Are All People Married?
Determining Obligatory Attributes in Knowledge Bases

Jonathan Lajus
Telecom ParisTech
Paris, France
jlajus@telecom-paristech.fr

Fabian M. Suchanek
Telecom ParisTech
Paris, France
suchanek@telecom-paristech.fr

ABSTRACT

An attribute is obligatory for a class in a Knowledge Base (KB), if all instances of the class have the attribute in the real world. For example, *hasBirthDate* is an obligatory attribute for the class *Person*, while *hasSpouse* is not. In this paper, we propose a new way to model incompleteness in KBs. From this model, we derive a method to automatically determine obligatory attributes – using only the data from the KB. Our algorithm can detect such attributes with a precision of up to 90%.

KEYWORDS

Knowledge Bases, Completeness, Classes, Attributes

ACM Reference Format:
Jonathan Lajus and Fabian M. Suchanek. 2018. Are All People Married? Determining Obligatory Attributes in Knowledge Bases. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186010

1 INTRODUCTION

Recent years have seen the rise of large knowledge bases (KBs). These include, among others, YAGO, Wikidata, DBpedia, BabelNet, and NELL on the academic side, and Google's Knowledge Vault and Microsoft's Satori on the industrial side. These KBs contain millions of entities (such as cities, universities, or famous people), and billions of facts about them (such as which city is located in which country, or which scientist works at which university). The KBs find applications in information retrieval, machine translation, and question answering.

The usefulness of these applications depends on the data quality of the knowledge base. One important dimension of quality is the correctness of the data. But there is another important dimension: the completeness of the data – i.e., whether or not a statement about an entity is missing from the KB. Data completeness affects queries about cardinalities, about existence, and about top-ranked entities. For example, if the population of Tokyo is missing from the KB, then a query about the top-10 most populous cities in the world will return a factually wrong result.

If we knew that every city has to have a population, we could know that the reason for Tokyo's missing population is not that Tokyo does not have a population in the real world, but that the number was not added to the KB. We could thus alert the user that the data on which the query is computed is known to be incomplete. We say that the population is an *obligatory attribute* for the class *city*. Not all attributes are obligatory. For example, not every city has to be the capital of a region. The same goes for other classes: Every person has to have a birth date, but not every person has to be married.

If we were able to distinguish obligatory attributes from optional ones, we could see more easily where information is missing in the KB. This, in turn, could help us qualify the answers to our queries. Several approaches allow querying incomplete data, if the degree of completeness is known [13, 15, 18]. The obligatory attributes can also help the designers of the knowledge base focus their effort on completing the data. For example, collaborative knowledge bases such as Wikidata could ask contributors specifically for the obligatory attributes of a new entity. Finally, the obligatory attributes can give semantics to classes. For example, the characteristics of actors is that they act in a movie. Such information can help decide whether an entity belongs to a class or not, it can guide the process of taxonomy design, and it can help define schema constraints [11]. We note that even obligatory attributes with a few counter-examples would be helpful for these goals. For example, it is good to know that people generally have a nationality – even if there are some people who do not have one. Our goal is to find the rule rather than the exception.

It is not easy to determine whether an attribute is obligatory or not. Today's KBs contain not just people and cities, but literally hundreds of thousands of other classes. They also contain hundreds, if not thousands of attributes. It is thus infeasible to specify the obligatory attributes manually. It is also hard to find them automatically: In YAGO, e.g., 2% of soccer players have a club – and that is an obligatory attribute for professional soccer players. At the same time, 2% of people have a spouse – and that is an optional attribute. Using the available data to determine obligatory attributes thus amounts to generalizing from a few instances to all instances of a class. This is a very difficult endeavor – even for humans. The case of KBs is even more intricate, because most KBs do not explicitly say that a statement does not hold in reality. For example, the KBs do not say that Pope Francis is *not married*. Rather, they operate under the Open World Assumption: A statement may be missing from the KB either because it was not added, or because it does not hold in reality. Thus, we find ourselves with the task of generalizing from a few instances in the absence of counter-examples.

In this paper, we present methods that can detect obligatory attributes automatically. Our key idea is to use the class hierarchy: Most modern KBs contain extensive class hierarchies (YAGO, e.g., contains 650,000 classes; DBpedia and Wikidata have manually

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8.
https://doi.org/10.1145/3178876.3186010

designed taxonomies). And yet, the KBs use the class hierarchy mainly to specify domain and range constraints. They do not exploit the semantics of the hierarchy any further. Our idea is to make use of the classes to determine obligatory attributes. More precisely, our contributions are as follows:

- a formal definition of the problem of obligatory attributes
- a probabilistic model for the incompleteness of a KB
- an algorithm that can determine obligatory attributes automatically
- extensive experiments on different datasets with different competitors, showing that obligatory attributes can be detected with a precision of up to 90%.

This paper is structured as follows. We first discuss related work in Section 2 and the preliminaries in Section 3. Then we present the formal definition of our problem in Section 4, and our approach in Section 5. Section 6 presents experiments, before Section 7 concludes.

2 RELATED WORK

Query Completeness. Much recent work [13, 15, 18] has investigated the completeness of queries when the completeness of the data is known. These approaches are orthogonal to our work, which aims to establish whether the data is complete in the first place.

Measuring Incompleteness. Several studies have confirmed that KBs are incomplete. A watermarking study [20] reports that 69%–99% of instances in YAGO and DBpedia lack at least one property that other entities in the same class have. In Freebase, 71% of people have no known place of birth, and 75% have no known nationality [3]. Wikidata is aware of the problem of incompleteness, and has developed tools to specify completeness [4], as well as tools to manually add completeness information [2]. Unlike our work, these approaches do not aim at determining completeness automatically.

Determining Incompleteness. Closest to our work, several approaches have recently taken to measure the incompleteness in knowledge bases [7, 19]. However, these works determine whether a particular subject (such as Emmanuel Macron) is incomplete with respect to a particular attribute (such as *birthDate*). Our work, in contrast, aims at determining whether an attribute is obligatory or not for a given class. It thus operates on the schema level.

