Scalable Possibilistic Testing of SubClassOf Axioms Against RDF Data to Enrich Schemas

Andrea G. B. Tettamanzi
Univ. Nice Sophia Antipolis, I3S, UMR 7271
06900 Sophia Antipolis, France
Email: andrea.tettamanzi@unice.fr

Catherine Faron-Zucker
Univ. Nice Sophia Antipolis, I3S, UMR 7271
06900 Sophia Antipolis, France
Email: faron@unice.fr

Fabien Gandon
INRIA
Sophia Antipolis, France,
Email: fabien.gandon@inria.fr

***Abstract*—Axiom scoring is a critical task both for the automatic enrichment/learning and for the automatic validation of knowledge bases and ontologies. We develop an axiom scoring heuristics based on possibility theory, which aims at overcoming some limitations of scoring heuristics based on statistical inference and working with open-world semantics. Since computing the possibilistic score can be computationally quite heavy, we propose a method based on time capping to alleviate the computation of the heuristics without giving up the precision of the scores. We evaluate our proposal by applying it to the problem of testing `SubClassOf` axioms against the DBpedia RDF dataset.**

I. Introduction

It is common practice, in the semantic Web, to put a strong emphasis on the construction or reuse of ontologies based on a principled conceptual analysis of a domain of interest, as a prerequisite for the organization of the Linked Open Data (LOD), much like a database schema must be designed before a database can be populated. While this approach is quite successful when applied to specific domains, it does not scale well to more general settings; it is aprioristic and dogmatic; it does not lend itself to a collaborative effort; it does not encourage the "raw data, now!" movement; etc. That is why an alternative, bottom-up, *grass-roots* approach to ontology and knowledge base creation better suits many scenarios: instead of postulating an *a priori* conceptualization of reality (i.e., an ontology) and requiring that facts comply with it, one can start from RDF facts and learn OWL 2 axioms. The two approaches can even be complementary when considering the validation, extension or revision of an existing schemas with regard to a base of facts.

Recent contributions towards the automatic creation of OWL 2 ontologies from large repositories of RDF facts include FOIL-like algorithms for learning concept definitions [**?**], statistical schema induction via association rule mining [**?**], and light-weight schema enrichment methods based on the DL-Learner framework [**?**], [**?**]. All these methods apply and extend techniques developed within inductive logic programming (ILP) [**?**]. For a recent survey of the wider field of ontology learning, see [**?**].

On a related note, there exists a need for evaluating and validating ontologies, be they the result of an analysis effort or of a semi-automatic learning method. This need is witnessed by general methodological investigations [**?**], [**?**] and surveys [**?**] and tools like OOPS! [**?**] for detecting pitfalls in ontologies.

Ontology engineering methodologies, such as METHONTOLOGY [**?**], distinguish two validation activities, namely verification (through formal methods, syntax, logics, etc.) and validation through usage. Whilst this latter is usually thought of as user studies, an automatic process of validation based on RDF data would provide a cheap alternative, whereby the existing linked data may be regarded as usage traces that can be used to test and improve the ontologies, much like log mining can be used to provide test cases for development in the replay approaches. Alternatively, one may regard the ontology as a set of integrity constraints and check if the data satisfy them, using a tool like Pellet integrity constraint validator (ICV), which translates OWL ontologies into SPARQL queries to automatically validate RDF data [**?**]. A workshop on RDF validation has been organized in 2013.[1] The mission of the RDF Data Shapes Working Group is to produce a W3C Recommendation for describing structural constraints and validate RDF instance data against those.[2] A similar approach also underlies the idea of test-driven evaluation of linked data quality [**?**]. To this end, OWL ontologies are interpreted under the closed-world assumption and the weak unique name assumption.

Yet this validation process may be seen from a reverse point of view: instead of starting from the *a priori* assumption that a given ontology is correct and verify whether the facts contained in an RDF base satisfy it, one may treat ontologies like hypotheses and develop a methodology to verify whether the RDF facts corroborate or falsify them. Ontology learning and validation are thus strictly related. They could even be seen as an agile and test-driven approach to ontology development, where the linked data is used as a giant test case library not only to validate the schema but even to suggest new developments.

Ontology learning and validation rely critically on (candidate) axiom scoring. In this paper, we will tackle the problem of testing a single, isolated axiom, which is anyway the first step to solve the problem of validating an entire ontology. Furthermore, to validate our approach on a very concrete case, we applied it to OWL 2 `SubClassOf` axioms.

The most popular scoring heuristics proposed in the literature are based on statistical inference. We argue that such a probability-based framework is not always completely satisfactory. We propose an axiom scoring heuristics based on a formalization in possibility theory of the notions of logical

[1] http://www.w3.org/2012/12/rdf-val/report
[2] http://www.w3.org/2014/data-shapes/charter

content of a theory and of falsification, loosely inspired by Karl Popper's approach to epistemology, and working with an open-world semantics. Our proposal is coherent with a recently proposed possibilistic extension of description logics [**?**], [**?**].

Some preliminary results [**?**] indicated that applying a possibilistic approach to test candidate axioms for ontology learning produces very promising results and suggested that the same approach could also be beneficial for ontology and knowledge base validation. At the same time, the proposed heuristics is much heavier, from a computational point of view, than the probabilistic scores it aims to complement. Fortunately, there is evidence (see [**?**] and Section **??** below) that the time it takes to test an axiom tends to be inversely proportional to its score. This suggests that (1) time-capping the test might be an acceptable additional heuristics to decide whether to accept or reject a candidate axiom, for an axiom which takes too long to test will likely end up having a very negative score; and that (2) ordering candidate axioms will enable to optimize the number of tested and learned axioms in a given time period. In this paper, we follow this suggestion and investigate the effectiveness of time-capped possibilistic testing of OWL axioms against the facts contained in an RDF repository. Our research question is, therefore: "Can time capping alleviate the computation of the proposed possibilistic axiom scoring heuristics without giving up the precision of the scores?". This paper is organized as follows: Section **??** presents the principles of axiom testing. Section **??** critically reviews probability-based axiom scoring heuristics and Section **??** proposes an alternative heuristic based on possibility theory. A framework for axiom scoring based on such heuristic is then presented in Section **??** and evaluated on subsumption axioms in Section **??**. Section **??** draws some conclusions and directions for future work.

