On the Explanation of SameAs Statements using Argumentation

Abdallah Arioua, Madalina Croitoru, Laura Papaleo, Nathalie Pernelle, Swan Rocher

HAL Id: lirmm-01337012
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01337012v1

Submitted on 10 Nov 2022

On the Explanation of SameAs Statements using Argumentation

Abdallah Arioua[1], Madalina Croitoru[2], Laura Papaleo[4,5], Nathalie Pernelle[3], Swan Rocher[2]

[1] GraphIK, INRA, Montpellier, France
[2] GraphIK, Univ. of Montpellier, France
[3] LaHDAK, LRI, Univ. Paris Sud, France
[4] ICT Department, Metropolitan City of Genoa, Italy
[5] Tetherless World Constellation, Rensselaer Polytechnic Institute, USA

Abstract. Due to the impressive growing of the LOD graph in the last years, assuring the quality of its content is becoming a very important issue. Thus, it is crucial to design techniques for supporting experts in validating facts and links in complex data sources. Here, we focus on identity links (*sameAs*) and apply argumentation semantics to (*i*) detect inconsistencies in sameAs statements and to (*ii*) explain them to the experts using dialogues. We formalize the framework, explaining its purposes. Finally we provide a promising preliminary evaluation and discuss on some interesting future directions we foresee.

1 Introduction

Today, we are experiencing an unprecedented production of resources, published as *Linked Open Data* (LOD). This is leading to the creation of a *global data space* with billions of assertions [9]. RDF [24] provides formal ways to build these assertions. Most of the RDF links, connecting resources coming from different data sources, are *identity links*, also called *sameAs statements*. They are defined using the *owl:sameAs* property, expressing that two URIs actually refer to the same thing [1]. Unfortunately, many existing identity links do not reflect genuine real identity [16, 15] and therefore might lead to inconsistencies. Over the years, inconsistency-tolerant semantics (e.g. [8, 7, 26, 27]) have been proposed for query answering over potentially inconsistent existing data (and thus overcoming inconsistencies within the data).

In this work, we formalize *explanation dialogues* that use *argument-based explanation* based on inconsistency tolerant semantics. Our explanation dialogue supports a domain expert in discovering inconsistencies as in (eventually) performing corrections on erroneous data, or in revising the logical rules used for the invalidation or even in deciding the (potential) redesign of the initial linking strategy.

The *explanation dialogue* relies on a method for invalidating the sameAs that computes *repairs* so that, when a sameAs is *not entailed by the defined semantics*, an explanation of the reasons against this entailment is provided.

This is the first work that uses argumentation for sameAs links invalidation together with a formalization of a general explanation framework supporting the dialogue between user and reasoner. The salient point of this paper is to show how inconsistency-

tolerant semantics can represent a first step in the direction of the design of new inter-active paradigms for assessing the quality of sameAs statements.

The paper is organized as follows. Section 2 argues on related works while in Section 3 we provide background notions for argumentation theory. Section 4 is devoted to the presentation of our argumentation problem for sameAs invalidation. Section 5 formally introduces the novel *Explanation Dialogue* and Section 6 provides an example of the overall strategy implemented in a prototype. Finally, Section 7 draws some concluding remarks and possible future directions.

2 Related Work

To the extent of our knowledge the work presented here is the first attempt to combine argumentation theory, identity links evaluation and explanation dialogues, however, related works can be found in the context of sameAs evaluation and in general approaches which use argumentation in the Semantic Web.

For what concern the sameAs validation problem, it is very recent and few methods exist. In [17] an approach is presented where the structural properties of large graphs of sameAs links are analyzed, without analyzing the quality. In [22] a framework dedicated to the assessment of sameAs using network metrics is described, while in [23] the authors reported on the quality of sameAs links in the LOD using a manual method. In [15], the author illustrates how to assess the quality of sameAs, using a constraint-based method which, in the end, consider only one property (name of the entity), while in [29] an ontology-based logical invalidation method is presented which discovers invalid sameAs by the use of contextual graphs build around the resources, thus using more properties. Finally, the recent work presented in [14] evaluate a sameAs by using position and relevance of each resource involved with regards to the associated DBpedia categories, modeled through two probabilistic category distribution and selection functions. We need to recall that there exist a lot of linking methods (see [21] as survey) that, during their process of sameAs discovery, include a strategy for evaluating the reliability of the sameAs just computed.

Regarding argumentation in the Semantic Web, several works exist that mainly address ontologies alignment agreement based on argumentation theory (e.g. [25, 18, 19]). Basically, all of them use argumentation to provide a final agreement (or a final answer), and do not exploit the argumentation as a form of explanation of the answer to a query. Recently, in [10] the problem of *data fusion* in Linked Data is addressed, by adopting a bipolar argumentation theory (with fuzzy labeling) to reason over inconsistent information sets, and to provide a unique answer.
This last method has common points with our line of work, namely the use of argumentation theory to detect inconsistencies, but the scenarios in which the approach is exploited are different as its general aim is. This obviously leads to different addressed issues and proposed solutions.

3 Background Notions

There exist two major approaches for representing an ontology for the OBDA (Ontology-Based Data Access) problem: (i) *Description Logics* (DL) (such as $\mathscr{EL}$ [4] and DL$_{Lite}$ [12] families) and (ii) *Rule-based Languages* (such as Datalog$_{+/-}$ [11] language). Despite its undecidability when answering conjunctive queries, different decidable fragments of Datalog$_{+/-}$ have been studied in the literature [6]. They overcome their limitations allowing *n*-arity for predicates and cyclic structures. We consider *the positive existential* conjunctive fragment of first-order logic, denoted by FOL($\wedge, \exists$), which is composed of formulas built with the connectors ($\wedge, \rightarrow$) and the quantifiers ($\exists, \forall$).