Schema Mining. Other work has investigated the more general problem of schema mining [1, 8, 10, 17, 22]. The work of [17] mines domain and range constraints for relations. Our work, in contrast, mines relations that are obligatory for classes. [10] uses machine learning to find OWL class descriptions. However, they rely on negative facts given by the user, or on prior knowledge about the schema – while we require none of these. [22] mines Horn rules on a KB. However, this approach is not targeted towards sparse obligatory attributes. We use it as a baseline in our experiments. [8] also mines Horn rules, and can deal with sparse data. At the same time, it cannot mine rules with existential variables in the head. Any such rule would trivially have a confidence of 100% in their model, because the model makes the Partial Completeness Assumption. Another work [1] mines definitions of classes. This approach comes closer to our goal, but is not exactly targeted towards obligatory attributes. We use such an approach as a baseline in our experiments.

3 PRELIMINARIES

Knowledge Bases. We are concerned with KBs such as YAGO, DBpedia, and Wikidata. These use a set $\mathcal{I}$ of instances (such as *Macron* or the year *2017*)[1] and a set $\mathcal{P}$ of property names (such as *presidentOf*). We assume that $\mathcal{P}$ contains for every $p \in \mathcal{P}$ also its inverse p^-. Furthermore, in all of the following, we assume that $\mathcal{I}$ and $\mathcal{P}$ are fixed and global sets. In this context, a KB can be seen as a set $K \subseteq \mathcal{I} \times \mathcal{P} \times \mathcal{I}$ of facts (such as $\langle Macron, presidentOf, France \rangle$). Each fact consists of a subject $s \in \mathcal{I}$, a property $p \in \mathcal{P}$, and an object $o \in \mathcal{I}$, and we write it as $p(s, o)$. We assume that for every $p(s, o) \in K$, we also have $p^-(o, s) \in K$. A property p is a function in a KB K, if it has at most one object for each subject. Each KB K also defines a set $C_K \subseteq 2^{\mathcal{I}}$ of named classes (such as *Person* or *President*).

Ideal KB. For our problem, we consider a (hypothetical) ideal KB $\mathcal{W}$, which contains all facts of the real world. With this, our work is in line with the other work in the area [7, 13, 15, 18, 19], which also assumes an ideal KB. The problems of determining how such a KB could look are discussed in [19].

A knowledge base K is *correct* if $K \subseteq \mathcal{W}$ and it is *complete* if $\mathcal{W} \subseteq K$. The assumption that the KB is complete is called the *Closed-World Assumption (CWA)*:

$$\forall r, a, b : r(a, b) \notin K \Rightarrow r(a, b) \notin \mathcal{W} \qquad (1)$$

The CWA is too strong in practice. The *Partial Completeness Assumption (PCA)* [8] states that if a KB K knows a fact about an instance, then it knows all facts with the same property about the instance:

$$\forall r, a, b, b' : r(a, b) \in K \wedge r(a, b') \notin K \Rightarrow r(a, b') \notin \mathcal{W} \qquad (2)$$

Finally, the *Open-World Assumption (OWA)* states that nothing follows from the absence of a fact in the KB, i.e., the absence of evidence is not evidence of absence.

Generalization Rules. The *subject set* p_K of a property p in a knowledge base K is the set of all instances that have the property p in K: $p_K = \{x | \exists y : p(x, y) \in K\}$. A *generalization rule* for a KB K is a formula of the form $A \subseteq B$, where A and B are classes of K, subject sets of K, or intersections thereof. For example, if $President_{\mathcal{W}}$ is the class of presidents in the real world, then the following generalization rule says that all presidents are presidents of some country:

$$President_{\mathcal{W}} \subseteq presidentOf_{\mathcal{W}}$$

With this, we can already make a simple observation:

Proposition 1 (Heredity). *In any KB K, for every class $c_K \in C_K$, any subclass $s_K \subseteq c_K$, and every property p: if $c_K \subseteq p_K$ then $s_K \subseteq p_K$.*

[1] For our work, we do not distinguish literals and instances.

This proposition tells us that if a generalization rule holds for a class, it also holds for all subclasses. The *confidence* of a generalization rule is defined as

$$conf(A \subseteq B) = \frac{|A \cap B|}{|A|}$$

Finally, we can make a second simple observation:

Proposition 2 (Separation). *If $c_K \subseteq p_K$ for some class c_K of some KB K and some property p, then the following holds for any class c'_K of K:*

$$conf(c_K \cap p_K \subseteq c'_K) = conf(c_K \subseteq c'_K)$$

4 MODEL

4.1 Problem Definition

Goal. In this paper, we aim to find generalization rules of the form $c_{\mathcal{W}} \subseteq p_{\mathcal{W}}$. Such a rule says that every instance of $c_{\mathcal{W}}$ must have the property p in the real world. We call p an *obligatory attribute* of the class c. For example, we aim to mine

$$Film_{\mathcal{W}} \subseteq directed^{-}_{\mathcal{W}}$$

Here, *directed*$^-$ is an obligatory attribute for the class *Film*, i.e., every film has to have a director. The difficulty is to find such a rule in $\mathcal{W}$ by looking only at the data of a given KB K. In the following, we write $c_{\mathcal{W}}$ for the class c in the real world, and c_K for the corresponding class in K.

Baseline 1. One way to find obligatory attributes is assume that the KB is complete (Closed World Assumption) and correct. Under these assumptions, we can predict that an attribute p is obligatory for a class c if and only if all instances of c have p in the KB K:

$$(c_K \subseteq p_K) \overset{?}{\Rightarrow} (c_{\mathcal{W}} \subseteq p_{\mathcal{W}})$$

This is how a rule mining system under the Closed World Assumption would proceed [22], if applied naively to our problem. In practice, however, KBs are rarely complete. They operate under the Open World Assumption. There will be hardly any property p that all instances of class c have.

Baseline 2. Another method would be to predict that an attribute p is obligatory for a class c, if the corresponding generalization rule has a confidence above a threshold θ in the KB K:

$$conf(c_K \subseteq p_K) \geq \theta \overset{?}{\Rightarrow} (c_{\mathcal{W}} \subseteq p_{\mathcal{W}})$$

For example, if more than 90% of presidents in K have the property *presidentOf*, then we would predict that *presidentOf* is an obligatory attribute for the class of presidents. The problem is that an attribute may be very prevalent without being obligatory. For example, many film directors also acted in movies – but acting in a movie is not an obligatory attribute for film directors.

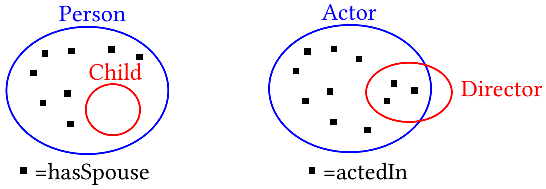

Figure 1: Examples of attributes and classes.