II. Principles of Axiom Testing

Testing an axiom against an RDF dataset can be done by checking whether the formulas entailed by it are confirmed by the facts contained in the RDF dataset.[3]

A. Direct Model-Theoretic Semantics for OWL 2

We refer to the model-theoretic semantics of OWL 2 as defined in [**?**].[4] An interpretation $\mathcal{I}$ for a datatype map D and a vocabulary V over D is defined by an interpretation domain $\Delta^{\mathcal{I}} = \Delta_I \cup \Delta_D$ (Δ_I is the *object domain* and Δ_D the *data domain*), and a valuation function $\cdot^{\mathcal{I}}$ with seven restrictions: $\cdot^{C}$ mapping class expressions to subsets of Δ_I, $\cdot^{OP}$ mapping object properties to subsets of $\Delta_I \times \Delta_I$, $\cdot^{DP}$ mapping data properties to subsets of $\Delta_I \times \Delta_D$, $\cdot^{I}$ mapping individuals to elements of Δ_I, $\cdot^{DT}$ mapping datatypes to subsets of Δ_D, $\cdot^{LT}$ mapping literals to elements of the set of data values $(DT)^{DT}$ of D and $\cdot^{FT}$ mapping facets to subsets of $(DT)^{DT}$. Let ϕ be a candidate axiom; we denote by u_ϕ the support of ϕ, i.e., the cardinality of the set of formulas entailed by ϕ which will be tested against the facts contained in the RDF dataset. We shall define this notion of support with respect to an RDF dataset more precisely.

[3]Note that calling linked data search engines like Sindice could virtually extend the dataset to the whole LOD cloud.

[4]http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/, Section 2.2 Interpretations

B. Content of an Axiom

Let ϕ be a candidate axiom; we denote by u_ϕ the support of ϕ, i.e., the cardinality of the set of formulas entailed by ϕ which will be tested against the facts contained in the RDF dataset. We shall define this notion of support with respect to an RDF dataset more precisely.

Let BS be a finite set of formulas, constructed from the set-theoretic formulas expressing the semantics of ϕ, which we will call *basic statements*, which can be tested against an RDF dataset.

Let $gp(?\mathtt{x})$ and $gp(?\mathtt{x}, ?\mathtt{y})$ be SPARQL graph patterns where variables `?x` and `?y` occur (other variables may occur as well) and let $[gp(?\mathtt{x})]$ and $[gp(?\mathtt{x}, ?\mathtt{y})]$ denote the result set of SPARQL queries `SELECT ?x WHERE` { $gp(?\mathtt{x})$ } and `SELECT ?x ?y WHERE` { $gp(?\mathtt{x}, ?\mathtt{y})$ }, respectively.

BS contains all formulas constructed from the set-theoretic formula expressing the semantics of ϕ by omitting all quantifiers, replacing all symbols x denoting an individual of $\Delta^{\mathcal{I}}$ by every resource r or literal l occurring in the RDF dataset (this may be construed as $r^{\mathcal{I}} = x$ or $l^{\mathcal{I}} = x$), and replacing all symbols C denoting subsets of $\Delta^{\mathcal{I}}$ by $[gp(?\mathtt{x})]$, where $gp(?\mathtt{x})$ is an appropriate SPARQL translation of C, and all symbols R denoting subsets of $\Delta_I \times \Delta_I$ or $\Delta_I \times \Delta_D$ by $[gp(?\mathtt{x}, ?\mathtt{y})]$, where $gp(?\mathtt{x}, ?\mathtt{y})$ is an appropriate SPARQL translation of R.

For example, let us consider the test of candidate axiom ϕ = `SubClassOf(dbo:LaunchPad dbo:Infrastructure)` (or `dbo:LaunchPad` $\sqsubseteq$ `dbo:Infrastructure` in Description Logics (DL) syntax) in the DBpedia dataset. The semantics of ϕ is $\texttt{dbo:LaunchPad}^{\mathcal{I}} \subseteq \texttt{dbo:Infrastructure}^{\mathcal{I}}$, which can be also written as

$$\forall x \in \Delta^{\mathcal{I}}, x \in \texttt{dbo:LaunchPad}^{\mathcal{I}} \Rightarrow x \in \texttt{dbo:Infrastructure}^{\mathcal{I}}.$$

We may thus construct the relevant set of basic statements as

$$\{ \quad r \in [gp_{\texttt{dbo:LaunchPad}}(?\mathtt{x})] \Rightarrow r \in [gp_{\texttt{dbo:Infrastructure}}(?\mathtt{x})] : \\ r \text{ is a resource occurring in DBPedia} \quad \}.$$

We define the *content* of an axiom ϕ that we wish to evaluate as the set of basic statements it entails,

$$\text{content}(\phi) = \{\psi \in \text{BS} : \phi \models \psi\}. \tag{1}$$

The cardinality of $\text{content}(\phi)$ is finite, because BS is finite, and every entailment of formula $\psi \in \text{content}(\phi)$ may be tested, because it is a basic statement. Now we can define the support of ϕ as the cardinality of $\text{content}(\phi)$:

$$u_\phi = \|\text{content}(\phi)\|. \tag{2}$$

We denote by u_ϕ^+ the number of formulas entailed by ϕ (confirmations); and by u_ϕ^- the number of such formulas which are not entailed by ϕ (counterexamples). A few interesting properties of these three cardinalities are:

$$u_\phi^+ + u_\phi^- \quad \leq \quad u_\phi; \tag{3}$$

$$u_\phi^+ = u_{\neg\phi}^-, \quad u_\phi^- \quad = \quad u_{\neg\phi}^+, \quad u_\phi = u_{\neg\phi}. \tag{4}$$

For example, in the DBpedia dataset, out of the 85 entailments which can be tested for the candidate axiom ϕ = `SubClassOf(dbo:LaunchPad dbo:Infrastructure)`,

83 entailments hold, i.e. there are 83 instances in the RDF dataset which confirm it: $u_\phi^+ = 83$; and 1 entailment does not hold, i.e. there is 1 counterexample in the dataset : $u_\phi^- = 1$.