We consider first-order vocabularies with constants but no other function symbol. A term *t* is a constant or a variable. Different constants represent different values (unique name assumption), an atomic formula (or atom) is of the form $p(t_1, ..., t_n)$ where *p* is an n-ary predicate, and $t_1, ..., t_n$ are terms. A *ground* atom is an atom with no variables. A variable in a formula is *free* if it is not in the scope of a quantifier. A formula is *closed* if it has not free variable. We denote by $\mathbf{X}$ (bold font) sequences of variables $X_1, .., X_k$ with $k \geq 1$. A *conjunct* $C[\mathbf{X}]$ is a finite conjunction of atoms, where $\mathbf{X}$ is the sequence of variables occurring in *C*. Given an atom or a set of atoms *A*, $vars(A)$, $consts(A)$ and $terms(A)$ denote its set of variables, constants and terms, respectively.

An ***existential rule*** is a first-order formula of the form $R = \forall \mathbf{X} \forall \mathbf{Y}(H[\mathbf{X}, \mathbf{Y}]) \rightarrow \exists \mathbf{Z} C[\mathbf{Z}, \mathbf{Y}]$, with $vars(H) = \mathbf{X} \cup \mathbf{Y}$, and $vars(C) = \mathbf{Z} \cup \mathbf{Y}$ where *H* and *C* are *conjuncts* (hypothesis & conclusion of *R*), respectively. $R = (H, C)$ is a contracted form for *R*. An existential rule with an empty hypothesis is called a *fact*. A *fact* is an existentially closed (with no free variable) conjunct. A *rule* $r = (H, C)$ is *applicable* to a set of facts *F* iff there exists $F' \subseteq F$ such that there is a homomorphism π from *H* to the conjunction of elements of F'. If a rule *r* is applicable to a set *F*, its application according to π produces a set $F \cup \{\pi(C)\}$. The new set $F \cup \{\pi(C)\}$, denoted also by $r(F)$, is called *immediate derivation* of *F* by *r*. Finally, we say that a set of facts $F \subseteq \mathscr{F}$ and a set of rules $\mathscr{R}$ *entail* a fact *f* (and we write $F, \mathscr{R} \models f$) iff the closure of *F* by all the rules entails *f* (i.e. $\mathtt{Cl}_{\mathscr{R}}(F) \models f$). A ***negative constraint*** is a first-order formula $n = \forall \mathbf{X} H[\mathbf{X}] \rightarrow \bot$ where $H[\mathbf{X}]$ is a conjunct called hypothesis of *n* and $\mathbf{X}$ sequence of variables appearing in the hypothesis. A ***knowledge base*** $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ is composed of, a finite set of facts $\mathscr{F}$, a finite set of existential rules $\mathscr{R}$ and a finite set of negative constraints $\mathscr{N}$. Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, a set $F \subseteq \mathscr{F}$ is said to be *inconsistent* iff there exists a constraint $n \in \mathscr{N}$ such that $F \models H_n$, where H_n is the hypothesis of the constraint *n*. A set of facts is consistent iff it is not inconsistent. A ***conjunctive query*** (CQ) has the form $Q(\mathbf{X}) = \exists \mathbf{Y} \Phi[\mathbf{X}, \mathbf{Y}]$ where $\Phi[\mathbf{X}, \mathbf{Y}]$ is a conjunct such that $\mathbf{X}$ and $\mathbf{Y}$ are variables in Φ. A *Boolean* CQ (BCQ) is a CQ with *yes* or *no* as answer.

Inconsistency Handling. If a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ is inconsistent, then everything is entailed from it. A common way to face inconsistency [7, 26] is to construct maximal (with respect to set inclusion) consistent subsets of $\mathscr{F}$, called *repairs*, denoted by $\mathscr{R}epair(\mathscr{K})$. In this paper, we consider a fragment of our language where the deduction method (the chase) halts, thus the closure $\mathtt{Cl}_{\mathscr{R}}(F)$ of any set of facts *F* is *finite*. Once the repairs are computed, different semantics can be used for query

answering over the knowledge base. Here we focus on *brave-semantics* [26] and *ICR-semantics* [7].

The brave-semantics accepts a query if it is entailed from at least one repair. This kind of semantics has been criticized because it allows conflicting answers. Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base and let Q be a query. Q is brave-entailed from $\mathscr{K}$, written $\mathscr{K} \models_{brave} Q$ if and only if: $\exists \mathscr{A} \in \mathscr{R}epair(\mathscr{K})$ such that $\text{Cl}_{\mathscr{R}}(\mathscr{A}) \models Q$. A prudent and more preservative semantics has been proposed in [7]. Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base and let Q be a query. Q is ICR-entailed from $\mathscr{K}$, written $\mathscr{K} \models_{ICR} Q$ if: $\bigcap_{\mathscr{A} \in \mathscr{R}epair(\mathscr{K})} \text{Cl}_{\mathscr{R}}(\mathscr{A}) \models Q$.

An alternative method for handling inconsistency is the use of **argumentation**. Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, the *corresponding argumentation framework* $\mathscr{A}\mathscr{F}_{\mathscr{K}}$ is a pair (Arg, Att) where Arg is the set of arguments that can be constructed from $\mathscr{F}$ and Att is an asymmetric binary relation called *attack* defined over $\text{Arg} \times \text{Arg}$ (as defined in [13]). Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, an argument $\mathbf{a}$ is a tuple $\mathbf{a} = (F_0, F_1, \ldots, F_n, C)$ where: $(F_0, \ldots, F_n)$ is an $\mathscr{R}$-derivation of F_0 in $\mathscr{K}$, such that (i) F_0 is $\mathscr{R}$-consistent and (ii) C is an atom, a conjunction of atoms, the existential closure of an atom or the existential closure of a conjunction of atoms such that $F_n \models C$. F_0 is the support of the argument $\mathbf{a}$ ($\text{Supp}(\mathbf{a})$) and C is its conclusion ($\text{Conc}(\mathbf{a})$).

An argument $\mathbf{a}$ *supports* a query Q if $\text{Conc}(\mathbf{a})$ entails Q and $\mathbf{a}$ is *against* Q if it attacks at least one argument that supports Q. An attack between two arguments $\mathbf{a}$ and $\mathbf{b}$ expresses the **conflict** between their conclusions and supports. Thus, $\mathbf{a}$ attacks $\mathbf{b}$ iff there exists $f \in \text{Supp}(\mathbf{a})$ (f is a fact) such that the set $\{\text{Conc}(\mathbf{b}), f\}$ is $\mathscr{R}$-inconsistent.

Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base and $\mathscr{A}\mathscr{F}_{\mathscr{K}}$ be its *corresponding argumentation framework*. Let $E \subseteq \text{Arg}$ be a set of arguments. We say that E is *conflict free* iff there exist no arguments $a, b \in E$ such that $(a, b) \in \text{Att}$. E *defends* an argument a iff for every argument $b \in \text{Arg}$, if we have $(b, a) \in \text{Att}$ then there exists $c \in E$ such that $(c, b) \in \text{Att}$. E is *admissible* iff it is conflict free and defends all its arguments. E is a *preferred extension* iff it is maximal (with respect to set inclusion) admissible set (please see [20] for other types of semantics). We denote by $\text{Ext}(\mathscr{A}\mathscr{F}_{\mathscr{K}})$ the set of all extensions of $\mathscr{A}\mathscr{F}_{\mathscr{K}}$. a is *sceptically accepted* if it is in all extensions, *credulously accepted* if it is in at least one extension and *not accepted* if not in any extension.

In [13] has been proved the equivalence between *skeptical acceptance under preferred semantics* and *ICR-entailment*. This allows us to use the argumentation approach in our explanation dialogue (Sec.5) as to ensure its correctness and completeness w.r.t. ICR query explanation and failure.

Given a knowledge base $\mathscr{K}$ and a query Q, the general problem is to explain if Q is entailed by $\mathscr{K}$ or not. Let $\mathscr{K}$ be an inconsistent knowledge base, Q a Boolean conjunctive query. $\Pi = \langle \mathscr{K}, Q \rangle$ is a query result explanation problem (QREP) iff (i) $\mathscr{K}$ is inconsistent, and (ii) $\mathscr{K} \models_{brave} Q$. [3]. Using ICR semantics we distinguish:

1. The *Query Failure Explanation Problem* (QFEP). In the ICR setting, a QREP Π is defined to be a QFEP iff $\mathscr{K} \not\models_{ICR} Q$.
2. The *Query Acceptance Explanation Problem* (QAEP): In the ICR setting, a QREP Π is a QAEP iff $\mathscr{K} \models_{ICR} Q$.

The first one refers to the case when the query fails (*no* answer) due to contradictions; the second refers to the case when the query is accepted, so a *yes* answer is obtained.

4 QFEP for SameAs Invalidation

Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathcal{AF}_{\mathcal{K}}$ its *corresponding argumentation framework*. We define now the main components of $\mathcal{K}$ for a QFEP in case of a sameAs invalidation.

Defining the Facts, the Rules and the Negative Constraints. $\mathcal{F}$ is a set of facts including (*i*) RDF triples, coming from RDF graphs representing the knowledge described in (possibly) different inter-connected datasets, and (*ii*) facts asserting similarity values between specific literals. These second type of facts are in the form of

$$is[\text{prop}]Diff[\text{SimFunction}](x, y, \sigma)$$

where (*i*) $[prop]$ is the name of a datatype (inverse-functional) functional property, (*ii*) $[SimFunction]$ is a similarity measure (e.g. Jaccard, Levenshtein, ...), (*iii*) x, y are literals and (*iv*) σ is a similarity value between x and y. These facts are considered when σ is less than a given threshold ε, defined for the similarity measure $[SimFunction]$ of a given property $[prop]$.

There are several kind of logical rules that we consider. There are rules defined by the W3C standards: for instance, we exploit the OWL2 RL rules which define the $owl : sameAs$ predicate as being reflexive, symmetric, and transitive, and the rules that axiomatize the standard replacement properties. We also use rules declared or discovered using mining techniques on RDF triples. For these kind of rules, here, we consider two types of properties: functional and inverse-functional properties [1].

When a property p is a **datatype functional property**, it can be expressed via the following logical rule: $p(r, v) \wedge p(r, v') \rightarrow isEquiv(v, v')$, where $isEquiv$ expresses equivalence of two literals. If the property p is an **object functional property**, the following logical rule can be used: $p(r, v) \wedge p(r, v') \rightarrow sameAs(v, v')$. Instead, when p is an **inverse-functional property**, the logical rule is $p(w_1, x) \wedge p(w_2, x) \rightarrow sameAs(w_1, w_2)$

We also add the set of rules which have all the following form:

$$is[\text{prop}]Diff[\text{SimFunction}](x, y, \sigma) \rightarrow isDiff(x, y)$$

A rule like this basically asserts that, when two literals x and y have a low similarity value σ for a specific property $[prop]$, they are declared as different (thus the fact $isDiff(x, y)$ is added to $\mathcal{F}$).

In our setting, the negative constraints are very simple. The only necessary negative constraints are in the following form: $isEquiv(x, y) \wedge isDiff(x, y) \rightarrow \perp$, where $isEquiv(x, y)$ are predicates coming from the rules defined for the datatype functional properties and $isDiff(x, y)$ comes from the similarity value between the literals. Note that all the other negative constraints, meaningful for discovering inconsistencies for a given *sameAs*, can be logically derived from the rules defined before. In case of a datatype functional property *title* the following leads to an inconsistency:

$$sameAs(s,o) \wedge title(s,w) \wedge title(o,w_1) \wedge isDiff(w,w_1) \rightarrow \perp$$

This can be derived from one rule and a negative constraint, namely:

1. $sameAs(s,o) \wedge title(s,w) \wedge title(o,w_1) \rightarrow isEquiv(w,w_1)$
2. $isEquiv(w,w_1) \wedge isDiff(w,w_1) \rightarrow \perp$

The problem QFEP. For completing the components and thus the instantiation of the QFEP, using ICR semantics, in the setting of sameAs invalidation, we need to define the query Q which is basically a *sameAs(x,y)*. The problem becomes:

Query Failure Explanation Problem (*QFEP$_{sameAs}$*). Given a knowledge base $\mathcal{K} = (\mathcal{F},\mathcal{R},\mathcal{N})$ with $\mathcal{F}$, $\mathcal{R}$, $\mathcal{N}$ defined above and the query Q as a *sameAs(x,y)* statement. The *QFEP$_{sameAs}$*, in the ICR setting (which is equivalent to $\mathcal{AF}_{\mathcal{K}}$ as *argumentation framework*) is a QREP where $\mathcal{K} \not\models_{ICR} Q$.

