# Are Names Meaningful? Quantifying Social Meaning on the Semantic Web

Steven de Rooij, Wouter Beek, Peter Bloem, Frank van Harmelen, and Stefan Schlobach
{s.rooij,w.g.j.beek,p.bloem,frank.van.harmelen,k.s.schlobach}@vu.nl

Dept. of Computer Science, VU University Amsterdam, NL

**Abstract.** According to its model-theoretic semantics, Semantic Web IRIs are individual constants or predicate letters whose names are chosen arbitrarily and carry no formal meaning. At the same time it is a well-known aspect of Semantic Web *pragmatics* that IRIs are often constructed mnemonically, in order to be meaningful to a human interpreter. The latter has traditionally been termed 'social meaning', a concept that has been discussed but not yet quantitatively studied by the Semantic Web community. In this paper we use measures of mutual information content and methods from statistical model learning to quantify the meaning that is (at least) encoded in Semantic Web names. We implement the approach and evaluate it over hundreds of thousands of datasets in order to illustrate its efficacy. Our experiments confirm that many Semantic Web names are indeed meaningful and, more interestingly, we provide a quantitative lower bound on how much meaning is encoded in names on a per-dataset basis. To our knowledge, this is the first paper about the interaction between social and formal meaning, as well as the first paper that uses statistical model learning as a method to quantify meaning in the Semantic Web context. These insights are useful for the design of a new generation of Semantic Web tools that take such social meaning into account.

## 1 Introduction

The Semantic Web constitutes the largest logical database in history. Today it consists of at least tens of billions of atomic ground facts formatted in its basic assertion language RDF. While the meaning of Semantic Web statements is formally specified in community Web standards, there are other aspects of meaning that go beyond the Semantic Web's model-theoretic or formal meaning [12].

Model theory states that the particular IRI chosen to identify a resource has no semantic interpretation and can be viewed as a black box: "urirefs are treated as logical constants."[1] However, in practice IRIs are not chosen randomly, and similarities between IRIs are often used to facilitate various tasks on RDF data,

[1] See `https://www.w3.org/TR/2002/WD-rdf-mt-20020429/#urisandlit`

with ontology alignment being the most notable, but certainly not the only one. Our aim is to evaluate (a lower bound on) the amount of information the IRIs carry about the structure of the RDF graph.

**A simple example:** Taking RDF graphs $G$ (Listing 1.1) and $H$ (Listing 1.2) as an example, it is easy to see that these graphs are structurally isomorphic up to renaming of their IRIs. This implies that, under the assumption that IRIs refer to objects in the world and to concepts, graphs $G$ and $H$ denote the same models.[2]

**Listing 1.1.** Serialization of graph $G$.

```
abox:item1024  rdf:type     tbox:Tent    .
abox:item1024  tbox:soldAt  abox:shop72  .
abox:shop72    rdf:type     tbox:Store   .
```

**Listing 1.2.** Serialization of graph $H$.

```
fy:jufn1024  pe:ko9sap_     fyufnt:Ufou   .
fy:jufn1024  fyufnt:tmffqt  fy:aHup       .
fy:aHup      pe:ko9sap_     fyufnt:70342  .
```

Even though graphs $G$ and $H$ have the same formal meaning, an intelligent agent – be it human or not – may be able to glean more information from one graph than from the other. For instance, even a human agent that is unaware of RDF semantics may be inclined to think that the object described in graph $G$ is a tent that is sold in a shop. Whether or not the constant symbols `abox:item1024` and `fy:jufn1024` denote a tent is something that cannot be glanced from the formal meaning of either graph. In this sense, graph $G$ may be said to purposefully mislead a human agent in case it is not about a tent sold in a shop but about a dinosaur trodding through a shallow lake. Traditionally, this additional non-formal meaning has been called *social meaning* [11].

While social meaning is a multifarious notion, this paper will only be concerned with a specific aspect of it: *naming*. Naming is the practice of employing sequences of symbols to denote concepts. Examples of names in model theory are individual constants that denote objects and predicate letters that denote relations. *The claim we want to substantiate in this paper is that in most cases names on the Semantic Web are meaningful.* This claim cannot be proven by using the traditional model-theoretic approach, according to which constant symbols and predicate letters are arbitrarily chosen. Although this claim is widely recognized among Semantic Web practitioners, and can be verified after a first glance at pretty much any Semantic Web dataset, there have until now been no attempts to quantify the *amount* of social meaning that is captured by current naming

[2] Notice that the official semantics of RDF [13] is defined in terms of a Herbrand Universe, i.e., the IRI `dbr:London` does not refer to the city of London but to the syntactic term `dbr:London`. Under the official semantics graphs $G$ and $H$ are therefore *not* isomorphic and they do *not* denote the same models. The authors believe that RDF names refer to objects and concepts in the real world and not (solely) to syntactic constructs in a Herbrand Universe.

practices. We will use mutual information content as our quantitative measure of meaning, and will use statistical model learning as our approach to determine this measure across a large collection of datasets of varying size.