In the following we will first report the probability-based candidate axiom scoring then we will present the possibilistic axiom scoring we propose. In both approaches, the computation of axiom scores is based on the above presented notions of support, confirmation and counterexamples.

III. Probability-Based Candidate Axiom Scoring

A statistics-based heuristics for the scoring of candidate axioms used in the framework of knowledge base enrichment [**?**] may be regarded essentially as scoring a candidate axiom by an estimate of the probability that it entails a syntactic consequence of it, based on the facts stored in the RDF repository. Notice that every formula which is a syntactic consequence of a candidate axiom is both a potential confirmation (if the entailment holds) and a potential disconfirmation or falsifier (if the entailment does not hold) for the candidate axiom. In this section we critically review the probability-based axiom scoring heuristics to motivate an alternative based on possibility theory and presented in the next section.

This probability-based approach relies on the assumption of a binomial distribution, which applies when an experiment (here, checking if a candidate axiom entails a syntactic consequence of it) is repeated a fixed number of times, each trial having two possible outcomes (conventionally labeled *success* and *failure*), the probability of success being the same for each trial, and the trials being statistically independent. We might call these experiment outcomes *confirmation*, if the observed fact agrees with the candidate axiom (success), and *counterexample* or *falsifier*, if the observed fact contradicts it (failure).

Most concept and schema induction approaches proposed in the literature (e.g., [**?**], [**?**], [**?**]) use precision or confidence, defined as $\hat{p}_\phi = u_\phi^+/u_\phi$, i.e., the proportion of confirmations (or correct classifications/predictions) out of all instances considered (or classified/predicted) as the score of ϕ.

However, Bühmann and Lehmann point out [**?**] that estimating the probability of confirmation of axiom ϕ just by $\hat{p}_\phi = u_\phi^+/u_\phi$ would be too crude and would not take the magnitude of u_ϕ into account. They suggest instead to carry out such parameter estimation by performing a statistical inference.

One of the most basic analyses in statistical inference is to form a confidence interval for a binomial parameter p_ϕ (probability of confirmation of axiom ϕ), given a binomial variate u_ϕ^+ for support u_ϕ and a sample proportion $\hat{p}_\phi = u_\phi^+/u_\phi$. Most introductory statistics textbooks use to this end the Wald confidence interval, based on the asymptotic normality of $\hat{p}_\phi$, and estimate the standard error. This $(1-\alpha)$ confidence interval for p_ϕ would be

$$\hat{p}_\phi \pm z_{\alpha/2}\sqrt{\hat{p}_\phi(1-\hat{p}_\phi)/u_\phi}, \tag{5}$$

where z_c denotes the $1-c$ quantile of the standard normal distribution.

Now, the central limit theorem applies poorly to this binomial distribution with $u_\phi < 30$ or where $\hat{p}_\phi$ is close to 0 or 1. The normal approximation fails totally when $\hat{p}_\phi = 0$ or $\hat{p}_\phi = 1$. That is why Bühmann and Lehmann [**?**] base their probabilistic score on Agresti and Coull's binomial proportion confidence interval [**?**], an adjustment of the Wald confidence interval which goes: "Add two successes and two failures and then use Formula **??**." It should be observed, however, that such adjustment is specific for constructing 95% confidence intervals.

A remark about such approaches is in order. They only look for confirmations of ϕ, and treat the absence of a confirmation as a failure in the calculation of the confidence interval. This is like making an implicit closed-world assumption. In reality, definitions of explicit failures can be given (see, e.g., the one we will propose in Section **??**), but then the probability of finding a confirmation and the probability of finding a counterexample do not necessarily add to one, because there is a non-zero probability of finding neither a confirmation nor a counterexample for every tested entailment, as stated in Equation **??**. For example, in the DBpedia dataset, among the 85 entailments which can be tested for the candidate axiom ϕ = `SubClassOf(dbo:LaunchPad dbo:Infrastructure)`, one of them, involving individual `:USA`, corresponds to neither a confirmation nor a counterexample of the axiom. The probability of finding neither a confirmation nor a counterexample for any entailment is thus 1.1765%. Bühmann and Lehmann's scoring method should thus be corrected in view of the open-world assumption, for example by using $\hat{p}^* = u_\phi^+/(u_\phi^+ + u_\phi^-)$ as the sample proportion instead of $\hat{p}$.

However, there is a more fundamental critique to the very idea of computing the likelihood of axioms based on probabilities. In essence, this idea relies on the assumption that it is possible to compute the probability that a formula ϕ is an axiom given some evidence e in the RDF repository, for example e = "ψ such that $\phi \models \psi$", or e = "ψ such that $\psi \models \neg\phi$", or e = "ψ such that $\phi \not\models \psi$", etc., which, by Bayes' formula, may be written as

$$\Pr(\phi \mid e) = \frac{\Pr(e \mid \phi)\Pr(\phi)}{\Pr(e \mid \phi)\Pr(\phi) + \Pr(e \mid \neg\phi)\Pr(\neg\phi)} \tag{6}$$

Therefore, in order to compute (or estimate) such probability, one should at least be able to estimate probabilities such as

- the probability that a fact confirming ϕ is added to the repository given that ϕ holds;
- the probability that a fact contradicting ϕ is added to the repository by mistake, i.e., given that ϕ holds;
- the probability that a fact confirming ϕ is added to the repository by mistake, i.e., given that ϕ does not hold;
- the probability that a fact contradicting ϕ is added to the repository given that ϕ does not hold.