Baseline 3. Yet another idea (inspired by [1]) is to make use of the taxonomy. Given a property p and a KB K, we can find the lowest class c_K of the taxonomy such that nearly all instances with p fall into that class. This is motivated by the contraposition of Proposition 2. Formally, the method predicts that, for any property p, for any class c_K of a KB K, and for a threshold θ:

$$\begin{array}{ll} conf(p_K \subseteq c_K) \geq \theta & \wedge \\ \forall c'_K \subset c_K : conf(p_K \subseteq c'_K) < \theta & \end{array} \overset{?}{\Rightarrow} (c_{\mathcal{W}} \subseteq p_{\mathcal{W}})$$

This approach will work well for properties whose domain is a class, such as the property *presidentOf* with the class *President*. However, it will work less well if the attribute applies to only a subset of the class. For example, every person x with $\exists y : hasSpouse(x, y) \in K$ belongs to the class *Person*. Thus, the above confidence will be 1 for *hasSpouse* and *Person*, and the method will conclude that every person is married.

4.2 Our Approach

Our idea is based on the assumption that the incompleteness of the KB is distributed equally across all classes of the KB. If we find a class that has a very low density of an attribute (while others have a high density), then we conclude that this low density indicates that the attribute is not obligatory for that class.

As an example, consider the class of all adult people (i.e., all persons without the class of children, Figure 1). The attribute *hasSpouse* is much more prevalent in that class than in the class of children. It is unlikely that all missing *hasSpouse* facts in the class *Child* are due to incompleteness. Therefore, we can conclude that not all children are married in the real world. This means that *hasSpouse* cannot be obligatory for *Child* (and, hence, not for *Person*).

Now consider the example in Figure 1 on the right. Some instances of the class *Director* have the attribute *actedIn*. However, the density of that attribute increases if we consider the intersection of *Director* with *Actor*. Hence, *actedIn* cannot be obligatory for the class *Director*.

We will now show how to formalize this idea, and under which conditions we can guarantee that an attribute is not obligatory for a class.

4.3 Assumptions

In order to deduce formal statements about obligatory attributes in the real world from our KB K, we have to make a number of assumptions about K.

Assumption 1 (Correctness of the KB K). *Every fact that appears in the KB K also appears in the ideal KB $\mathcal{W}$: $K \subseteq \mathcal{W}$.*

This assumption basically says that the KB does not contain wrong statements. This is a strong assumption, which may not hold in practice [5, 21]. However, we use it here mainly for our theoretical model. Our experiments will show that our method works even if there is some amount of noise in the data. We make a second assumption:

Assumption 2 (Class Hierarchy of the KB K). *The classes of the KB K are correct and complete, i.e., $C_K = C_{\mathcal{W}}$.*

Again, this is a strong assumption that we use mainly for our theoretical model. In practice, three types of problems can appear. First, an instance can belong to a wrong class in the knowledge base. Second, an instance may be tagged with a too general class (e.g., *Macron* belongs to *Person*, but not to *President*). Finally, a class may be missing altogether (such as *SciencesPoAlumni* for *Macron*). These problems impact our method, as we discuss in Section 6.5. However, for Wikidata, the class system that we use appears sufficiently complete and correct to make our method work. For YAGO, the data is known to be highly accurate [21], and furthermore, the Wikipedia categories are included in the class hierarchy. This makes the hierarchy sufficiently complete for our method to work. In DBpedia, in contrast, each instance is tagged with only one class. This results in so much incompleteness that our method cannot work. For example, in DBpedia, the proportion of singers who wrote a song is higher than the proportion of song-writers who wrote a song. This indicates that many singers should actually (also) be tagged as song-writers – which they are not.

Assumption 2 allows us to omit the subscript from the classes from now on. With Assumptions 1 and 2, we can already show:

Proposition 3 (Upper bound for Confidence). *Under Assumptions 1 and 2,*

$$conf(c \subseteq p_K) \leq conf(c \subseteq p_{\mathcal{W}})$$

for any KB K, any class c, and any property p.

This proposition holds because Assumption 1 tells us that $x \in p_K$ implies $x \in p_{\mathcal{W}}$. Furthermore, Assumption 2 tells us that the classes of K are the classes of $\mathcal{W}$.

4.4 Random sampling model

Our method assumes that the incompleteness of the KB is evenly distributed. More formally, let us consider the space of all possible KBs under Assumption 1. These are $\Omega = 2^{\mathcal{W}}$. We assume a probability distribution $\mathbb{P}(\cdot)$ over this space. Given a property p and instances s, o, the statement $p(s, o) \in K$ becomes a boolean random variable defined on a KB K, and we denote it by $p(s, o)$. In the same way, the expression $|p_K \cap c|$ becomes a numerical random variable defined on a KB K, and we denote it by $|p \cap c|$. Likewise, $conf(c \subseteq p_K)$ becomes a numerical random variable, and we denote it by $conf(c \subseteq p)$. We constrain $\mathbb{P}(\cdot)$ by the following assumption:

Assumption 3 (Random sampling). *On the space of all KBs in $\Omega = 2^{\mathcal{W}}$, there exists a probability l_p for each property p such that:*

$$\forall x, y. \mathbb{P}(p(x, y)) = \begin{cases} l_p, & \text{if } p(x, y) \in \mathcal{W} \\ 0, & \text{otherwise} \end{cases}$$

The second case follows from Assumption 1. The first case states that facts with the property p in our KB come from a uniform random sampling of all true facts with property p in the real-world.

Several factors can thwart this assumption. First, the KB may be biased towards popular instances. For example, Wikipedia contains more information about American actors than about Polish actors, and people magazines are more concerned about the extra-marital affairs of actors than about the affairs of an architect. Thus, any KB that extracts from these sources will be biased. Second, the information extraction itself may have a bias. For example, several information extraction methods feed from the Wikipedia infoboxes. These infoboxes come in a number of pre-defined templates, and these templates define the properties. This entails that the presence or absence of a property in the KB depends on whether the instance happens to belong to an infobox template that defines this property or not. That said, making such simplifying assumptions about the probability distribution of facts is not unusual [3, 12, 16]. Our experiments will show that our model works also in cases where this assumption is violated to some degree.

We constrain $\mathbb{P}(\cdot)$ further by adding in the PCA (Equation 2).