In this paper we make the following contributions:

1. We prove that Semantic Web names are meaningful.
2. We quantify *how much* meaning is (at least) contained in names on a per-dataset level.
3. We provide a method that scales comfortably to datasets with hundreds of thousands of statements.
4. The resulting approach is implemented and evaluated on a large number of real-world datasets. These experiments do indeed reveal substantial amounts of social meaning being encoded in IRIs.

To our knowledge, this is the first paper about the interaction between social and formal meaning, as well as the first paper that uses statistical model learning as a method to quantify meaning in the Semantic Web context. These insights are useful for the design of a new generation of Semantic Web tools that take such social meaning into account.

## 2 Method

**RDF graphs & RDF names** An RDF graph $G$ is a set of atomic ground expressions of the form $p(s, o)$ called triples and often written as $\langle s, p, o \rangle$, where $s$, $p$ and $o$ are called the subject, predicate and object term respectively. Object terms $o$ are either IRIs or RDF literals, while subject and predicate terms are always IRIs. In this paper we are specifically concerned with the social meaning of RDF names that occur in the subject position of RDF statements. This implies that we will not consider unnamed or blank nodes, nor RDF literals which only appear in the object position of RDF statements [5].

**IRI meaning proxies** What IRIs on the Semantic Web mean is still an open question, and in [11] multiple meaning theories are applied to IRI names. However, none of these different theories of meaning depend on the IRI trees, neither their structure nor their string-labels. Thus, whatever theory of IRIs is discussed in the literature, it is always independent of the string (the name) that makes up the IRI. The goal of this paper is to determine if there are some forms of meaning for an IRI that correlate with the choice of their name (as defined by the IRI trees above).

For this purpose, we will use the same two "proxies" for the meaning of an IRI that were used in [10]. The first proxy for the meaning of an IRI $s$ the *type-set* of $x$: the set of classes $Y^C(x)$ to which an IRI $x$ belongs. The second proxy for the meaning of an IRI $x$ is the *property-set* of $x$: the set of properties $Y^P(x)$ that are applied to IRI $x$. Using the standard intension ($Int$) and extension ($Ext$)

functions for RDF semantics [13] we define these proxies in the following way:

$$\textbf{Type-set: } Y^C(x) := \{c \mid \langle Int(x), Int(c)\rangle \in Ext(Int(\texttt{rdf:type}))\}$$
$$\textbf{Property-set: } Y^P(x) := \{p \mid \exists o.\, \langle Int(x), Int(o)\rangle \in Ext(Int(p))\}$$

Notice that every subject term has a non-empty property-set (every subject term must appear in at least one triple) but some subject terms may have an empty type-set (in case they do not appear as the subject of a triple with the `rdf:type` predicate). We will simply use $Y$ in places where both $Y^C$ and $Y^P$ apply. Since we are interested in relating names to their meanings we will use $X$ to denote an arbitrary IRI name and will write $\langle X, Y\rangle$ for a pair consisting of an arbitrary IRI name and either of its meaning proxies.

**Mutual information** Two random variables $X$ and $Y$ are *independent iff* $P(X, Y) = P(X) \cdot P(Y)$ for all possible values of $X$ and $Y$. Mutual information $I(X; Y)$ is a measure of the *dependence* between $X$ and $Y$, in other words a measure of the discrepancy between the joint distribution $P(X, Y)$ and the product distribution $P(X) \cdot P(Y)$:

$$I(X; Y) = E[\log P(X, Y) - \log P(X) \cdot P(Y)],$$

where $E$ is the expectation under $P(X, Y)$. In particular, there is no mutual information between $X$ and $Y$ (i.e. $I(X; Y) = 0$) when $X$ and $Y$ are independent, in which case the value of $X$ carries no information about the value of $Y$ or vice versa.

**Information and codes** While the whole paper can be read strictly in terms of probability distributions, it may be instructive to take an information theoretical perspective, since information theory inspired many of the techniques we use. Very briefly: it can be shown that for any probability distribution $P(X)$, there exists a prefix-free encoding of the values of $X$ such that the codeword for a value $x$ has length $-\log P(x)$ bits (all logarithms in this paper are base-2). "Prefix-free means" that no codeword is the prefix of another, and we allow non-integer codelengths for convenience. The inverse is also true: for every prefix free encoding (or "code") for the values of $X$, there exists a probability distribution $P(X)$, so that if element $x$ is encoded in $L(x)$ bits, it has probability $P(x) = 2^{-L(x)}$ [4, Theorem 5.2.1].

Mutual information can thus be understood as the expected number of bits we waste if we encode an element drawn from $P(X, Y)$ with the code corresponding to $P(X)P(Y)$, instead of the optimal choice, the code corresponding to $P(X, Y)$.

### Problem statement and approach

We can now define the central question of this paper more precisely. Let $o$ be an IRI. Let $n(o)$, $c(o)$ and $p(o)$ be its name (a Unicode string), its type-set and its predicate-set respectively. Let $O$ be a random element so that P(O) is a uniform distribution over all IRIs in the domain. Let $X = n(O)$, $Y^C = c(O)$ and $Y^P(O)$. As explained, we use $Y^C$ and $Y^P$ as *meaning proxies*, if the value of $X$ can be reliably used to predict the value of $Y^C$ or $Y^P$, then we take $X$ to contain information about its meaning. The treatment is the same for both proxies so we will use $Y$ as a symbol for a meaning proxy in general to report results for both.