At this point, we formally instantiated a QFEP as a *sameAs invalidation problem*. Given a sameAs statement (as query), by the use of facts, rules and negative constraints as described above, we are able to discover if the sameAs is not entailed with respect to the given knowledge base (in ICR semantics). This proves that a sameAs invalidation method can be seen as a instantiation of QFEP in ICR. By itself, this represents an interesting result when searching for effective methods for evaluating the quality of sameAs statements. But, we also need interactions with the domain experts to explain the problems encountered and to support the corrective actions. In the following, we define our explanation framework (and dialogues) which provides these interactive functionalities.

5   The Explanation Framework

It is clear that if a *sameAs* has problems, it makes sense to show to the experts what kind of actions and negative conditions lead to this answer. Here we introduce our explanation framework that is custom-tailored for the problem of *Query Failure Explanation Problem* under ICR-semantics in inconsistent knowledge bases.

Example 1 (Motivating Example). Let us consider the case of a QFEP $\Pi = \langle \mathcal{K}, Q \rangle$ with a query as $Q = worksIn(Linda,Statistics)$. The dialogue we would like is similar to the following:

Actor	Dialogue expression
User	Why not *worksIn(Linda,Statistics)*?
Reasoner	Because Linda works in Accounting.
User	Clarify?
Reasoner	Because Linda uses office o_1 and o_1 is located in Accounting, so Linda works in Accounting.
User	How's that a problem?
Reasoner	The following negative constraint is violated $\forall x \forall y \forall z \, (worksIn(x,y) \wedge worksIn(x,z) \wedge y \neq z) \rightarrow \perp$.
User	Understood.

This simple example (not explicitly related to SameAs) is only to clarify that, in our ideal explanation framework, each iteration need to respect certain rules and some predefined locutions must be used (like *understood*, *clarify*, *why*, etc). In addition, all the information must be represented as *arguments* and/or *elaboration of arguments*. Finally,

our dialogue will use a *turn taking mechanism* where the `User` and the `Reasoner` switch turn at each stage.

In the following, we formalize the dialogue system and a legal dialogue for our explanation framework and, for doing this, we define the necessary syntax and semantics. The formalization is based on a very preliminary work [3], where the idea of dialogue was firstly introduced. The novelty here is the full formalization of the dialogue with specific references and custom definitions to the problem at hand.

5.1 Syntax

Definition 1 (Dialogue system). *Given a QFEP $\Pi = \langle \mathcal{K}, Q \rangle$. A dialogue system for Π is a tuple $\mathcal{D} = (\Pi, \mathcal{P}r, \mathcal{U}, \mathbb{R})$, where Π is the **topic**, $\mathcal{P}r$ is the set of **participants**, $\mathcal{U}$ is a finite set of the allowed **utterances**, $\mathbb{R}$ is an irreflexive binary relation defined over $\mathcal{U}$ called the **reply** relation.*

The definition above is intentionally general, the reader should note that, in the case of this work, the **topic** of the dialogue is a discussion that aims to get the `User` understand the refusal of a query Q (*sameAs(x,y)*) in the $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ with $\mathcal{F}, \mathcal{R}, \mathcal{N}$ defined in the previous Section. The **participants** $\mathcal{P}r = \{\texttt{Reasoner}, \texttt{User}\}$ are *(i)* the `User`, namely the domain expert who is analysing the quality of a set of sameAs, *(ii)* and the `Reasoner`, who represents an agent providing explanations in case of refusal.

The set of allowed **utterances** $\mathcal{U}$ and the **reply** relation $\mathbb{R}$ for our dialogue system $\mathcal{D}$ is given in table 1. Note that a, a', t, t' and Q in the table represent metavariables of arbitrary well-formed syntactical objects (e.g. queries, arguments, integers, etc.) of an arbitrary formal language.

Utterances	Description	Reply
EXPLAIN$(t,\mathbf{U},Q)$	User requests an explanation for a query Q	ATTEMPT$(t',\mathbf{R},a')$
ATTEMPT$(t,\mathbf{R},a)$	Reasoner provides an explanation a	CLARIFY$(t',\mathbf{U},a)$ or NEGATIVE$(t',\mathbf{U},a')$
CLARIFY$(t,\mathbf{U},a)$	User requests a clarification regarding the explanation a	CLARIFICATION$(t',\mathbf{R},a')$
CLARIFICATION$(t,\mathbf{R},a)$	Reasoner clarifies by advancing a	DEEPEN$(t',\mathbf{U},a)$ or NEGATIVE$(t',\mathbf{U},a)$
DEEPEN$(t,\mathbf{U},a)$	User requests a deepening regarding the explanation a	DEEPENING$(t',\mathbf{R},a')$
DEEPENING$(t,\mathbf{R},a)$	Reasoner deepens by advancing a	NEGATIVE$(t',\mathbf{U},a)$
POSITIVE$(t,\mathbf{U},Q)$	User acknowledges understanding of Q	no reply
NEGATIVE$(t,\mathbf{U},a)$	User disacknowledges understanding of Q and present the argument a that supports the query Q	ATTEMPT$(t',\mathbf{R},a')$

Table 1: The set of allowed utterances $\mathcal{U}$. In the table $\mathbf{U}$ is `User` and $\mathbf{R}$ is `Reasoner`.

A dialogue *D* has a potential infinite sequence of legal utterances. An utterance is considered a **legal reply** for another utterance iff it is a correct reply with respect to the reply relation $\mathbb{R}$ and it is the turn of the participant *x* to talk. We provide here a simple explanatory example.