Assumption 4 (PCA). *On the space of all KBs in $\Omega = 2^{\mathcal{W}}$, $\mathbb{P}(K) = 0$ if there exists a property p (which is not an inverse) and instances x, y, y' with $p(x, y) \in K, p(x, y') \notin K$ and $p(x, y') \in \mathcal{W}$.*

The PCA is a common assumption for the KBs we consider [3, 8]. It has been experimentally shown to be correct in a large number of cases [9]. Again, we need the PCA mainly for our model. Our experiments will show that our method gracefully translates to scenarios where the PCA does not hold for all properties. In particular, our method is robust enough to work also with the inverses of properties, for which the PCA usually does not hold. In the appendix, we prove:

Theorem 1 (Random sampling under PCA). *Under Assumptions 3 and 4, for each property p with probability l_p (as given by Assumption 3),*

$$\forall x : \mathbb{P}(\exists y : p(x, y)) = \begin{cases} l_p, & \text{if } x \in p_{\mathcal{W}} \\ 0, & \text{otherwise} \end{cases}$$

Theorem 1 tells us that the truth value of $\exists y : p(x, y) \in K$ for an instance x in a KB K can be seen as a random draw of a Bernoulli variable with a parameter l_p. This allows us to derive

$$|p \cap c| \sim \sum_{x \in c, x \in p_{\mathcal{W}}} \textsc{Bernoulli}(l_p) = \textsc{Binom}(|p_{\mathcal{W}} \cap c|, l_p)$$

This allows for the following proposition.

Proposition 4 (Biased estimator). *The confidence of a generalization rule in Ω follows a binomial distribution divided by a constant:*

$$conf(c \subseteq p) \sim \frac{\textsc{Binom}(|c \cap p_{\mathcal{W}}|, l_p)}{|c|}$$

Hence, the expected confidence of the rule in Ω is a biased estimator for the confidence of the rule in $\mathcal{W}$:

$$\mathbb{E}[conf(c \subseteq p)] = l_p \times conf(c \subseteq p_{\mathcal{W}})$$

This proposition confirms that, in our model, the confidence of $c \subseteq p_{\mathcal{W}}$ cannot be estimated from the data in our KB alone, as long as l_p remains unknown. The proposition also allows us to predict the behavior of Baseline 2 with parameter θ (see again Section 4.1). For a predicate p, if $\theta > l_p$, then the baseline is less likely to find all the correct classes for the predicate p, but the classes it finds have a high probability of being correct. We show in the appendix:

Proposition 5 (Unbiased estimator). *Given two classes c, c' and a property p, the expected confidence of $c \cap p \subseteq c'$ in Ω is an unbiased estimator for the confidence in $\mathcal{W}$:*

$$\mathbb{E}[conf(c \cap p \subseteq c')] = conf(c \cap p_{\mathcal{W}} \subseteq c')$$

This proposition finally establishes a link between the (expected) observed confidence in our KB and the confidence in the real world.

5 ALGORITHM

In this section, we first define our main indicator score for obligatory attributes. We then present our algorithm and propose some variations of this algorithm.

5.1 Confidence Ratio

Our main indicator score for obligatory attributes is defined as follows:

Definition 1 (Confidence ratio). *Given a KB K, a property p, and two classes c and c' with $|c \cap c'| \neq 0$ and $|c \setminus c'| \neq 0$, the confidence ratio is*

$$s_p^K(c, c') = \frac{conf(c \setminus c' \subseteq p_K)}{conf(c \cap c' \subseteq p_K)}$$

This expression compares the ratio of instances with p in $c \cap c'$ to the ratio of instances with p in $c \setminus c'$. It represents the influence of being in the class c' for the instances of a class c on p. This ratio is similar to the relative risk that is used in clinical tests. We can now make the following observation (which we prove in the appendix):

Proposition 6 (Main observation). *If p is an obligatory attribute for some class c, then for every class c' with $|c \cap c'| \neq 0$ and $|c \setminus c'| \neq 0$,*

$$\mathbb{E}[s_p(c, c')] = 1$$

Here, $s_p(c, c')$ is a random variable, and hence does not carry the K. The observation tells us that the density of p in c should not be influenced by c' if p is obligatory for c. The probability of a KB where c' influences p is low in our probability space.

Measure instability. Our confidence ratio estimate will suffer from instability when the expected number of instances with a property p in an intersection is inferior to 1. In that case, there might be no instance with the property p in the intersection, and the confidence ratio will be infinite. In practice, this happens in small intersections for highly incomplete properties. Therefore, we decide to consider only *stable classes*. Given a class c and a property p in a KB K, an intersecting class c' is stable if either the expected number of instances ($conf(c \subseteq p_K) \times |c \cap c'|$) or the actual number ($|c \cap c' \cap p_K|$) is at least 1. The same has to hold for class differences.

5.2 Algorithm

Proposition 6 allows us to make statements about a generalization rule $c \subseteq p_{\mathcal{W}}$ in the real world purely by observing an incomplete knowledge base K. All we have to do is to check the classes c' that intersect with the class c. If the ratio of instances of p_K in $c \cap c'$ is very different from the ratio in $c \setminus c'$, then it is very unlikely that $c \subseteq p_{\mathcal{W}}$ holds. Furthermore, if $s_p^K(c', c) \gg 1$, then p cannot be obligatory for $c \cap c'$. Thus, it cannot be obligatory for c and c'.

These considerations give us Algorithm 1. This algorithm takes as input a KB K, a class c, and a property p. The algorithm also uses two thresholds: θ is the margin that we allow s_p^K to deviate from 1. The larger the threshold, the more obligatory attributes the algorithm will find – and the more likely it is that some of them will be wrong. The threshold θ' is the minimum support allowed for the rule $c \subseteq p$ to be considered. In practice, we set θ' to 100, as in AMIE [8]. Our algorithm returns *false* if the generalization rule $c \subseteq p_{\mathcal{W}}$ should be rejected – either because the support is too small (Lines 1-2), or because there is a stable intersecting class c' with $s_p^K(c, c') \neq 1$ (Lines 4-5), or $s_p^K(c', c) \gg 1$ (Lines 6-7). If neither is the case, the algorithm returns *true*.

Caveat. Our algorithm will return *true* if it finds no reason to reject a class. This, however, does not necessarily mean that the attribute is obligatory in this class. In particular, our algorithm may perform poorly if there is no class where the attribute is obligatory. However, our experiments show that despite this caveat, the method works well in practice.