We take the IRIs from an RDF dataset and consider them to be a sequence of randomly chosen IRIs from the dataset's domain with names $X_{1:n}$ and corresponding meanings $Y_{1:n}$. Our method can now be stated as follows:

> If we can show that there is *significant* mutual information between the name $X$ of an IRI and its meaning $Y$, then we have shown that the IRIs in this domain carry information about their meaning.

This implies a *best-effort* principle: if we can predict the value of $Y$ from the value of $X$ we have shown that $X$ carries meaning. However, if we did not manage this prediction, there may yet be smarter methods to do so and we have not proved anything. For instance, an IRI that seems to be a randomly generated string could always be an encrypted version of a meaningful one. Only by cracking the encryption could we prove the connection. Thus, we can prove conclusively that IRIs carry meaning, but not prove conclusively that they do not.

Of course, even randomly generated IRIs might, through chance, provide *some* information about their meaning. We use a *hypothesis test* to quantify the amount of evidence we have. We begin with the following null hypothesis:

> $H_0$: There is no mutual information between the IRIs $X_{1:n}$ and their meanings $Y_{1:n}$.

There are two issues when calculating the mutual information between names and meaning proxies for real-world data:

1. **Computational cost:** The straightforward method for testing independence between random variables is the use of a $\chi^2$-test. Unfortunately, this results in a computational complexity that is impractical for all but the smallest datasets.
2. **Data sparsity:** For many names there are too few occurrences in the data in order for a statistical model to be able to learn its meaning proxies. In these cases we must learn predict the meaning from attributes shared by different IRIs with the same meaning (clustering "similar" IRIs together).

To reduce computational costs, we develop a less straightforward *likelihood ratio test* that does have acceptable computational properties. To combat data-sparsity, we exploit the hierarchical nature of IRIs to group together IRIs that share initial segments. Where we do not have sufficient occurrences of the full IRI to make a useful prediction, we can look at other IRIs that share some prefix, and make a prediction based on that.

### Hypothesis testing

The approach we will use is a basic statistical hypothesis test: we formulate a null hypothesis (that the IRIs and their meanings have no mutual information) and then show that under the null hypothesis, the structure we observed in the data is very unlikely.

Let $X_{1:n}, Y_{1:n}$ denote the data of interest and let $P_0$ denote the true distribution of the data under the null hypothesis that $X$ and $Y$ are independent:

$$P_0(Y_{1:n}|X_{1:n}) = P_0(Y_{1:n}).$$

We will develop a likelihood ratio test to disprove the null hypothesis. The likelihood ratio $\mathit{\Lambda}$ is the ratio between the probability of the data if the null hypothesis is true, divided by the probability of the data under an *alternative model* $P_1$, which in this case attempts to exploit any dependencies between names and semantics of terms. We are free to design the alternative model as we like: the better our efforts, the more likely we are to disprove $P_0$, if it can be disproven. We can never be sure that we will capture all possible ways in which a meaning can be predicted from its proxy, but, as we will see in Section 4, a relatively straightforward approach suffices for most datasets.

**Likelihood ratio** The likelihood ratio $\mathit{\Lambda}$ is a test statistic contrasting the probability of the data under $P_0$ to the probability under an alternative model $P_1$:

$$\mathit{\Lambda} = \frac{P_0(Y_{1:n}|X_{1:n})}{P_1(Y_{1:n}|X_{1:n})} = \frac{P_0(Y_{1:n})}{P_1(Y_{1:n}|X_{1:n})}$$

If the data is sampled from $P_0$ (as the null hypothesis states) it is extremely improbable that this alternative model will give much higher probability to the data than $P_0$. Specifically:

$$P_0(\mathit{\Lambda} \leq \lambda) \leq \lambda \tag{1}$$

This inequality gives us a *conservative* hypothesis test: it may underestimate the statistical significance, but it will never *over*estimate it. For instance, if we observe data such that $\mathit{\Lambda} \geq 0.01$, the probability of this event under the null hypothesis is less than 0.01 and we can reject $H_0$ with significance level 0.01. The true significance level may be even lower, but to show that, a more expensive method may be required. To provide an intuition for what (1) means, we can take an information theoretic perspective. We rewrite:

$$P_0(-\log \mathit{\Lambda} \geq k) \leq 2^{-k} \qquad \text{with } k = -\log \lambda$$
$$-\log \mathit{\Lambda} = (-\log P_0(Y_{1:n} \mid X_{1:n})) - (-\log P_1(Y_{1:n} \mid X_{1:n}))$$

That is, if we observe a likelihood ratio of $\mathit{\Lambda}$, we know that the code corresponding to $P_1$ is $-\log \mathit{\Lambda}$ bits more efficient than $P_0$. Under $P_0$, the probability of this event is less than $2^{-k}$ (i.e. less than one in a billion for as few as 30 bits). Both codes are provided with $X_{1:n}$, but the first ignores this information while the second attempts to exploit it to encode $Y_{1:n}$ more efficiently. Finally, note that $H_0$ does not actually specify $P_0$, only that it is independent of $X_{1:n}$, so that we cannot actually compute $\Lambda$. We solve this by using

$$\hat{P}(Y = y) = \frac{|\{i \mid Y_i = y\}|}{n}$$

in place of $P_0$. $\hat{P}$ is guaranteed to upper-bound any $P_0$ (note that it "cheats" by using information from the dataset).[3]This means that by replacing the unknown $P_0$ with $\hat{P}$ we increase $\Lambda$, making the hypothesis test *more* conservative.