Now, it is not hard to argue that the above probabilities may vary as a function of the concepts and properties involved. Let us take a subsumption axiom `SubClassOf`$(C\ D)$ as an example. A fact confirming it is $D(a)$, with $C(a)$ in the dataset, whereas a fact contradicting it is $E(a)$, with $C(a)$ in the dataset and `DisjointClasses`$(E\ D)$ in the ontology. Assuming that

`SubClassOf`$(C\ D)$ holds, we may suspect that $D(a)$ is more likely to be found in the repository if D is either very specific or very general (like `foaf:Person`)), and less likely if it is somewhere in the middle. This supposition is based on our expectations of what people are likely to say about a. For instance, an average person, if asked "what is this?" when pointing to a basset hound, is more likely to answer "a dog" or "an animal" than, say, "a carnivore" or "a mammal", which, on purely logical grounds, would be perfectly valid things to say about it [**?**]. There is thus an inherent difficulty with estimating the above probabilities, one which cannot be solved otherwise than by performing a large number of experiments, whose results, then, would be hard to generalize. By this argument, any axiom scoring method based on probability or statistics is doomed to be largely arbitrary.

Another key argument for rejecting a probabilistic approach in the specific context of axiom induction from an RDF dataset like DBpedia is that this dataset contains facts automatically extracted from Wikipedia, which is the result of a collaborative effort — both in terms of edition (Wikipedia) and extraction (DBpedia mappings and schemas) —, whose coverage is not planned and subject to cultural and historical biases.[5] Therefore, there is no reason to assume that the facts contained in an RDF triple store be *representative* of all possible facts that could be recorded, unless that RDF store is the result of a planned and well-designed effort aimed at building a knowledge base providing uniform coverage of a given domain. Indeed, to use the number of facts supporting a hypothesis to estimate its probability one would have to make the very strong assumption that the finite set of facts in the RDF store is a representative sample of the infinite set of all "real" facts, whatever this means. Adopting a probabilistic approach whereas its assumptions are not fulfilled might lead to fallacious results.

IV. A Possibilistic Candidate Axiom Scoring

We propose an axiom scoring heuristics which captures the basic intuition behind the process of axiom discovery based on possibility theory: assigning to a candidate axiom a degree of possibility equal to 1 just means that this axiom is possible, plausible, i.e. is not contradicted by facts in the knowledge base. This is much weaker than assigning a probability equal to 1, meaning that the candidate axiom certainly *is* an axiom.

A. Possibility Theory

Possibility theory [**?**] is a mathematical theory of epistemic uncertainty. Given a finite universe of discourse Ω, whose elements $\omega \in \Omega$ may be regarded as events, values of a variable, possible worlds, or states of affairs, a possibility distribution is a mapping $\pi : \Omega \to [0, 1]$, which assigns to each ω a degree of possibility ranging from 0 (impossible, excluded) to 1 (completely possible, normal). A possibility distribution π for which there exists a completely possible state of affairs $(\exists \omega \in \Omega : \pi(\omega) = 1)$ is said to be *normalized*.

There is a similarity between possibility distribution and probability density. However, it must be stressed that $\pi(\omega) = 1$ just means that ω is a plausible (normal) situation and therefore should not be excluded. A degree of possibility can then be viewed as an upper bound of a degree of probability. Possibility theory is suitable to represent incomplete knowledge while probability is adapted to represent random and observed phenomena. We invite the reader to see [**?**] for a discussion about the relationships between fuzzy sets, possibility, and probability degrees.

A possibility distribution π induces a *possibility measure* and its dual *necessity measure*, denoted by Π and N respectively. Both measures apply to a set $A \subseteq \Omega$ (or to a formula ϕ, by way of the set of its models, $A = \{\omega : \omega \models \phi\}$), and are defined as follows:

$$\Pi(A) = \max_{\omega \in A} \pi(\omega); \tag{7}$$

$$N(A) = 1 - \Pi(\bar{A}) = \min_{\omega \in \bar{A}} \{1 - \pi(\omega)\}. \tag{8}$$

A few properties of possibility and necessity measures induced by a normalized possibility distribution on a finite universe of discourse Ω are the following. For all subsets $A \subseteq \Omega$,

1) $\Pi(\emptyset) = N(\emptyset) = 0, \quad \Pi(\Omega) = N(\Omega) = 1;$
2) $\Pi(A) = 1 - N(\bar{A})$ (duality);
3) $N(A) > 0$ implies $\Pi(A) = 1$,
   $\Pi(A) < 1$ implies $N(A) = 0$.

In case of complete ignorance on A, $\Pi(A) = \Pi(\bar{A}) = 1$.

B. Possibility and Necessity of an Axiom

The basic principle for establishing the possibility of a formula ϕ should be that the absence of counterexamples to ϕ in the RDF repository means $\Pi(\phi) = 1$, i.e., that ϕ is completely possible.

A hypothesis should be regarded as all the more *necessary* as it is explicitly supported by facts and not contradicted by any fact; and all the more *possible* as it is not contradicted by facts. In other words, given hypothesis ϕ, $\Pi(\phi) = 1$ if no counterexamples are found; as the number of counterexamples increases, $\Pi(\phi) \to 0$ strictly monotonically; $N(\phi) = 0$ if no confirmations are found; as the number of confirmations increases and no counterexamples are found, $N(\phi) \to 1$ strictly monotonically. Notice that a confirmation of ϕ is a counterexample of $\neg\phi$ and that a counterexample of ϕ is a confirmation of $\neg\phi$.