Algorithm 1: ObligatoryAttribute

```
  Input: KB $K$, class $c$, property $p$, threshold $\theta$,
         threshold $\theta' = 100$
  Output: true if $c \subseteq p_{\mathcal{W}}$ is predicted
1 if $|c \cap p_K| < \theta'$ then
2     return false
3 for stable class $c'$ do
4     if $|log(s_p^K(c, c'))| > log(\theta)$ then
5         return false
6     if $log(s_p^K(c', c)) > log(\theta)$ then
7         return false
8 return true
```

Example. Consider again the second example in Figure 1. On YAGO data, we obtain:

$$s_{actedIn}^{YAGO}(Director, Actor) = 0$$

This means that directors are unlikely to act in a movie if they are not also actors. Thus, our algorithm will reject the hypothesis that *actedIn* would be obligatory for the class *Director*. We also obtain

$$s_{actedIn}^{YAGO}(Actor, Director) = 0.77$$

Since this value is closer to 1, we understand that actors act no matter whether they are also directors or not. Hence, the class *Actor* will be accepted for thresholds θ above $\frac{1}{0.77} \approx 1.3$.

5.3 Variations

Relaxation. In practice, classes in a KB intersect only in small areas. Thus, when $s_\phi^K(c, c') \gg 1$, we decided to reject only c'. In this relaxed variant, the condition in Line 4 of Algorithm 1 becomes $log(s_p^K(c, c')) < -log(\theta)$.

Fisher's Exact Test. We also experimented the Fisher's Exact Test [6] instead of the confidence ratio. We replace the logarithm of the confidence ratio s in Line 4 of Algorithm 1 by the probability that $c \cap c'$ has higher values, and in Line 6 with the probability that it has lower values over the set of possible contingency tables with fixed marginals.

6 EXPERIMENTS

In this section, we evaluate our approach experimentally on large real-world KBs. We first evaluate our approach on YAGO, a KB for which we know that our Assumptions 1 and 2 hold by and large. Then, we submit our approach to a stress test: We run it on Wikidata, where less is known about our assumptions. Finally, we investigate how our approach could be generalized to composite classes.

6.1 Datasets

YAGO. We chose the YAGO3 knowledge base [14] for our experiments, because the data is of good quality (Assumption 1) and the taxonomy is extensive (Assumption 2). We use the facts of all instances, the full taxonomy, and the transitive closure of types. With this, our dataset contains more than 5 million instances and around 54,000 classes with more than 50 (direct or indirect) instances.

Wikidata. As a stress-test, we also evaluated our approach in Wikidata, where less is known about our Assumptions 1-4. We used the version from 2017-06-07, which contains more than 16,000 properties. This makes a manual evaluation impractical. Hence, we reduced the dataset to only people. However, all people are in only one class: *Human*. Therefore, we used the *occupation* property (P106) to define classes. For example, in Wikidata, Elvis Presley (Q303) has the occupations *FilmActor*, *Actor*, *Singer*, *Screenwriter*, *Guitarist* and *Soldier*. These occupations form their own hierarchy, which we use as class hierarchy. For example, *FilmActor* is a suboccupation of *Actor*, and thus becomes a subclass of it. This subset of Wikidata contains 1023 classes, around 1.6 million instances, and 2569 properties.

6.2 Gold Standard

Since our problem is novel, there is no previously published gold standard for it. Therefore, we had to construct a gold standard manually. For YAGO, we considered 68 properties (37 properties and their inverses), we determined the classes where more than 100 instances have the property, and we manually evaluated whether the attribute is obligatory or not. For Wikidata, we randomly selected 100 properties, and evaluated the output of each method manually. Our manual evaluation gives us an estimate for precision. Since Baseline 3 has a recall of 100% at maximal θ, we can use it to estimate our recall.

It is not always easy to determine manually whether an attribute is obligatory. For example, consider the attribute *isAffiliatedTo*. Is it obligatory for an artist to be affiliated to a museum, for a football player to be affiliated to a football club, or for people in general to be affiliated to their relatives? For our gold standard, we restricted ourselves to cases where we could clearly establish whether an attribute is obligatory or not, and removed all other cases.

Another problem arises for classes where the huge majority of instances have a particular attribute. For example, should we discard *hasNationality* as an obligatory attribute for *Person* because there exist stateless people? In such cases, we decided that the absence of the attribute is an exception to the rule that our method should not predict. Hence, we considered *hasNationality* obligatory. A related problem is that an attribute may not necessarily be obligatory for a class, but that de facto all instances have it. For example, we expect all instances of the class *RomanEmperor* to be dead by now, but what if a renewed Roman empire arises in the future? In such cases, we considered an attribute obligatory if de facto all known instances have it.

We constructed our gold standard according to these principles, and refer the reader to [19] for a more detailed discussion of such evaluations. All our datasets, as well as the gold standard and the evaluation results, are available at https://suchanek.name/work/publications/www-2018-data.

6.3 Evaluation Metric

The most intuitive way to evaluate the prediction of obligatory attributes would be to consider each predicted pair of a class and an attribute, and to compare this set to the gold standard. However, this comparison would not take into account the size of the class. For example, it is more important to predict that all organizations have a headquarters than that all Qatari ski champions have a gender, because there are many more of the former than of the latter. However, weighting each class by the number of instances causes another problem. Consider, e.g., the classes *Man* and *Woman*, which partition the class *Person* in our data[2]. If we predict that an attribute is obligatory for *Man* and for *Woman*, but not for *Person*, we would obtain a recall of only 50% – even though we predicted the attribute correctly for all instances.

To mitigate this problem, we compare, for each class c and for each property p, the actual set of predicted instances with the instances in the gold standard, i.e., we compare

$$P_p = \{x \in c | c \subseteq p_W \text{ predicted by our algorithm}\}$$

with

$$G_p = \{x \in c | c \subseteq p_W \text{ in the gold standard}\}$$

The true positives are the instances in the intersection of these sets. Then we compute the precision and recall as follows:

$$precision = \frac{\sum_p |P_p \cap G_p|}{\sum_p |P_p|}$$

$$recall = \frac{\sum_p |P_p \cap G_p|}{\sum_p |G_p|}$$

The F1-measure is computed as the harmonic mean of these.

6.4 YAGO Experiment

We ran all three baselines (Section 4) as well as our approaches (Section 5) on our YAGO dataset. Figure 2 shows the recall over the precision for each approach, with varying threshold θ.