## 3 The Alternative Model

As described in the previous section, we must design an alternative model that gives higher probability to datasets where there is mutual information between IRIs and their meanings.[4] *Any* alternative model yields a valid test, but the better our design, the more likely it is we will be to be able reject the null-hypothesis, and the more strongly we will be able to reject it.

As discussed in the previous section, for many IRIs, we may only have one occurrence. From a single occurrence of an IRI we cannot make any meaningful predictions about its predicate-set, or its type-set. To make meaningful predictions, we cluster IRIs together. We exploit the hierarchical nature of IRIs by storing them together in a *prefix-tree* (also known as a *trie*). This is a tree with labeled edges where the root node represents the empty string and each leaf node represents exactly one IRI. The tree branches at every internal node into subtrees that represent (at least) two distinct IRIs that have a common prefix. The edge labels are chosen so that their concatenation along a path starting at the root node and ending in some node $n$ always results in the common prefix of the IRIs that are reachable from $n$. In other words: leaf nodes represent full IRIs and non-leaf nodes represent IRI prefixes. Since one IRI may be a strict prefix of another IRI, some non-leaf nodes may represent full IRIs as well.

For each IRI in the prefix tree, we choose a node to represent it: instead of using the full IRI, we represent the IRI by the prefix corresponding to the node, and use the set of all IRIs sharing that prefix to predict the meaning. Thus, we are faced with a trade-off: if we choose a node too far down, we will have too few examples to make a good prediction. If we choose a node too far up, the prefix will not contain any information about the meaning of the IRI we are currently dealing with.

Once the tree has been constructed we will make the choice once for all IRIs by constructing a *boundary*. A boundary $B$ is a set of tree nodes such that every path from the root node to a leaf node contains exactly one node in $B$. Once the

[3] A detailed proof for this, and for (1) is shared as an external resource at `http://wouterbeek.github.io/iswc2016_appendix.pdf`

[4] Or, equivalently, we must design a code which exploits the information that IRIs carry about their meaning to store the dataset efficiently.

boundary has been selected we can use it to map each IRI $X$ to a node $n_X$ in $B$. Multiple IRIs can be mapped onto the same boundary node. Let $X^B$ denote the node in the prefix tree for IRI $X$ and boundary $B$. We use $\mathcal{B}$ to denote the set of all boundaries for a given IRI tree.

For now, we will take the boundary as a given, a parameter of the model. Once we have described our model $P_1(Y_{1:n} \mid X_{1:n}, B)$ with $B$ as a parameter, we will describe how to deal with this choice.

We can now describe our model $P_1$. The most natural way to describe it, is as a sampling process. Note that we do not actually implement this process, it is simply a construction. We only compute the probability $P_1(Y_{1:n} \mid X_{1:n}, B)$ that a given set of meanings emerges from this process. Since we will use an IRI's boundary node boundary in place of the full IRI, we can rewrite

$$P_1(Y_{1:n} \mid X_{1:n}, B) = P_1(Y_{1:n} \mid X^B_{1:n}).$$

When viewed as a sampling process, the task of $P_1$ is to label a given sequence of IRIs with randomly chosen meanings. Note that when we view $P_0$ this way, it will label the IRIs independently of any information about the IRI, since $P_0(Y_{1:n} \mid X_{1:n}) = P_0(Y_{1:n})$. For $P_1$ to assign datasets with meaningful IRIs a higher probability than $P_0$, $P_1$ must assign the same meaning to the same boundary node more often than it would by chance.

We will use a Pitman-Yor process [16] as the basic structure of $P_1$.

We assign meanings to the nodes $X^B_i$ in order. At each node, we decide whether to sample its meaning from the global set of possible meanings $\mathcal{Y}$ or from the meanings that we have previously assigned to this node.

Let $\mathcal{Y}_i$ be the set of meanings that have previously been assigned to node $X^B_i$: $\mathcal{Y}_i = \{y_j \,|\, j \leq i \wedge X^B_j = X^B_{i+1}\}$.

With probability $\frac{(|\mathcal{Y}_i|+1)/2}{i+\frac{1}{2}}$, we choose a meaning for $X^B_i$ that has not been assigned to it before (i.e. $y \in \mathcal{Y} - \mathcal{Y}_i$). We then choose meaning $y$ with probability $\frac{|\{j \leq i : Y_j = y\}| + \frac{1}{2}}{i + |\mathcal{Y}|\frac{1}{2}}$[5]. Note that both probabilities have a self-reinforcing effect: every time we choose to sample a new meaning, we are more likely to do so in the future, and every time this results in a particular meaning $y$, we are more likely to choose $y$ in the future.