Example 2 (Legal/Illegal Reply). Consider the dialogue:
$\langle$EXPLAIN$(1,\texttt{User},Q)$, ATTEMPT$(2,\texttt{Reasoner},a)$, CLARIFY$(3,\texttt{User},a)$, NEGATIVE$(4,\texttt{User},a')\rangle$
As one may notice, the reply NEGATIVE to CLARIFY is illegal because it is not in $\mathbb{R}$. A legal reply would be CLARIFICATION$(4,\texttt{Reasoner},a')$.

At this point, it is necessary to define the protocol which will decide if a dialogue is legal or not. We introduce the following definition for a Legal Dialogue.

Definition 2 (Legal Dialogues). *Given a dialogue D_n at stage n, $n \geq 0$. The dialogue D_n is legal iff:*

Empty dialogue rule: *if $n = 0$ then D_0 is legal.*

Commencement rule: *if $n = 1$ then $D_1 = \langle u_1 \rangle$ is legal iff $u_1 = \text{EXPLAIN}(1, \texttt{User}, Q)$.*

Dialogue rules: *if $n > 1$ then D_n is legal iff D_{n-1} is legal and u_n is a legal reply to u_{n-1} and there is no $u_i \in D_{n-1}$, $i < n$ and u_n **equals** u_i.*

Our definition indicates that an empty dialogue is legal. Furthermore, a legal dialogue always starts with an explanation request made by the `User`. Also, the protocol defines a legal dialogue as a sequence of utterances which legally replies to each other and no utterance is repeated twice.

5.2 Semantics

Now we shift to the semantic aspect of the dialogue where we deal with the content of the utterances. For instance, the utterance $\text{EXPLAIN}(2, \texttt{User}, Q)$ is legal (syntactically correct) but it will not be semantically legal if $\Pi = \langle \mathcal{K}, Q \rangle$ is not a query result explanation problem (or, in our more specific case a QFEP). The same applies to the utterance $\text{ATTEMPT}(2, \texttt{Reasoner}, a)$ if a is not an argument or a combination of arguments in our argumentation framework.

In Table 2 we put the conditions under which a given utterance or a reply is considered semantically legal in our setting. Here a *deepening* of an argument a explains the conflict between a and another argument b by showing the set of violated constraints. A *clarification*, instead, intends to unfold the knowledge (rules) used in the argument a to exhibit the line of reasoning that drives the conclusion.

Utterances	Conditions
$\text{EXPLAIN}(t, \texttt{User}, Q)$	$\Pi = \langle \mathcal{K}, Q \rangle$ is a QFEP
$\text{ATTEMPT}(t, \texttt{Reasoner}, a)$	a is an argument in $\mathscr{AF}_{\mathcal{K}}$
$\text{CLARIFY}(t, \texttt{User}, a)$ or $\text{DEEPEN}(t, \texttt{User}, a)$	a is an argument in $\mathscr{AF}_{\mathcal{K}}$
$\text{CLARIFICATION}(t, \texttt{Reasoner}, a)$	a is a clarification of an arbitrary argument in $\mathscr{AF}_{\mathcal{K}}$
$\text{DEEPENING}(t, \texttt{Reasoner}, a)$	a is a deepening of an arbitrary argument in $\mathscr{AF}_{\mathcal{K}}$
$\text{POSITIVE}(t, \texttt{User}, Q)$	$\Pi = \langle \mathcal{K}, Q \rangle$ is the topic of the dialogue
$\text{NEGATIVE}(t, \texttt{User}, a)$	a is an argument in $\mathscr{AF}_{\mathcal{K}}$

Table 2: The utterances and their semantical conditions. $\mathcal{K}$ is an inconsistent knowledge base defined as in Section 4 and $\mathscr{AF}_{\mathcal{K}}$ is the corresponding argumentation framework.

The semantical legality must also be considered within a context where replies are taken into account. Table 3 indicates the conditions under which a reply is semantically legal. For instance, a reply by the utterance $\text{ATTEMPT}(2, \texttt{Reasoner}, a)$ to the utterance $\text{EXPLAIN}(1, \texttt{User}, Q)$ is legal but it will not be semantically legal if a is not a proponent (opponent) argument of the query Q.

The dialogue is defined as a finite set of semantically legal moves. An explanation dialogues is typed, depending on its topic. Here, the topic is a QFEP, thus our explanation dialogue D_n is called a *Query Failure Explanation Dialogue* QFED: the `Reasoner` will show, by presenting opponent arguments, why a query Q has failed.

Utterances	Replies	Conditions
EXPLAIN(t,U,Q)	ATTEMPT(t',R,a')	a' either supports or is against Q
ATTEMPT(t,R,a)	CLARIFY(t',U,a')	$a' = a$
ATTEMPT(t,R,a)	NEGATIVE(t',U,a')	a' supports or is against Q
CLARIFY(t,U,a)	CLARIFICATION(t',R,a')	if a' is a clarification of a
DEEPEN(t,U,a)	DEEPENING(t',R,a')	if a' is a deepening of a
CLARIFICATION(t,R,a)	DEEPEN(t',U,a')	$a' = a$
CLARIFICATION(t,R,a) or DEEPENING(t,R,a)	NEGATIVE(t',U,a')	a' is empty or a' supports or is against Q
POSITIVE(t,U,Q)	no reply	no condition
NEGATIVE(t,U,a)	ATTEMPT(t',R,a')	$a' \neq a$ and a' either supports or is against Q

Table 3: The replies and their semantical conditions. Here U is for `User` and R is for `Reasoner`.

6   First Results and Discussion

To verify our strategy, we have implemented a prototype of the explanation dialogue that communicates with a Datalog$_{+/-}$ rule-based reasoner called *Graal* [5]. For the knowledge base, we considered facts from the CORA dataset [28] and sameAs computed using the SILK framework [2]. We provide an example of sameAs invalidation explaining what has been obtained while running dialogues and we discuss over these results. Due to space limitations, here we present a single example and we provide only a meaningful portion of the set of facts, rules and negative constraints (only those related to the sameAs used in the query or in the dialogue).

Let us consider a query Q as *sameAs*(r_1,r_2), where r_1, r_2 are URIs describing two resources in CORA. We show our explanation framework in the form of a QFED, where the `User` and the `Reasoner` interact in order to explain why Q is invalid. In Table 4 we report a subset of the knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ we used. This subset provides sufficient details to discuss over the results.