Here are a few postulates, based on our previous discussion, the possibility and necessity functions should obey:

1) $\Pi(\phi) = 1$ if $u_\phi^- = 0$;
2) $N(\phi) = 0$ if $u_\phi^- > 0$ or $u_\phi^+ = 0$;
3) let $u_\phi = u_\psi$; then $\Pi(\phi) > \Pi(\psi)$ iff $u_\phi^- < u_\psi^-$;
4) let $u_\phi = u_\psi$; then $N(\phi) > N(\psi)$ iff $u_\phi^+ > u_\psi^+$ and $u_\phi^- = 0$;
5) let $u_\phi = u_\psi = u_\chi$ and let $u_\psi^- < u_\phi^- < u_\chi^-$: then
$$\frac{\Pi(\psi) - \Pi(\phi)}{u_\phi^- - u_\psi^-} > \frac{\Pi(\phi) - \Pi(\chi)}{u_\chi^- - u_\phi^-},$$
i.e., the first counterexamples found to an axiom should determine a sharper decrease of the degree to

[5] For example, at the level of pop music, the coverage of DBpedia is very much biased towards anglophone artists. Even in domains, such as geographical data, which one would expect to be much more uniform and extensive, it turns out that the coverage of Wikipedia is far from being uniform.

which we regard the axiom as possible than any further counterexamples, because these latter will only confirm our suspicions and, therefore, will provide less and less information;

6) let $u_\phi = u_\psi = u_\chi$ and $u_\psi^- = u_\phi^- = u_\chi^- = 0$, and let $u_\psi^+ < u_\phi^+ < u_\chi^+$: then

$$\frac{N(\phi) - N(\psi)}{u_\phi^+ - u_\psi^+} > \frac{N(\chi) - N(\phi)}{u_\chi^+ - u_\phi^+},$$

i.e., in the absence of counterexamples, the first confirmations found to an axiom should determine a sharper increase of the degree to which we regard the axiom as necessary than any further confirmations, because these latter will only add up to our acceptance and, therefore, will provide less and less information.

A definition of Π and N which satisfies the above postulates is, for $u_\phi > 0$,

$$\Pi(\phi) = 1 - \sqrt{1 - \left(\frac{u_\phi - u_\phi^-}{u_\phi}\right)^2}; \quad (9)$$

$$N(\phi) = \begin{cases} \sqrt{1 - \left(\frac{u_\phi - u_\phi^+}{u_\phi}\right)^2}, & \text{if } u_\phi^- = 0, \\ 0, & \text{if } u_\phi^- > 0. \end{cases} \quad (10)$$

Notice that this is by no means the only possible definition, but we choose it because it is the simplest one (it derives from a quadratic equation; a linear equation would not satisfy all the postulates).

It may be shown that the above definition satisfies the duality of possibility and necessity, in that $N(\phi) = 1 - \Pi(\neg\phi)$ and $\Pi(\phi) = 1 - N(\neg\phi)$. As a matter of fact, we will seldom be interested in computing the necessity and possibility degrees of the negation of OWL 2 axioms, for the simple reason that, in most cases, the latter are not OWL 2 axioms themselves. For instance, while $C \sqsubseteq D$ is an axiom, $\neg(C \sqsubseteq D) = C \not\sqsubseteq D$ is not.

C. Axiom Scoring

We combine the possibility and necessity of an axiom to define a single handy acceptance/rejection index (ARI) as follows:

$$\mathrm{ARI}(\phi) = N(\phi) - N(\neg\phi) = N(\phi) + \Pi(\phi) - 1 \in [-1, 1]. \quad (11)$$

A negative $\mathrm{ARI}(\phi)$ suggests rejection of ϕ ($\Pi(\phi) < 1$), whilst a positive $\mathrm{ARI}(\phi)$ suggests its acceptance ($N(\phi) > 0$), with a strength proportional to its absolute value. A value close to zero reflects ignorance about the status of ϕ.

V. A Framework for Candidate Axiom Testing

However, unlike interpretation domains, RDF stores are incomplete and possibly noisy. To learn axioms from an RDF dataset, the open-world hypothesis must be made: the absence of supporting evidence does not necessarily contradict an axiom, and an axiom might hold even in the face of a few counterexamples. For example, for 143 out of 541 `SubClassOf` axioms in the DBpedia ontology, no resource in the DBpedia dataset provides any evidence; for 28, at least one counterexample is found in DBpedia 3.9. Axiom `SubClassOf(dbo:Person dbo:Agent)` even has 76 counterexamples!

A general algorithm for testing all the possible OWL 2 axioms in a given RDF store is beyond the scope of this paper. Here, we will restrict our attention to `Class` and `ObjectComplementOf` class expressions and to `SubClassOf` axioms. Scoring these axioms with their ARI requires to compute the interpretation of `Class` and `ObjectComplementOf` class expressions.

A. Computational Definition of the Interpretation of `Class` and `ObjectComplementOf` Class Expressions

We define a mapping $Q(E, \texttt{?x})$ from OWL 2 class expressions to SPARQL graph patterns, where E is an OWL 2 class expression, and `?x` is a SPARQL variable, such that the query `SELECT DISTINCT ?x WHERE` $\{\ Q(E, \texttt{?x})\ \}$ returns all the individuals which are instances of E, which we will denote by $[Q(E, \texttt{?x})]$.

For a `Class` class expression A (i.e., an atomic concept in Description Logics (DL)), $Q(A, \texttt{?x}) = \texttt{?x a } A$, where A is a valid IRI. For an `ObjectComplementOf` class expression, things are slightly more complicated, since RDF does not support negation. The model-theoretic semantics of class expressions of the form `ObjectComplementOf(`C`)` ($\neg C$ in DL syntax), where C denotes a class, is $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$. The obvious definition

$$Q(\neg C, \texttt{?x}) = \{\ \texttt{?x ?p ?o .}\ \texttt{FILTER NOT EXISTS}\ Q(C, \texttt{?x})\ \}, \quad (12)$$

has the problem of treating negation as failure, like in databases, where the closed-world assumption is made. Since we want to preserve an open-world semantics, $Q(\neg C, \texttt{?x})$ should be defined differently, as the union of the concepts that are disjoint from C. One might try to express this as the set of individuals x that are instances of a concept C' such that no individual z such that $C(z)$ is an instance of C', yielding the query

