and dj_t may be helpful. On some occasions, this may be beneficial: the experiments show that, when applied for T and S, the two property axioms that use the disjunctive-form of ARI because of their inherent missing information, their inclusion is favoured earlier, and harder to dismiss, with no inconsistencies resulting from it. However, for the conjunctive-form using properties, the effect is diametrically opposite, and axioms are maintained regardless of how many inconsistencies they may introduce.

6. Conclusions

This paper presented an adaptation of the possibilistic approach to axiom scoring to the context of RDF data streams for ontology evolution. As such, the approach was used 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. Results show the possibilistic approach can be applied to suggest potential axioms for ontology properties in an instance-guided ontology evolution scenario, but not without some caveats. The different approaches to possibility and necessity proposed in literature were recontextualized in terms of their bias towards confirmations or counterexamples – showing that some axioms, namely transitivity and symmetry, benefit from a more lenient approach, relying more on confirmations than on counterexamples – while the others benefit from stricter acceptance conditions to prevent the proliferation of inconsistencies in data. Nonetheless, the experiments suggest ARI is not sufficiently robust: the approach cannot account for the possibility of some of the counterexamples to be noise, which is a common situation when dealing with real-world data and affects decisions about the validity of the ontology.

Additionally, the experiments allowed for a better understanding of the impact of the size and variety of the data when it comes to axiom scoring, and how these need to be addressed for ontology evolution processes that take time periods into consideration.

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, which is very much a possibility when using real world data sources. Including axioms that are not 100% supported by ARI may result in identifying said errors, but also in erroneously discarding valid information. A more malleable approach, which acknowledges the knowledge 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, this allows for the continued integration of axioms that may lead to inconsistency scenarios. Ultimately, the approach is helpful for the construction and evolution of ontologies from RDF data from streams, but any results need to be understood in the context of their domain and analysed *a posteriori*.

Finally, one must also consider the over-characterization of the properties in an ontology. Results show there will almost always be one or more proposed axioms for each property, which may or may not make sense depending on the applicational context: upper-level ontologies, by definition and application, benefit from excluding superfluous axioms. However, there are also many scenarios in which the addition of said axioms will allow for richer inference processes to happen, unlocking 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 would not have been possible without the informed and pertinent insights from INRIA's mOeX

team and the efforts made by Wikidata's Project Pokémon[2].

References

[1] Polleres A, et al. How Does Knowledge Evolve in Open Knowledge Graphs? Transactions on Graph Data and Knowledge 2023;1. https://doi.org/10.4230/TGDK.1.1.11.

[2] Khadir AC, Aliane H, Guessoum A. Ontology learning: Grand tour and challenges. Comput Sci Rev 2021;39:100339. https://doi.org/10.1016/J.COSREV.2020.100339.

[3] Su X, et al. Stream reasoning for the internet of things: Challenges and gap analysis. ACM International Conference Proceeding Series 2016;13-15-June-2016. https://doi.org/10.1145/2912845.2912853.

[4] Esling P, Agon C. Time-series data mining. ACM Comput Surv 2012;45:12. https://doi.org/10.1145/2379776.2379788.

[5] Hamilton JD. Time series analysis. Princeton University Press.

[6] Burek P, Scherf N, Herre H. Ontology patterns for the representation of quality changes of cells in time. J Biomed Semantics 2019;10. https://doi.org/10.1186/s13326-019-0206-4.

[7] Canito A, et al. Using sensor data to detect time-constraints in ontology evolution. Integr Comput Aided Eng 2023;30:169–84. https://doi.org/10.3233/ICA-230703.

[8] Tettamanzi AGB, Faron-Zucker C, Gandon F. Possibilistic testing of OWL axioms against RDF data. IJAR 2017;91:114–30. https://doi.org/10.1016/j.ijar.2017.08.012.

[9] Al-Aswadi FN, Chan HY, Gan KH. Automatic ontology construction from text: a review from shallow to deep learning trend. Artif Intell Rev 2020;53:3901–28. https://doi.org/10.1007/S10462-019-09782-9/FIGURES/3.

[10] Buitelaar P, Cimiano P, Magnini B. Ontology Learning from Text: An Overview 2005.

[11] Scheffler I, Goodman N. Selective confirmation and the ravens: A reply to Foster. J Philos 1972;69:78. https://doi.org/10.2307/2024647.

[12] Bühmann L, Lehmann J. Universal OWL axiom enrichment for large knowledge bases. Lecture Notes in Computer Science 2012;7603 LNAI:57–71. https://doi.org/10.1007/978-3-642-33876-2_8/COVER.

[13] Tettamanzi AGB, Faron-zucker C, Gandon F. Testing owl axioms against RDF facts: A possibilistic approach. Lecture Notes in Computer Science 2014;8876:519–30. https://doi.org/10.1007/978-3-319-13704-9_39.

[14] Malchiodi D, Tettamanzi AGB. Predicting the possibilistic score of OWL axioms through modified support vector clustering. Proceedings of the ACM Symposium on Applied Computing 2018:1984–91. https://doi.org/10.1145/3167132.3167345.

[15] Nguyen TH, Tettamanzi AGB. An evolutionary approach to class disjointness axiom discovery. Proceedings - 2019 IEEE/WIC/ACM International Conf on Web Intelligence, WI 2019:68–75. https://doi.org/10.1145/3350546.3352502.

[16] Tettamanzi AGB, Zucker CF, Gandon F. Dynamically time-capped possibilistic testing of SubClassOf axioms against RDF data to enrich schemas. Proceedings of the 8th International Conf on Knowledge Capture, K-CAP 2015 2015. https://doi.org/10.1145/2815833.2815835.

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

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

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

[20] Bruskiewich R, et al. The Plant Ontology Consortium and Plant Ontologies. Comp Funct Genomics 2002;3:137–42. https://doi.org/10.1002/CFG.154.

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

---

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