To be more clear, the query $Q = $ *sameAs*(r_1,r_2) involves two resources which describe two 'conferences' with title (*confName*) 'proceedings aaai-98' and 'in proceedings of aaai', respectively. The query Q is *not entailed*, according to the inconsistency-tolerant semantics $\mathscr{AF}_{\mathcal{K}}$: the two conferences are *not the same*. In Table 5 we show our explanation dialogue providing details on the reasons *why Q is not entailed*.

As mentioned in the formal specification of the dialogue in Section 5, utterances succession respects certain constraints: in step **1.** the `User` is the one who is allowed to make the opening *move* (EXPLAIN), not the `Reasoner`. At step **2.** the `Reasoner` responds providing an argument against the query (ATTEMPT) and the request for clarification (CLARIFY) at step **3.** made by the `User` is followed by a response made by the `Reasoner` (CLARIFICATION). Note that, after this clarification, the possible utterances can be: (*i*) a deepening request (DEEPEN), followed immediately by a deepening response (DEEPENING) or (*ii*) a NEGATIVE (understanding dis-acknowledgment) since, according to the semantical conditions we provided in Table 3, another deepening request is prohibited.

Another interesting property of our explanation dialogue is that it provides to the domain expert (`User`) the possibility to ask additional *follow-ups*. In the portion of dialogue described in Table 6, we report an extension of the previous dialogue (Table 5), where the `User` inputs additional arguments supporting her query Q and thus she asks

Facts (portion of $\mathscr{F}$)
$sameAs(r_3,r_2), sameAs(r_2,r_4), sameAs(a_1,a_2), sameAs(r_1,r_2)$
$confName(r_1,$'proceedings aaai-98')
$confName(r_3,$'in proceedings aaai-98')
$confName(r_2,$'in proceedings of aaai')
$confName(r_4,$'in proc. aaai')
$isconfNameDiffLevenshtein($'proceedings aaai-98','in proceedings of aaai',$0.73)$
$isconfNameDiffLevenshtein($'proceedings aaai-98','in proceedings aaai-98',$0.73)$
$isconfNameDiffLevenshtein($'in proceedings aaai-98','in proc. aaai',$0.73)$
$isconfNameDiffLevenshtein($'proceedings aaai-98','in proc. aaai',$0.41)$
$ispageFromDiffJaccard(30,15,0)$
$published(a_1,r_1), published(a_2,r_3)$
$pageFrom(a_2,15), pageFrom(a_1,30)$

Rules (portion of $\mathscr{R}$)
$sameAs(x,y) \wedge published(x,w_1) \wedge published(y,w_2) \rightarrow sameAs(w_1,w_2)$
$sameAs(x,y) \wedge pageFrom(x,w_1) \wedge pageFrom(y,w_2) \rightarrow isEquiv(w_1,w_2)$
$sameAs(x,y) \wedge confName(x,w_1) \wedge confName(x,w_1) \rightarrow isEquiv(w_1,w_2)$
$sameAs(x,y) \wedge sameAs(y,z) \rightarrow sameAs(x,z)$
$sameAs(y,x) \rightarrow sameAs(x,y)$
$isDiff(y,x) \rightarrow isDiff(x,y)$
$isconfNameDiffLevenshtein(x,y,\sigma) \rightarrow isDiff(x,y)$
$ispageFromDiffJaccard(x,y,\sigma) \rightarrow isDiff(x,y)$

Negative Constraints (portion of $\mathscr{N}$ and implicitly derivable)
$isEquiv(x,y) \wedge isDiff(x,y) \rightarrow \perp$
[derivable negative constraints]
$sameAs(x,y) \wedge pageFrom(x,w_1) \wedge pageFrom(x,w_1) \wedge isDiff(w_1,w_2) \rightarrow \perp$
$sameAs(x,y) \wedge confName(x,w_1) \wedge confName(x,w_1) \wedge isDiff(w_1,w_2) \rightarrow \perp$

Table 4: A portion of the facts $\mathscr{F}$, rules $\mathscr{R}$ and negative constraints $\mathscr{N}$ used to build our knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$.

for further explanations. We continue from step 7 of Table 5 and, instead of declaring 'understood' (POSITIVE), we disacknowledge the dialogue by providing a feedback in form of an argument.

Finally, to better illustrate the explanation dialogue, we present here the sequence of utterances, in terms of the formal model we formalized before. The dialogue D_i ($i = 7$) depicted in Table 5 is the following, where a is an argument and C_a, D_a are clarification and deepening of a, respectively.

$\langle$EXPLAIN$(1, \texttt{User}, Q),$ ATTEMPT$(2, \texttt{Reasoner}, a),$ CLARIFY$(3, \texttt{User}, a),$
CLARIFICATION$(4, \texttt{User}, C_a),$ DEEPEN$(5, \texttt{User}, a),$ DEEPENING$(6, \texttt{User}, D_a),$
POSITIVE$(7, \texttt{User}, Q)\rangle$

The second dialogue (Table 6) is composed by 9 steps. Its formal representation as sequence of utterances is:

$\langle$EXPLAIN$(1, \texttt{User}, Q),$ ATTEMPT$(2, \texttt{Reasoner}, a),$ CLARIFY$(3, \texttt{User}, a),$
CLARIFICATION$(4, \texttt{User}, C_a),$ DEEPEN$(5, \texttt{User}, a),$ DEEPENING$(6, \texttt{User}, D_a),$
NEGATIVE$(7, \texttt{User}, a'),$ ATTEMPT$(8, \texttt{Reasoner}, a''),$ POSITIVE$(9, \texttt{User}, Q)\rangle$

It is worthy to make a consideration on the semantics of the utterance NEGATIVE, which has two goals. First, it declares that the User has not understood the last explanation; second, it provides to the Reasoner a feedback. This feedback is in form of an