$$Q(\neg C, \texttt{?x}) = \{\ \texttt{?x a ?dc .}\ \texttt{FILTER NOT EXISTS}\ \{\ \texttt{?z a ?dc .}\ \ Q(C, \texttt{?z})\ \}\ \}, \quad (13)$$

where `?z` is a variable that does not occur anywhere else in the query. This translation is conceptually more satisfactory than the one in Equation **??**, but it just pushes the problem one step further, because this way of testing whether two concepts are disjoint is based on negation as failure too. The only way to be certain that two classes are disjoint would be to find an axiom to this effect in the ontology:

$$Q(\neg C, \texttt{?x}) = \{\ \texttt{?x a ?dc .}\ \texttt{?dc owl:disjointWith}\ C\ \}, \quad (14)$$

otherwise, either we find an individual which is an instance of both classes, and thus we know the two classes are not disjoint, or we do not, in which case the two classes may or may not be disjoint. The fact is, very few `DisjointClasses` axioms are currently found in existing ontologies. For example, in the DBpedia ontology, the query:

Fig. 1. A schematic illustration of the heuristics used to capture negation under the open world assumption. D'' is a concept which is declared to be disjoint with C in the RDF repository.

`SELECT ?x ?y { ?x owl:disjointWith ?y }`, executed on November 22, 2013 returned 17 solutions only.

To compare these three alternative definitions of $Q(\neg C, \texttt{?x})$, we may refer to the diagram in Figure **??**. We wish to estimate the actual extension of $\neg C$. Clearly, $Q(C, \texttt{?x})$ (in dark grey) underestimates the real extension of C (in light gray). Therefore, we may say that Equation **??** overestimates the real extension of $\neg C$, in the sense that it will regard as instances of $\neg C$ all individuals a for which "a `a` C" is not found in the RDF repository.

Now, if b is such that "b `a` C" is not known, but "b `a` D'" is known for some class D' and some instances of D' are known to be also instances of C, then it might well be that b is an instance of C as well. If, however a is such that "a `a` C" is not known, but "a `a` D" is known for some class D but no instance of D is known that is also an instance of C, then we are more likely to believe that a is not an instance of C. Therefore Equation **??** regards as instances of $\neg C$ fewer individuals, those for which it is highly likely that they do not belong to C. It might still overestimate the extension of $\neg C$, but much less than Equation **??**. In fact, it might even underestimate it, as far as we know.

On the other hand, it is certain that Equation **??** will underestimate the extension of $\neg C$, to the point that it will equate it with the empty set if no triple of the form "D'' `owl:disjointWith` C" is declared in the RDF repository. Furthermore, it might well be that an individual is an instance of $\neg C$ even though it is not an instance of an atomic class disjoint with C!

To sum up, Equation **??** looks like a sensible compromise between Equation **??** (too optimistic) and Equation **??** (too pessimistic).

We will end this section by arguing that a suitable definition of confirmation to adopt in this framework is Scheffler and Goodman's *selective confirmation* [**?**], which characterizes a confirmation as a fact not simply satisfying an axiom, but, further, favoring the axiom rather than its contrary. For instance, the occurence of a black raven *selectively confirms* the axiom `Raven` $\sqsubseteq$ `Black` because it both confirms it and fails to confirm its negation, namely that there exist ravens that are not black. On the contrary, the observation of a green apple does not contradict `Raven` $\sqsubseteq$ `Black`, but it does not disconfirm `Raven` $\not\sqsubseteq$ `Black` either; therefore, it does not selectively confirm `Raven` $\sqsubseteq$ `Black`. We will incorporate this principle in our computational definition of a confirmation in Section **??** below.

B. Computational Definition of the Content of `SubClassOf` Axioms

The semantics of `SubClassOf` axioms of the form $C \sqsubseteq D$ in DL syntax is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, which may also be written $x \in C^{\mathcal{I}} \Rightarrow x \in D^{\mathcal{I}}$. The content of such axioms may thus be defined as

$$\text{content}(C \sqsubseteq D) = \{D(a) : C(a) \text{ in the RDF store}\}, \quad (15)$$

because, if $C(a)$ holds, then

$$C(a) \Rightarrow D(a) \equiv \neg C(a) \vee D(a) \equiv \bot \vee D(a) \equiv D(a).$$

The support $u_{C \sqsubseteq D}$ of such axioms can be computed with the following SPARQL query:

```
SELECT (count(DISTINCT ?x) AS ?u)
WHERE {Q(C, ?x)}.
```
(16)

C. Computational Definition of the ARI of `SubClassOf` Axioms

In order to compute $ARI(C \sqsubseteq D)$, we must provide a computational definition of $u^{+}_{C \sqsubseteq D}$ and $u^{-}_{C \sqsubseteq D}$. We start with the following statements:

- confirmations are individuals i such that $i \in [Q(C, \texttt{?x})]$ and $i \in [Q(D, \texttt{?x})]$;
- counterexamples are individuals i such that $i \in [Q(C, \texttt{?x})]$ and $i \in [Q(\neg D, \texttt{?x})]$.

This may be translated into the following two SPARQL queries to compute $u^{+}_{C \sqsubseteq D}$ and $u^{-}_{C \sqsubseteq D}$ respectively:

```
SELECT (count(DISTINCT ?x) AS ?nConfirm)
WHERE { Q(C, ?x) Q(D, ?x) }
```
(17)

and

```
SELECT (count(DISTINCT ?x) AS ?nCounter)
WHERE { Q(C, ?x) Q(¬D, ?x) }.
```
(18)

Notice that an i such that $i \in [Q(C, \texttt{?x})]$ and $i \notin [Q(D, \texttt{?x})]$ does not contradict $C \sqsubseteq D$, because it might well be the case that the assertion "i `a` D" is just missing. Likewise, an $i \in [Q(\neg D, \texttt{?x})]$ such that $i \in [Q(\neg C, \texttt{?x})]$ will not be treated as a confirmation, based on our choice to regard as evidence in favor of a hypothesis only selective confirmations.