Baseline 1. Recall that this baseline (inspired by [22]) labels an attribute as obligatory in a class, if all instances of the class have this attribute in the KB. This baseline performs like Baseline 2 at $\theta = 1$.

[2] We are talking about a property of our data, not about genders in the real world.

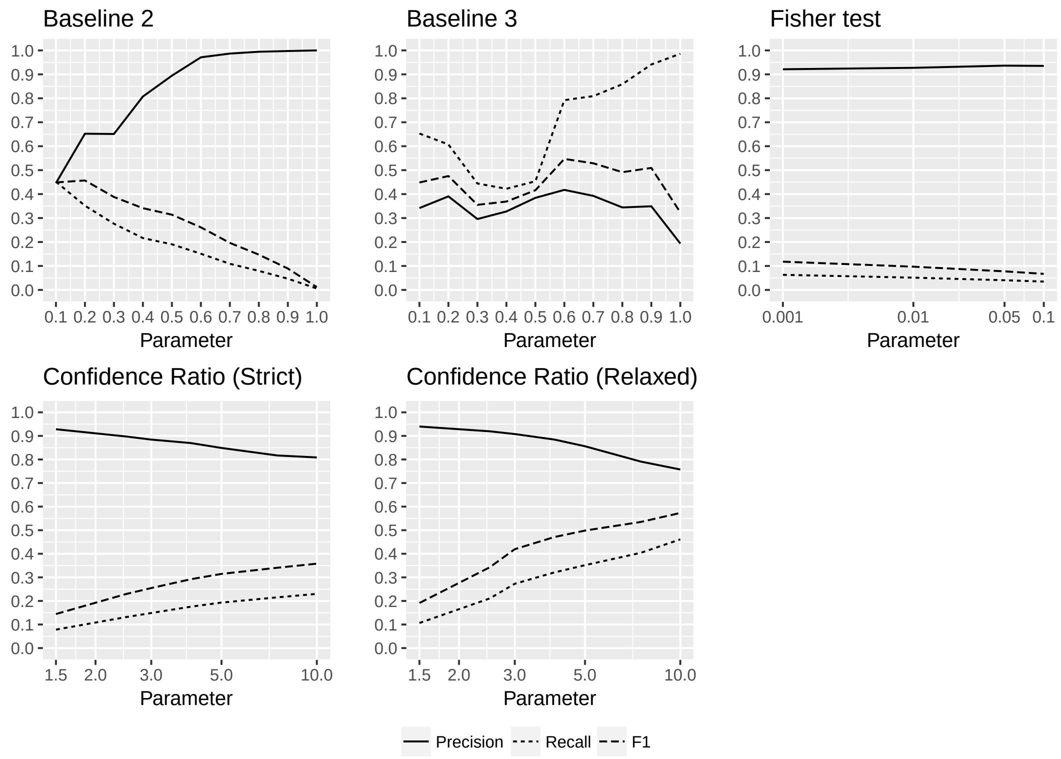

Figure 2: Influence of θ on precision, recall and F1 for the different algorithms
Baseline 1 is Baseline 2 for parameter $\theta = 1$

Unsurprisingly, it has a very good precision, but a very bad recall: Only very few attributes (such as *label*) appear on all instances.

Baseline 2. This baseline relaxes Baseline 1 by labeling an attribute as obligatory if it is very prevalent in the class. For smaller θ, this method has a better recall than Baseline 1. However, it cannot exceed an F1 value of 45%. This is because there is no global threshold θ that would work well for all attributes. The baseline will work better if the KB is more complete. At the same time, the more complete the KB is, the less novel information there is to predict.

Baseline 3. This baseline (inspired by [1]) considers an attribute obligatory for a class if the vast majority of instances with that attribute fall in that class. The somewhat unusual curve comes from the fact that the baseline chooses the deepest class in the taxonomy where the target rule holds. While the method achieves slightly better F1 values (55%), its precision never exceeds 42%.

Confidence Ratio (Strict). This is our approach, based on the ratio of an attribute in a class and its intersections with the other classes (Algorithm 1). Different from Baseline 2, it delivers a very high precision (always > 80%) – at the expense of somewhat lower recall. The best F1 measure is 37%.

Confidence Ratio (Relaxed). The relaxed variant of our method is less conservative. It trades off precision for higher recall. Indeed, we see that recall increases steadily with growing θ, while precision decreases gently. This allows for very good trade-offs between the two, with the maximum F1 value easily surpassing 55%. It is thus our method of choice.

Fisher's Test. This variation of our approach aims to make the Confidence Ratio less vulnerable to small data sizes. This is indeed what happens. However, the method errs on the side of caution: it has a very good precision (always > 90%), but a mediocre recall. Hence, the best F1 value is quite low (12%). To increase this recall, the significance level of this test would have to be increased by a factor of several orders of magnitude, which would defy its purpose. The method should thus be seen as a stable, but inherently precision-oriented method.

Comparison. Figure 3 plots precision and recall for each of the methods across the spectrum of parameter values.[3] Baseline 3 achieves the highest recall. However, its precision never exceeds 42%, which makes the method unusable in practice. On the other side of the spectrum, Baseline 2 offers very good precision – but it cannot achieve good recall. Our relaxed confidence ratio occupies a sweet spot between the two: a precision between 75% and 95%, at a recall of 45% and 10%, respectively. It thus dominates the other methods in the mid-range between good recall and good precision.

[3] Different values for θ can give the same combination of precision and recall, whence the "loop" of Baseline 3.