1. `User:` **Why** r_1 **and** r_2 **are not the same?** • *formalism*: $Q = sameAs(r_1, r_2)$? • *explanation*: This is an **explanation request** made by the `User` (EXPLAIN).
2. `Reasoner:` **Because** r_1 **is the same as** r_4. • *formalism*: $\langle \{sameAs(r_1, r_2), sameAs(r_2, r_4)\}, sameAs(r_1, r_4)\rangle$. • *explanation*: `Reasoner` provides an **attempt** with an argument **against** Q (ATTEMPT).
3. `User:` **clarify?** • *explanation*: The `User` asks for a **clarification**, meaning the *rule* relying the support and the conclusion of the argument before. (CLARIFY)
4. `Reasoner:` r_1 **is the same of** r_2, r_2 **is the same of** r_4 **and by transitivity** r_1 **is the same of** r_4 • *formalism*: $\langle \{sameAs(r_1, r_2), sameAs(r_2, r_4)\}, \{sameAs(x, y) \wedge sameAs(y, z) \rightarrow sameAs(x, z)\}, \{sameAs(r_1, r_4)\}\rangle$. • *explanation*: The `Reasoner` provides the **clarification**. In this case, the rule of transitivity for *sameAs* (CLARIFICATION)
5. `User:` **Why is this a problem?** • *explanation*: the `User` wants to know more about the logic behind the attack (DEEPEN)
6. `Reasoner:` $confName_{r_1}$ **is 'proceedings aaai-98'** (a) **and** $confName_{r_4}$ **is 'in proc. aaai'** (b). a, b **have Levenshtein similarity of** 0.41. **It is impossible that** r_1 **and** r_4 **are the same thing and they have dissimilar conference names** ($confName$ **is** *functional*). • *formalism*: - $sameAs(r_1, r_4), confName(r_4, b), confName(r_1, a)$ - $isconfNameDiffLevenshtein(a, b, 0.41) \rightarrow isDiff(a, b)$ - $sameAs(x, y) \wedge confName(x, w_1) \wedge confName(x, w_1) \wedge isDiff(w_1, w_2) \rightarrow \bot$. • *explanation*: The `Reasoner` provides the facts, the rules and the negative constraints that triggers the attack. (DEEPENING)
7. `User:` **Understood.** • *explanation*: the `User` ends the dialogue (POSITIVE)

Table 5: A query failure explanation dialogue for a sameAs query involving the resources r_1 and r_2. For each dialogue we outline the formalism and the utterances involved.

7. `User:` $sameAs(a_1, a_2)$, a_1 **is published in** r_1, a_2 **is published in** r_3, $sameAs(r_3, r_2)$, **thus** r_1 **is the same as** r_2. (*published* **is functional.**) • *formalism*: - $sameAs(a_1, a_2), published(a_1, r_1), published(a_2, r_3), sameAs(r_3, r_2)$ - $sameAs(a_1, a_2) \wedge published(a_1, r_1) \wedge published(a_2, r_3) \rightarrow sameAs(r_1, r_3)$ - $sameAs(r_1, r_3) \wedge sameAs(r_3, r_2) \rightarrow sameAs(r_1, r_2)$. • *explanation*: The `User` gives disacknowledges and presents an argument that supports her query Q (NEGATIVE).
8. `Reasoner:` **But** a_1 **has** *pageFrom* **value** 30, a_2 **has** *pagefrom* **value** 15 **and** 30 **and** 15 **are different** • *formalism*: - $sameAs(a_1, a_2), pageFrom(a_1, 30), pageFrom(a_2, 15)$ - $ispageFromDiffJaccard(30, 15, 0) \rightarrow isDiff(30, 15)$ - $sameAs(x, y) \wedge pageFrom(x, w_1) \wedge pageFrom(x, w_1) \wedge isDiff(w_1, w_2) \rightarrow \bot$. • *explanation*: `Reasoner` provides an **attempt** with an argument **against** the previous argument (ATTEMPT).
9. `User:` **Understood.** • *explanation*: the `User` ends the dialogue (POSITIVE)

Table 6: A new portion of the failure explanation dialogue for an invalid sameAs involving the resources r_1 and r_2. In this case, the user asks for further explanations by providing an argument against the reasoner conclusion.

argument a'. Thus, if the `User` has an expectation about a query and her expectation is endorsed by an argument then, she can present this argument in this utterance. Henceforth, NEGATIVE$(7, \text{User}, a')$ can be read as "*I do not understand why Q is not entailed **given that** the argument a' supports it*". When a' is empty, the user has no argument to propose.

6.1 Discussion

Our tests on the prototype have shown that, running dialogues on various sameAs statements (computed externally and considered potentially problematic[1]) was a support for different corrective actions. In some cases, errors in the data have been found (e.g. resource 100001135 has *confYear* property value 0, while its correct sameAs resources are conferences of the year 1995, or resource 100000021 has *pageFrom* to 24.1 which is again an error since it should be 24). Thanks to the dialogues with the reasoner, the expert has easily located these problems. In some other tests, the explanation dialogue supported the expert to understand that an update of some similarity functions used in specific properties was necessary (e.g. Levensthein instead of Jaccard for *confName*), or that the threshold ε to determine "dissimilar literals" had to be lowered for some properties (e.g. *title*). Finally, at the very first running, we used a set of sameAs links computed loosely (full of erroneous links). Thanks to the explanation dialogue it was clear that every sameAs query had strong inconcistencies over fundamental properties and values, thus this supported the idea to redo the linking process with a different strategy (in our case using composite keys in the linkage phase).

An important question may occur at this point, *"what happens if the* `Reasoner` *has multiple explanations (several potential arguments against/for the query)?"*.
In this case, we adapt a selection strategy: we choose each time which argument must be presented. In this work we aim at providing a general account for such process, thus we use the concept of a *selection function* $\mathbb{S}$ over a set of arguments. Note that $\mathbb{S}$ can be instantiated to express preferences with respect to some criteria that can possibly be defined by the expert `User`, such as "the property *confName* is very important (high weight $w_{confName}$)" or "the property *year* may contain errors, thus it has lower importance (low weight w_{year})". To order the sameAs presented to the expert, we used *Graal* to compute all the conflicts in the knowledge base. Then, we highlighted those sameAs statements that were more involved in conflicts (and sub-sequentially more present in attacks in the corresponding argumentation framework). These sameAs have been compared with the gold standard of the CORA dataset, and they have been used to define the order by which the dialogue should propose the sameAs links to the `User`. The sameAs links with most attacks, thus the most debatable ones, were showed first. The procedure we used to compute the conflicts is expensive from a computational point of view[2]. Such approach can be further improved in future work, by suitably adapting the conflict computation in order to obtain an incremental any-time algorithm with better computational properties.