D. Heuristics based on Time Capping

The results of our first experimentation described in the following Section **??** show that the time it takes to test an axiom tends to be upper bounded by the inverse of $1+\text{ARI}(\phi)$: an axiom which takes too long to test will likely end up having a very negative score. We defined two heuristics based on this idea.

- We time-cap the SPARQL queries to compute the ARI of a candidate axiom and decide whether to accept or reject it, since above a computation time threshold, the axiom being tested is likely to get a negative ARI and be rejected.
- We construct candidate axioms of the form $C \sqsubseteq D$, by considering the subclasses C in increasing order of the number of classes D sharing at least one instance with C. This enables us to maximize the number of tested and accepted axioms in a given time period, since it appears that the time it takes to test $C \sqsubseteq D$ increases with that number and the lower the time, the higher the ARI.

VI. Evaluation on SubClassOf Axiom Testing

A. Experimental Protocol

We evaluated the proposed scoring heuristics by performing tests of subsumption axioms using DBpedia 3.9 in English as the reference RDF fact repository. In particular, on April 27, 2014, we downloaded the DBpedia dumps of English version 3.9, generated in late March/early April 2013, along with the DBpedia ontology, version 3.9. This local dump of DBpedia, consisting of 812,546,748 RDF triples, has been bulk-loaded into Jena TDB and a prototype for performing axiom tests using the proposed method has been coded in Java, using Jena ARQ and TDB to access the RDF repository.

We systematically generated and tested subsumption axioms involving atomic classes only according to the following protocol: for each of the 442 classes C referred to in the RDF repository, we construct all axioms of the form $C \sqsubseteq D$ such that C and D share at least one instance. Classes D are obtained with the following query:

```
SELECT DISTINCT ?D
WHERE {Q(C, ?x) . ?x a ?D}.
```

Experiments have been performed on two machines:

- a Fujitsu CELSIUS workstation equipped with twelve six-core Intel Xeon CPU E5-2630 v2 processors at 2.60GHz clock speed, with 15,360 KB cache each, 128 GB RAM, 4 TB of disk space with a 128 GB SSD cache, under the Ubuntu 12.04.4 LTS 64-bit operating system;
- a HP portable PC equipped with four two-cores Intel® Core™ i7-4600U CPUs at 2.10GHz clock speed, with a 4,096 KB cache, 16 GB RAM, 128 GB of disk space, under the Fedora 64-bit Linux operating system.

The former was used to test 644 axioms without time capping. The latter, much less powerful but representative of a common high-end laptop computer, was then used to obtain the results with time capping.

B. Results without Time Capping

We managed to test 644 axioms without time capping. Figure **??** compares the results obtained in this preliminary experiment to the probabilistic score proposed in [**?**]. As already discussed in [**?**], the proposed acceptance-rejection index is more accurate than the probabilistic score. However, its increased accuracy comes at a higher computational cost. The two heuristics, time capping and candidate axioms ordering, make up for it.

The results of this first experiment show that the time it takes to test an axiom tends to be inversely proportional to its score (see Figure **??**): an axiom which takes too long to test will likely end up having a very negative score. If we restrict our attention to the 197 axioms with an ARI above the acceptance threshold empirically set at $1/3$ in [**?**], we discover that the average elapsed time for testing them is 20.5 s, the median time is 156 ms, and the longest elapsed time is 1584.3 s, or 26 min 24 s. Out of the 197 accepted axioms, 135 (68.5%) were tested in less than 1 s, 165 (83.75%) in less than 10 s, 187 (95%) in less than 1 minute, and 195 (99%) in less than 10 minutes.

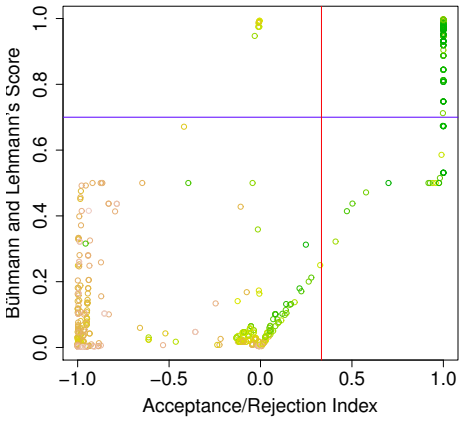

Fig. 2. A comparison of the acceptance/rejection index and the probability-based score used in [**?**] on axioms tested without time capping. The vertical line shows the acceptance threshold $\mathrm{ARI}(\phi) > 1/3$; the horizontal line the acceptance threshold of 0.7 for the probabilistic score.

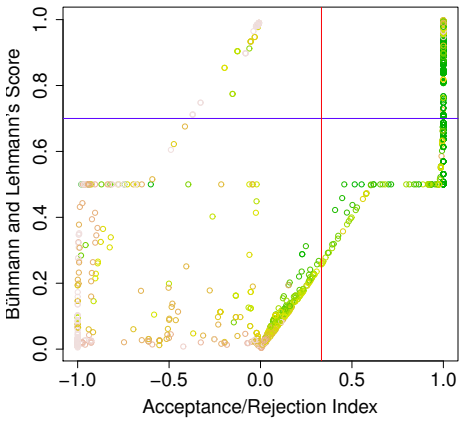

Fig. 3. A comparison of the acceptance/rejection index and the probability-based score used in [**?**] on axioms tested with a 20-minute time cap. The vertical line shows the acceptance threshold $\mathrm{ARI}(\phi) > 1/3$; the horizontal line the acceptance threshold of 0.7 for the probabilistic score.