If we do not choose to sample a new meaning, we draw $y$ from the set of meanings previously assigned to $X^B_i$. Specifically:

$$P(Y_i = y \mid X^B_i) = \frac{|\{j \leq i \mid X^B_j = X^B_{i+1}, Y_j = y\}| - \frac{1}{2}}{i + \frac{1}{2}}.$$

Note that, again, the meanings that have been assigned often in the past are assigned more often in the future. These “the rich-get richer”-effects mean that the Pitman-Yor process tends to produce power-law distributions.

[5] The Pitman-Yor process itself does not specify which new meaning we should choose, only that a new meaning should be chosen. This distribution on meanings in $\mathcal{Y}$ is inspired by the Dirichlet-Multinomial model.

Note that this sampling process makes no attempt to map the "correct" meanings to IRIs: it simply assigns random ones. It is unlikely to produce a dataset that actually looks natural us. Nevertheless, a natural dataset with mutual information between IRIs and meanings still has a much higher probability under $P_1$ than under $P_0$, which is all we need to reject the null hypothesis.

While it may seem from this construction that the order in which we choose meanings has a strong influence on the probability of the sequence, it can in fact be shown that every permutation of any particular sequence of meanings has the same probability (the model is *exchangeable*). This is a desirable property, since the order in which IRIs occur in a dataset is usually not meaningful.

To compute the probability of $Y_{1:n}$ for a given set of nodes $X_{1:n}$ we use

$$\begin{aligned}
&P_1(Y_{1:n} \mid X^B_{1:n}) = \prod_{i=0}^{n-1} P_1(Y_{i+1} \mid Y_{1:i}, X^B_{1:n}) \qquad \text{with}\\
&P_1(Y_{i+1} = y \mid Y_{1:i}, X^B_{1:n})\\
&= \begin{cases} \dfrac{(|\mathcal{Y}_i| + 1)\frac{1}{2}}{i + \frac{1}{2}} \cdot \dfrac{|\{j \le i : Y_j = y\}| + \frac{1}{2}}{i + |\mathcal{Y}|\frac{1}{2}} & \text{if } y \notin \mathcal{Y}_i,\\[2ex] \dfrac{|\{1 \le j \le i \mid X^B_j = X^B_{i+1}, Y_j = y\}| - \frac{1}{2}}{i + \frac{1}{2}} & \text{otherwise.} \end{cases}
\end{aligned}$$

**Choosing the IRI boundary** We did not yet specify which boundary results in clusters that are of the right size, i.e., which boundary choice of boundary gives us the highest probability for the data under $P_1$, and thus the best chance of rejecting the null hypothesis.

Unfortunately, which boundary $B$ is best for predicting the meanings $Y$ cannot be determined a priori. To get from $P_1(Y \mid X, B)$ to $P_1(Y \mid X)$, i.e. to get rid of the boundary parameter, we take a Bayesian approach: we define a prior distribution $W(B)$ on all boundaries, and compute the marginal distribution on $Y_{1:n}$:

$$P_1(Y_{1:n} \mid X_{1:n}) = \sum_{B \in \mathcal{B}} W(B) P_1(Y_{1:n} \mid X_{1:n}, B) \qquad (2)$$

This is our complete alternative model.

To define $W(B)$, remember that a boundary consists of IRI prefixes that are nodes in an IRI tree (see above). Let $\text{lcp}(x_1, x_2)$ denote the longest common prefix of the IRIs denoted by tree nodes $x_1$ and $x_2$. We then define the following distribution on boundaries:

$$W(B) := 2^{-|\{\text{lcp}(x_1,x_2) \mid x_1, x_2 \in B\}|}$$

Here, the set of prefixes in the exponent corresponds to the nodes that are in between the root and some boundary node, including the boundary nodes themselves. Therefore, the size of this set is equal to the number of nodes in the boundary plus all internal nodes that are closer to the root. Each such node divides the probability in half, which means that $W$ can be interpreted as the following generative process: starting from the root, a coin is flipped to decide for each node whether it is included in the boundary (in which case its descendants are not) or not included in the boundary (in which case we need to recursively flip coins to decide whether its children are).

The number of possible boundaries $\mathcal{B}$ is often very large, in which case computing 2 takes a long time. We therefore use a heuristic (Algorithm 1) to lower-bound (2), by using only those terms that contribute the most to the total. Starting with the single-node boundary containing only the root node, we recursively expand the boundary. We compute $P_1$ for all possible expansion of each boundary we encountered, but we recurse only for the one which provides the largest contribution.

Note that this only weakens the alternative model: the probability under the heuristic version of $P_1$ is always lower than it would be under the full version, so that the resulting hypothesis tests results in a higher $p$-value. In short, this approximation may result in fewer rejections of the null hypothesis, but when we do reject it, we know that we would also have rejected it if we had computed $P_1$ over all possible boundaries. If we cannot reject, there may be other alternative models that would lead to a rejection, but that is true for the complete $P_1$ in (2) as well. Algorithm 1 calculates the probability of the data under the alternative model, requiring only a single pass over the data for every boundary that is tested.