7 Concluding Remarks

In this paper we presented an *explanation dialogue* based on argumentation theory where a domain expert can interact with the reasoner regarding a problematic sameAs statement.

---

[1]Experiment, at this moment, with one domain expert

[2]exponential in the size of the facts in the knowledge base

The paper demonstrates the significance of the explanation framework by the use of a real world example. All the dialogue moves are detailed, so that the reader can comprehend the types of interactions allowed. To the extent of our knowledge, the work presented in this paper, is the first attempt to use argumentation for sameAs links invalidation and for providing an explanation framework.

The results we obtained with the first prototype are very promising, motivating us in the continuation of the research activity. In these days, we are working on conducting tests using different (in size and quality) synthetic datasets (e.g. OAEI) and, in the immediate future, we are planning to analyze and evaluate sameAs coming directly from the LOD. In parallel, we are studying suitable improvements and strategies in order to ensure scalability of the approach when dealing with big datasets.

Different interesting long-term research directions can be exploited. For example, it could be interesting to study how to design innovative methods for modeling and combining *contextual weights* associated to each property used in the QFEP. Such weights could depend on different factors such as the *reliability* (automatically acquired or computed) of each property in the initial dataset. In addition, these weights could include suggestions (or restrictions) provided directly from the expert/user (something like '*I trust this data, please consider it true over all the other computations*'), and so on.

Another interesting future research direction could be also to study suitable user interfaces (by the use of innovative interactive systems) in the explanation of the inconsistencies and the properties involved, such that the type of interactions as the way in which the arguments are presented could be more 'user-friendly' and supported by graphical representations.

Acknowledgments

References

1. *OWL 2 Web Ontology Language: Primer*. www.w3.org/TR/owl2-primer.
2. *Silk - The Linked Data Integration Framework*. Available at http://silk-framework.com/.
3. A. Arioua, N. Tamani, M. Croitoru, and P. Buche. Query failure explanation in inconsistent knowledge bases: A dialogical approach. In *Research and Development in Intelligent Systems XXXI*. Springer, 2014.
4. F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, 2005.
5. J.-F. Baget, M. Leclere, M.-L. Mugnier, S. Rocher, and C. Sipieter. Graal: A toolkit for query answering with existential rules. In *To Appear in Proc of RuleML*. 2015.
6. J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proceedings of IJCAI'11*, 2011.
7. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.

8. M. Bienvenu and R. Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of IJCAI'13*. AAAI Press, 2013.

9. C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Intern Journal Semantic Web Information Systems*, 5(3), 2009.

10. E. Cabrio, J. Cojan, S. Villata, and F. Gandon. Argumentation-based inconsistencies detection for question-answering over dbpedia. In *Proc. of the NLP&DBpedia workshop*, 2013.

11. A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:57–83, 2012.

12. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

13. M. Croitoru and S. Vesic. What can argumentation do for inconsistent ontology query answering? In *Scalable Uncertainty Management*, LNCS. Springer, 2013.

14. J. Cuzzola, E. Bagheri, and J. Jovanovic. *Database and Expert Systems Applications: DEXA 2015-Part I*, chapter Filtering Inaccurate Entity Co-references on the Linked Open Data, pages 128–143. Springer, 2015.

15. G. de Melo. Not quite the same: Identity constraints for the Web of Linked Data. In *Proc. of the Conference on Artificial Intelligence*. AAAI Press, 2013.

16. L. Ding, J. Shinavier, T. Finin, and D. McGuinness. owl:sameAs and Linked Data: An Empirical Study. In *International Web Science Conference*, 2010.

17. L. Ding, J. Shinavier, Z. Shangguan, and D. McGuinness. Sameas networks and beyond: Analyzing deployment status and implications of owl: sameas in linked data. In *Proc. ISWC*, LNCS. Springer, 2010.

18. P. Doran, V. Tamma, I. Palmisano, and T. Payne. Efficient argumentation over ontology correspondences. In *Intern. Conf. AAMAS*, 2009.

19. C. dos Santos and J. Euzenat. Consistency-driven argumentation for alignment agreement. In *International Workshop on Ontology Matching (OM-2010)*, 2010.

20. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI*, 77(2):321–357, 1995.

21. A. Ferrara, A. Nikolov, and F. Scharffe. Data linking. *J. Web Semantics*, 23, 2013.

22. C. Gueret, P. Groth, C. Stadler, and J. Lehmann. Assessing linked data mappings using network measures. In *Proc. Intern. Conf. ESWC*, LNCS. Springer, 2012.

23. H. Halpin, P. Hayes, and H. Thompson. When owl: sameas isn't the same: A preliminary theory of identity and inference on the semantic web. In *Intern. Workshop LHD*, 2011.

24. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan& Claypool, 1st edition, 2011.

25. L. Laera, I. Blacoe, V. Tamma, T. Payne, J. Euzenat, and T. Bench-Capon. Argumentation over ontology correspondences in MAS. In *Intern. Conf. AAMAS*, 2007.

26. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. Savo. Inconsistency-tolerant semantics for description logics. In *Proc. Intern. Conf. on Web Reasoning Rule Systems*, 2010.

27. T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Complexity of inconsistency-tolerant query answering in datalog+/-. In *Proc. of the Intern. Conf. on Ontologies, DB and Applications of Semantics*, LNCS. Springer, 2013.

28. A. McCallum, editor. *Cora Research Paper Dataset*. Available at http://people.cs.umass.edu/ mccallum/data.html.

29. L. Papaleo, N. Pernelle, F. Saïs, and C. Dumont. Logical detection of invalid sameas statements in RDF data. In *EKAW*, pages 373–384, 2014.