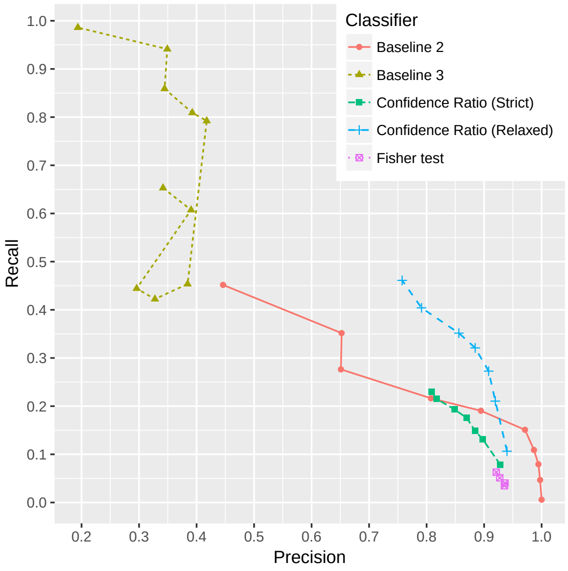

Figure 3: Precision and Recall on YAGO

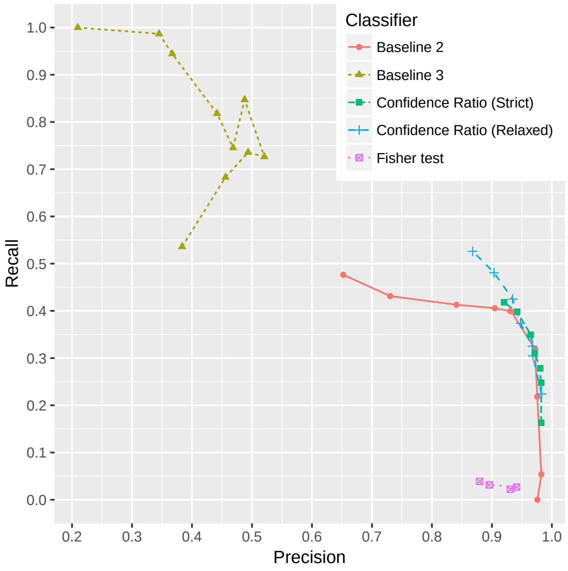

Figure 4: Stress test: Precision and Recall on Wikidata

Completeness of the attributes. By identifying classes in which an attribute is obligatory, our method identifies the entities that should have this attribute. If we compute the proportion of these entities that actually have the attribute in the data, we get an approximation of the completeness of the data. Table 1 shows the estimated completeness of the data according to different methods: the gold standard, Baseline 2 at different thresholds, and our method at different thresholds. We show 3 attributes that are obligatory in certain classes, and the deviation from the gold standard across all attributes. The small deviation for our method shows that we can approximate the real completeness quite well.

We can now also algorithmically answer the question raised in the title of this paper: No, not all people are married. Our method finds that *isMarriedTo* is an optional attribute for the class *Person*. However, marriage is obligatory for the classes *Spouse* and *RoyalConsort*.

		Baseline 2		CR (Relaxed)	
Attribute	Gold Standard	0.5	0.9	1.5	3
hasGender	0.58	0.51	0.91	0.79	0.58
wasBornIn	0.14	0.47	0.93	0.46	0.25
isMarriedTo	0.57	0.51	0.93	0.59	0.23
Avg-Squared error to GS (all p)		0.21	0.59	0.17	0.08

Table 1: Approximation of completeness of attributes

6.5 Wikidata Experiment

As a stress test, we also evaluated our method on Wikidata, where less is known about our assumptions. Figure 4 shows our results. We first note that all methods exhibit a similar behavior to the YAGO experiment. Baseline 3 has high recall, low precision ($< 55\%$) and remains unstable. Baseline 2 performs well, with a precision of 97% and a recall of 33% for threshold $\theta = 0.7$. This indicates that some of the properties in our data are already highly complete. Our method performs similarly to Baseline 2 in the precision range of 97%. However, in precision range of 93%, it has a higher recall than Baseline 2.

6.6 Artificial Classes

In the following two experiments, we investigate how our algorithm performs on artificially constructed classes. For this purpose, we constructed classes that depend on the facts in our KB. Since the facts are incomplete, these classes are incomplete, too, and Assumption 2 no longer holds.

Life Expectancy. We construct artificial classes for all people born before a certain decade t:

$$C_t = \{x | \exists y : birthDate(x, y) \wedge y < t\}$$

These classes form a taxonomy:

$$C_t \subseteq C_{t+10}$$

In this way, we generated the classes C_t for $t = 1700, 1710, \ldots, 2020$ in YAGO. We can now mine obligatory attributes also on these artificial classes. In particular we mine the generalization rule

$$C_t \subseteq deathDate_{\mathcal{W}}$$

Table 2 shows the t for which the rule holds, according to our relaxed algorithm. We see that for a conservative $\theta < 3$ (which delivered high precision also in the previous experiments), we get again very good estimates for t. As θ increases, our method starts to believe that all people (even younger ones) should have a death date – as expected. This experiment shows that our approach has the potential to mine obligatory attributes even on intensionally defined classes.

θ up to	1.3	2.5	5.0	9.5	10	20	30
t mined	1920	1930	1940	1950	1960	1970	1980

Table 2: Life expectancy experiment

Cardinality experiment. To illustrate the effect of more fine-grained classes on our algorithm, we constructed for every attribute p and every number n the classes p_{n+} as the set of entities having more than n objects for attribute p:

$$p_{n+} = \{x | \exists_{>n} y : p(x, y)\}$$

These classes form a taxonomy, with $p_{(n+1)+} \subseteq p_{n+}$. We added these classes to YAGO and we ran our algorithm with a small modification: for an attribute p, we never considered any class p_{n+} for the intersections. This is to exclude trivial rules of the form $p_{n+} \subseteq p$. In the end, our algorithm with threshold $\theta = \log(3)$ outputs 248 rules with cardinality classes. The new classes produce two effects (exemplified in Table 3): First, the algorithm now overfits and deduces that a birth date would be obligatory (only) for certain subclasses of people. Second, the algorithm can now make very fine grained predictions about the real world. Thus, it predicts that anyone who has more than 8 children in the KB is most likely married in the real world. We see this as an encouragement to investigate the potential of artificially constructed classes for future work.

Overfitting rules	
$created_{80+}$	$\Rightarrow$ *wasBornIn*
$playsFor_{14+}$	$\Rightarrow$ *wasBornIn*
$edited_{6+}$	$\Rightarrow$ *wasBornIn*
Fine-grained Predictions	
$hasChild_{8+}$	$\Rightarrow$ *isMarriedTo*
$actedIn_{49+}$	$\Rightarrow$ *isMarriedTo*
$isMarriedTo_{3+}$	$\Rightarrow$ *hasChild*
$actedIn_{24+}$	$\Rightarrow$ *hasChild*

Table 3: Cardinality experiment

7 CONCLUSION

In this paper, we have introduced the novel problem of mining obligatory attributes from knowledge bases. This is the problem of determining whether all instances of a given class have a given attribute in the real world – while all we have at our disposal is an incomplete KB. We have developed a new way to model the incompleteness of a KB statistically. From this model, we were able to derive the necessary conditions for obligatory attributes. Based on this, we have proposed an algorithm that can mine such attributes with a precision of up to 92%.