**Algorithm 1** Heuristic calculation for the IRI boundary.

```
procedure MarginalProbability($X_{1:n}$, $Y_{1:n}$, IRI tree with root $r$)
    $B \leftarrow \{r\}$                                   ▷ The boundary in the sum in (2)
    $Q \leftarrow \{r\}$                                   ▷ Queue of boundary states to be expanded
    $best\_term \leftarrow W(B)P_1(Y_{1:n} \mid X_{1:n}, B)$   ▷ Largest term found
    $acc \leftarrow best\_term$                            ▷ Accumulated probability
    while $Q \neq \emptyset$ do
        $n \leftarrow shift(Q)$
        $B' \leftarrow B \setminus \{n\} \cup children(n)$
        $term \leftarrow W(B)P_1(Y_{1:n} \mid X_{1:n}, B')$
        $acc \leftarrow acc + term$
        if $term \geq best\_term$ then
            $(B, best\_term) \leftarrow (B', term)$
            $add(Q, children(n))$
    return $acc$                                          ▷ Approx. $P_1(Y_{1:n} \mid X_{1:n})$ from below
```

## 4 Evaluation

In the previous section we have developed a likelihood ratio test which allows us to verify the null hypothesis that names are statistically independent from the two meaning proxies. Moreover, the alternative model $P_1$, provides a way of quantifying how much meaning is (at least) shared between IRI names $X$ and meaning proxies $Y$.

Since we calculate $P_1$ on a per-dataset basis our evaluation needs to scale in terms of the number of datasets. This is particularly important since we are dealing with Semantic Web data, whose open data model results in a very heterogeneous collection of real-world datasets. For example, results that are obtained over a relatively simple taxonomy may not translate to a more complicated ontology. Moreover, since we want to show that our approach and its corresponding implementation scale, the datasets have to be of varying size and some of them have to be relatively big.

For this experiment we use the LOD Laundromat data collection [1], a snapshot of the LOD Cloud that is collected by the LOD Laundromat scraping, cleaning and republishing framework. LOD datasets are scraped from open data portals like Datahub[6] and are automatically cleaned and converted to a standards-compliant format. The data cleaning process includes removing 'stains' from the data such as syntax errors, duplicate statements, blank nodes and more.

We processed $544,504$ datasets from the LOD Laundromat data collection, ranging from 1 to $129,870$ triples. For all datasets we calculate the $\Lambda$-value for the two meaning proxies $Y^C$ and $Y^P$, noting that if $\Lambda < \alpha$, then $p < \alpha$ also, and we can reject the null-hypothesis with significance level at least $\alpha$. We choose $\alpha = 0.01$ for all experiments.

Figure 1 shows the frequency with which the null hypothesis was rejected for datasets in different size ranges.

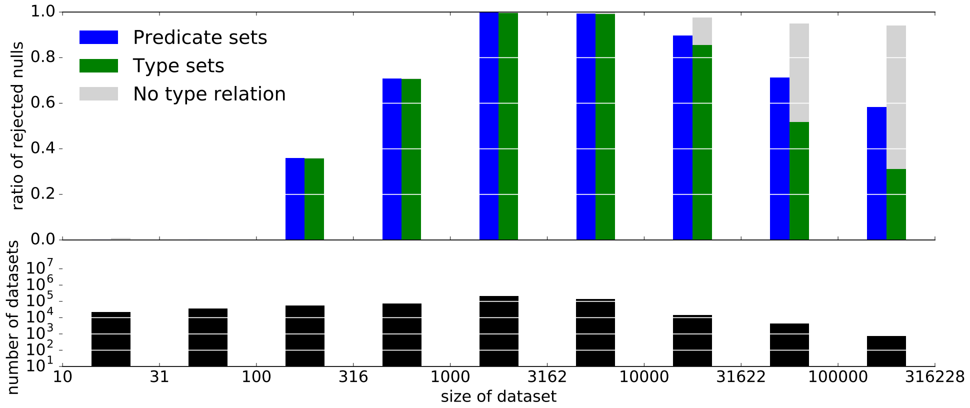

**Fig. 1.** The fraction of datasets for which we obtain a significant result at significance level $\alpha = 0.01$. Note that we group the datasets in logarithmic bins (i.e., the bin edges $\{e_i\}$ are chosen so that the values $\{\log e_i\}$ are linearly spaced. As explained in Section 2, all datasets have predicate-sets but not all datasets have type-sets. The fraction of datasets with no type-set is marked in gray.

[6] See `http://datahub.io`

The figure shows that for datasets with at least hundreds of statements our method is usually able to reliably refute the null hypothesis at a very strong significance level of $\alpha = 0.01$. $6,351$ datasets had no instance/class-assertions (i.e., `rdf:type`-statements) whatsoever (shown in gray in Figure 1). For these datasets it was therefore not possible to obtain results for $Y^C$.

Note that we may *not* conclude that no datasets with less than 100 statements contain meaningful IRIs. We had too little data to show meaning in the IRIs with our method, but other, more expensive methods may yet be successful.