Altogether, testing those 644 axioms took a staggering 20,328,791,473 ms (= 235 days 6 h 53 min 11.473 s). If all tests had been time-capped to exactly 10 minutes, the total elapsed time would have been "just" 182,481,760 ms (= 2 days 2 h 41 m 21.76 s), i.e., less than 0.9% of the actual time, with a two orders of magnitude speedup!

C. Results with Time Capping

Based on these observations, we decided to fix to 20 min (i.e., twice the time it took to test 99% of the accepted axioms on the more powerful machine) the threshold to time-cap the SPARQL queries to compute $u^{+}_{C \sqsubseteq D}$ and $u^{-}_{C \sqsubseteq D}$ in order to decide whether to accept or reject a candidate axiom $C \sqsubseteq D$.

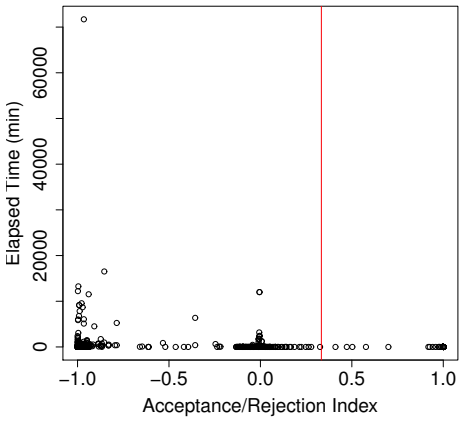

Fig. 4. Plot of the time taken for testing the systematically generated `SubClassOf` axioms without time capping as a function of ARI. The vertical line shows the acceptance threshold $\mathrm{ARI}(\phi) > 1/3$.

For this new round of experiments, we used the candidate axiom ordering heuristics in order to get as much tested axioms as possible. Thanks to the greatly reduced overhead, we were able to test 3,530 axioms at the time of writing. The results are shown in Figure **??**.

D. Qualitative Analysis of the Results

A human analysis of the results of the automatic process shows that out of the 2426 candidate axioms which were tested, 204 are questionnable, i.e. 8.3% of the results. 55 rejected axioms (scored by the system with an ARI value under the acceptance threshold) may be false negative and 149 accepted axioms (scored by the system with an ARI value above the acceptance threshold) may be false positive.

The close analysis of the possibly 55 false positive led to the following conclusions:

- 44 of them are time-capped axioms. That represents an error rate of 1.8%. If we take into account the dramatic improvement in terms of speed, this looks like a very reasonable price to pay in terms of accuracy degradation. In addition, it should be observed that, by construction, the errors are all in the same direction, i.e., some axioms which should be accepted are in fact rejected: at least, this is a conservative heuristics, since it does not generate false positives.
- Among the 11 remaining false positive, 4 of them have a positive ARI (under the acceptance threshold). This may lead to the reasonable conclusion that the candidate axioms rejected with a ARI close to the acceptance threshold should always be examined by an ontologist. The 7 remaining axioms involve very general classes as superclass, e.g., `dbo:Person`, `dbo:Product`. Their low scoring may be the result of incomplete knowledge due to the fact that people populating the DBpedia ontology will focus on more specific classes.

Conversely, the analysis of the 148 possibly false positive led to the following conclusions:

- Most of these axioms which should be rejected are inverted `subClassOf` relations between concepts (e.g. `dbo:Case` $\sqsubseteq$ `dbo:LegalCase` instead of `dbo:LegalCase` $\sqsubseteq$ `dbo:Case`). This occurs when counterexamples are missing (all instances of a class are instances of the other class too and the two axioms are positively scored).
- The acceptation of some axioms involving vague concepts is questionable. For instance, it seems that anything that can appear on a map could be typed with `gml:_Feature` and therefore many classes should be subclasses of it, but it is not clear whether this is correct or not. The same remark applies to `SubclassOf` axioms with class `dbo:Place` as superclass.
- Some axioms involve concepts used in a more general sense than it could be expected. Their acceptation is therefore dubious. It is for instance the case of `dbo:PokerPlayer` $\sqsubseteq$ `dbo:Athlete`. Its acceptation is not really a mistake in the sense that there are several other such concepts involving `dbo:Athlete`, e.g. `dbo:FigureSkater` $\sqsubseteq$ `dbo:Athlete`. These axioms are acceptable when considering `dbo:Athlete` in its general sense.
- Other questionable axioms are those involving a concept having at least two senses, e.g `dbo:Library` designating both a building and an institution. The joint acceptation of axioms `schema:Library` $\sqsubseteq$ `dbo:Organisation` and `schema:Library` $\sqsubseteq$ `dbo:Place` is not satisfactory.
- Other questionable axioms are those involving a concept both used as a zoological class name, a taxon, and therefore marked as subclass of `dbp:Species`, and as a set of animals, and therefore subclass of `dbo:Animal` and `dbo:Eukaryote`. This is for instance the case of `dbo:Insect`.
- The same confusion between the instance level and the ontological level explain that most of the axioms involving `skos:Concept` should be rejected, e.g., `dbo:Activity` $\sqsubseteq$ `skos:Concept` or `dbo:Train` $\sqsubseteq$ `skos:Concept`.

To sum up, the high scoring of most of the candidate axioms which should be rejected is due to misconceptions in the DBpedia RDF base, misuses of the DBpedia ontology by people populating it.

VII. Conclusion

We have presented a possibilistic axiom scoring heuristics which is a viable alternative to statistics-based heuristics. We have tested it by applying it to the problem of testing `SubClassOf` axioms against the DBpedia database. We have also proposed additional heuristics to greatly reduce its computational overhead, consisting of setting a time-out on

the test of each axiom and ordering the candidate axioms according to their score in order to optimize the number of axioms tested in a given time period.

Our results, albeit preliminary, strongly support the validity of our hypothesis that it is possible to alleviate the computation of the ARI without loosing too much in terms of accuracy.

In addition, the qualitative analysis of the results confirm the interest of using axiom scoring heuristics like ours not only to learn axioms from the LOD, but also to drive the validation and debugging of ontologies and RDF datasets.