For future work, we plan to study generalizations of our approach to artificially constructed classes, to rules with negation, or to rules with more complex general statements. We see this line of work as a first step towards deriving statistical confidence about real-world rules from an incomplete knowledge base. We hope that this research can deliver insights about the completeness of existing KBs, and that it can help making KBs ever more complete in the future.

Acknowledgments. This research was partially supported by the grant ANR-16-CE23-0007-01 ("DICOS"). We would also like to thank the four anonymous reviewers for their helpful insights and comments.

A PROOFS OF THE THEORETICAL RESULTS

Theorem 1 (Random sampling under PCA). *Under Assumptions 3 and 4, for each property p with probability l_p (as given by Assumption 3),*

$$\forall x : \mathbb{P}(\exists y : p(x, y)) = \begin{cases} l_p, & \textit{if } x \in p_{\mathcal{W}} \\ 0, & \textit{otherwise} \end{cases}$$

Proof. Let us assume that $\nexists y : p(x, y) \in \mathcal{W}$. Then Assumption 3 tells us that $\forall y : \mathbb{P}(p(x, y)) = 0$, and thus the second case of our theorem holds. Now let us consider the case where there exists z with $p(x, z) \in \mathcal{W}$. For any KB K, if K contains $p(x, z)$, then $\exists y : p(x, y)$ in K. If $\exists y : p(x, y)$ in K, then Assumption 4 tells us that K either contains $p(x, z)$, or else $\mathbb{P}(K) = 0$. As $p(x, z)$ and $\exists y : p(x, y)$ coincide on any KB K such that $\mathbb{P}(K) > 0$, $\mathbb{P}(\exists y : p(x, y)) = \mathbb{P}(p(x, z))$. Assumption 3 tells us that $\mathbb{P}(p(x, z)) = l_p$, which proves the first case of our theorem. □

Proposition 5 (Unbiased estimator). *Given two classes c, c' and a property p, the expected confidence of $c \cap p \subseteq c'$ in Ω is an unbiased estimator for the confidence in $\mathcal{W}$:*

$$\mathbb{E}[conf(c \cap p \subseteq c')] = conf(c \cap p_{\mathcal{W}} \subseteq c')$$

Proof. Consider the following two random variables:

$$X = |p \cap c \cap c'| \sim \textsc{Binom}(|p_{\mathcal{W}} \cap c \cap c'|, l_p)$$

$$Y = |p \cap c \setminus c'| \sim \textsc{Binom}(|p_{\mathcal{W}} \cap c \setminus c'|, l_p)$$

Let n be a natural number. X given $X+Y = n$ follows a hypergeometric distribution with parameters $(|p_{\mathcal{W}} \cap c|, n, conf(c \cap p_{\mathcal{W}} \subseteq c'))$. Then, $\mathbb{E}[X \mid X + Y = n] = n \times conf(c \cap p_{\mathcal{W}} \subseteq c')$. This implies

$$\mathbb{E}[conf(c \cap p \subseteq c') \mid X + Y = n] = conf(c \cap p_{\mathcal{W}} \subseteq c')$$

This is true for all estimates n. □

Proposition 6 (Main observation). *If p is an obligatory attribute for some class c, then for every class c' with $|c \cap c'| \neq 0$ and $|c \setminus c'| \neq 0$,*

$$\mathbb{E}[s_p(c, c')] = 1$$

Proof. We first show that

$$s_p^K(c, c') = \frac{|c \cap c'|}{|c \setminus c'|} \times \frac{1 - conf(c \cap p_K \subseteq c')}{conf(c \cap p_K \subseteq c')}$$

Proposition 5 then tells us that

$$\mathbb{E}[s_p(c, c')] = \frac{|c \cap c'|}{|c \setminus c'|} \times \frac{1 - conf(c \cap p_{\mathcal{W}} \subseteq c')}{conf(c \cap p_{\mathcal{W}} \subseteq c')}$$

Proposition 2 then implies

$$\mathbb{E}[s_p(c, c')] = \frac{|c \cap c'|}{|c \setminus c'|} \times \frac{1 - conf(c \subseteq c')}{conf(c \subseteq c')} = 1$$ □

REFERENCES

[1] Mehwish Alam, Aleksey Buzmakov, Victor Codocedo, and Amedeo Napoli. Mining Definitions from RDF Annotations Using Formal Concept Analysis. In *IJCAI*, 2015.

[2] F. Darari, S. Razniewski, R. Prasojo, and W. Nutt. Enabling fine-grained RDF data completeness assessment. In *ICWE*, 2016.

[3] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.

[4] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandecic. Introducing Wikidata to the linked data web. In *ISWC*, 2014.

[5] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of DBpedia, Freebase, Opencyc, Wikidata, and Yago. *Semantic Web*, 2016.

[6] R. A. Fisher. On the interpretation of chi square from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1), Jan 1922.

[7] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. Predicting Completeness in Knowledge Bases. In *WSDM*, 2017.

[8] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, 2013.

[9] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Fast Rule Mining in Ontological Knowledge Bases with AMIE+ . In *VLDBJ*, 2015.

[10] Sebastian Hellmann, Jens Lehmann, and Soeren Auer. Learning of OWL class descriptions on very large knowledge bases. *Int. J. Semantic Web Inf. Syst.*, 5, 04 2009.

[11] Holger Knublauch and Dimitris Kontokostas. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017. https://www.w3.org/TR/2017/REC-shacl-20170720/.

[12] Ni Lao, Tom M. Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, 2011.

[13] A. Y. Levy. Obtaining complete answers from incomplete databases. In *VLDB*, 1996.

[14] Farzaneh Mahdisoltani, Joanna Asia Biega, and Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*, 2015.

[15] A. Motro. Integrity = Validity + Completeness. *TODS*, 1989.

[16] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types . In *EMNLP*, 2012.

[17] Heiko Paulheim and Christian Bizer. Type inference on noisy RDF data. In *ISWC*, 2013.

[18] S. Razniewski, F. Korn, W. Nutt, and D. Srivastava. Identifying the extent of completeness of query answers over partially complete databases. In *SIGMOD*, 2015.

[19] S. Razniewski, F. M. Suchanek, and W. Nutt. But what do we actually know? In *AKBC workshop*, 2016.

[20] F. M. Suchanek, D. Gross-Amblard, and S. Abiteboul. Watermarking for Ontologies. In *ISWC*, 2011.

[21] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, 2007.

[22] Johanna Völker and Mathias Niepert. Statistical schema induction. In *ESWC*, 2011.