In Figure 2 we explore the correlation between the results for type-sets $Y^C$ and property-sets $Y^P$. As it turns out, in cases where we do find evidence for social meaning the evidence is often overwhelming, with a $p\Lambda$-value exponentially small in terms of the number of statements. It is therefore instructive to consider not the $\Lambda$-value itself but its binary logarithm. A further reason for studying $\log \Lambda$ is that $-\log \Lambda$ can be seen not only as a measure of evidence against the null hypothesis that $Y$ and $X$ are independent, but also as a conservative estimate of the mutual information $I(X{:}Y)$: predicting the meanings from the IRIs instead of assuming independence allows us to encode the data more efficiently by at least $-\log \Lambda$ bits.

In Figure 2, the two axes correspond to the two meaning proxies, with $Y^P$ on the horizontal and $Y^C$ on the vertical axis. To show the astronomical level of significance achieved for some datasets, we have indicated several significance thresholds with dotted lines in the figure. The figure shows results for $544,504$ datasets[7] and as Figure 2 shows, the overwhelming majority of these indicate very strong support for the encoding of meaning in IRIs, measured both via mutual information content with type-sets and with property-sets. Recall that $-\log \Lambda$ is a lower bound for the amount of information the IRIs contain about their meaning. For datasets that appear to the top-left of the diagonal property-sets $Y^P$ provide more evidence than type-sets $Y^C$. For points to the bottom-right of the diagonal, type-sets $Y^C$ provide more evidence than property-sets $Y^P$.

Only very few datasets appear in the upper-right quadrant. Manual inspection has shown that these are indeed datasets that use 'meaningless' IRIs. There are some datasets where the $\log \Lambda$ for property-sets is substantially higher than zero; this probably occurs when there are very many property-sets so that the alternative model has many parameters to fit, whereas the null model is a maximum likelihood estimate so it does not have to pay for parameter information.

Datasets that cluster around the diagonal are ones that yield comparable results for $Y^C$ and $Y^P$. There is also a substantial grouping around the horizontal axis: these are the datasets with poor `rdf:type` specifications. There is some additional clustering visible, reflecting that there is structure not only within individual Semantic Web datasets but also between them. This may be due to a

[7] Datasets with fewer than $1,000$ statements are not included in order to get a clear picture of what happens in case we have sufficient data to refute the null, as indicated by our observations from Figure 1. A zoomed out version of Figure 2, scaling to $\log(p)$ values of $-300,000$ is available at `https://goo.gl/r3uxpA`, but is not included in this paper because its scale is no longer suitable for print.

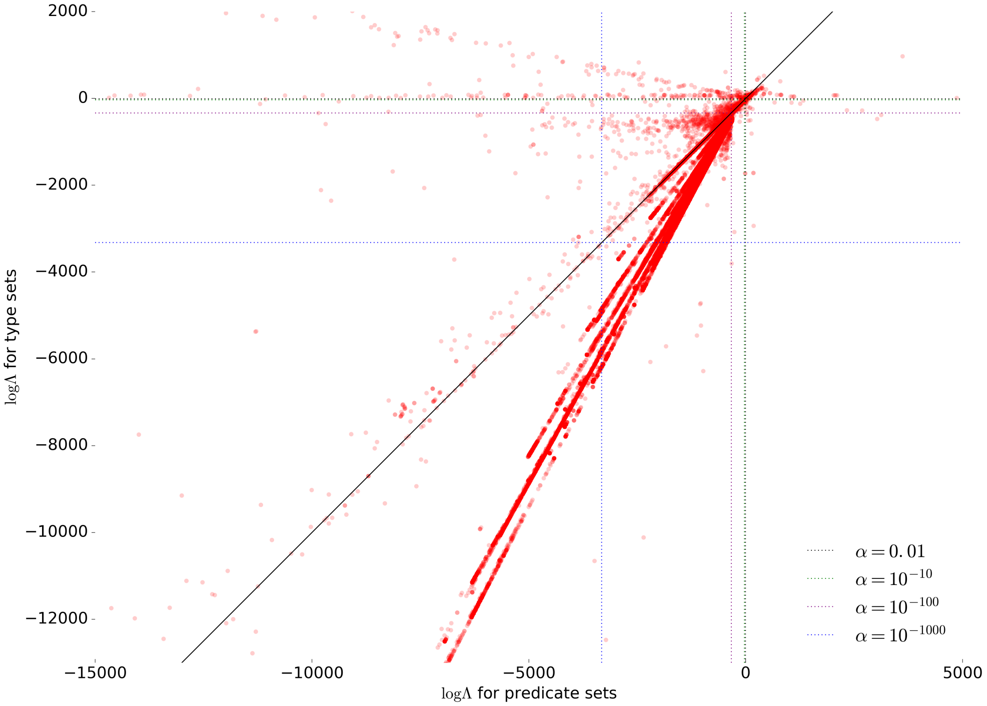

**Fig. 2.** This figure shows $\log\Lambda$ for both meaning proxies, for each dataset. Datasets that appear below a horizontal line provide sufficient evidence (at that $\alpha$) to refute the claim that Semantic Web names do not encode type-sets $Y^C$. Datasets that appear to the left of a vertical line provide sufficient evidence (at that $\alpha$) to refute the claim that Semantic Web names do not encode property-sets $Y^P$. Datasets containing no instance/class- or `rdf:type`-relations are not included.

single data creator releasing multiple datasets that share a common structure. These structures may be investigated further in future research.

The results reported on until now have been about the *amount of evidence against the null hypothesis*. In our final figure we report about the *amount of information that is encoded in Semantic Web names*. For this we ask ourselves the information theoretic question: how many bits of the schema information in $Y$ can be compressed by taking into account the name $X$? Again we make a conservative estimate: the average number of bits required to describe $Y$ is underestimated by the empirical entropy, whereas the average number of bits we need to encode $Y$ with our alternative model, given by $-\log(P_1(Y_{1:n}|X_{1:n})/n$, is an overestimate (because $P_1$ is an ad-hoc model rather than the true distribution). Again, we only consider datasets with more than $1,000$ statements.

The results in Figure 3 show that for many datasets more than half of the information in $Y$, and sometimes almost all of it, *can* in fact be predicted by

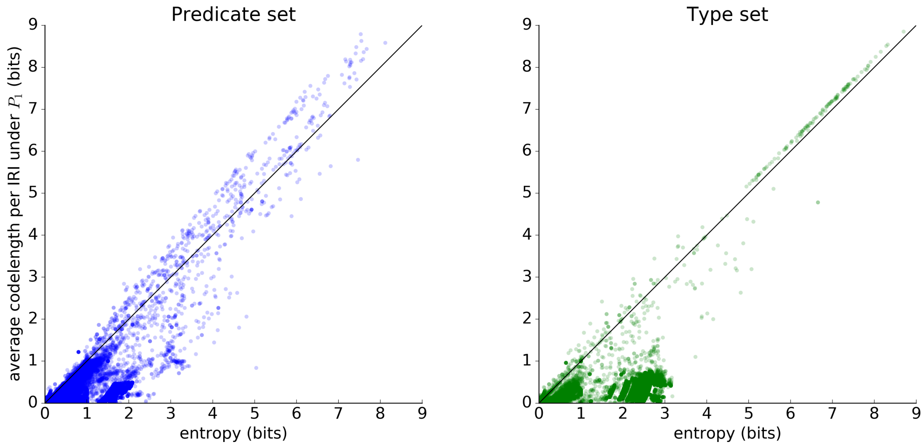

**Fig. 3.** Measuring the amount of information that is encoded in Semantic Web names. The horizontal axis shows the entropy of the empirical distribution of $Y$ for a given dataset, a lower-bound for the information contained in the meaning of the average IRI. The vertical axis shows the number of bits used to encode the average meaning by the code corresponding to $P_1$. This is an upper bound, since $P_1$ may not be the optimal model. Datasets containing no type relations are not included in the right-hand figure.

looking at the IRI. On the other hand, for datasets of high entropy the alternative model $P_1$ tends not to compress a lot. Pending further investigation, it is unclear whether this later result is due to inefficiency in the alternative model or because the IRIs in those datasets are just less informative.

## 5 Related work

**Statistical observations** Little is known about the information theoretic properties of real-world RDF data. Structural properties of RDF data have been observed to follow a power-law distribution. These structural properties include the size of documents [6] and frequency of term and schema occurrence [6,15,19]. Such observations have been used as heuristics in the implementation of triple stores and data compressors.

The two meaning proxies we have used were defined by [10] who report the empirical entropy and the mutual information of both $Y^C$ and $Y^P$ for various datasets. However, we note that the distribution underlying $Y^C$ and $Y^P$, as well as the joint distribution on pairs $\langle Y^P, Y^C \rangle$, is unknown and has to be *estimated* from the observed frequencies of occurrence in the data. This induces a bias in the reported mutual information. Specifically, the mutual information may be substantial even though the variables $Y^C$ and $Y^P$ are in fact *independent*. Our approach in Section 2 avoids this bias.

**Social Meaning** The concept of social meaning on the Semantic Web was actively discussed on W3C mailing lists during the formation of the original

RDF standard in 2003-2004. social meaning is similar to what has been termed the "human-meaningful" approach to semantics by [9]. While social meaning has been extensively studied from a philosophical point of view by [11], to the best of our knowledge there are no earlier investigations into its *empirical* properties.

Perhaps most closely related is again the work in [10]. They study the same two meaning proxies (which we have adopted from their work), and report on empirical entropy and mutual information of *between* two quantities. That is essentially different from our work, where we study the entropy and mutual information content not between these two quantities, but between each of them and the IRIs whose formal meaning they capture. Thus, [10] tells us whether type-sets are predictive of predicate-sets, whereas our work tells us whether IRIs are predictive of their type- and predicate-sets.

**Naming RDF resources** Human readability and memorization are explicit design requirements for URIs and IRIs. [3,8,20] At the same time, best practices have been described that advise against putting "too much meaning" into IRIs [20]. This mainly concerns aspects that can easily change over time and that would, therefore, conflict with the permanence property of so-called 'Cool URIs' [2]. Examples of violations of best practices include indicators of the status of the IRI-denoted resource ('old', 'draft'), its access level restrictions ('private', 'public') and implementation details of the underlying system ('`/html/`', '`.cgi`').

Several guidelines exist for minting IRIs with the specific purpose of naming RDF resources. [17] promotes the use of the aforementioned Cool URIs due to the improved referential permanence they bring and also prefers IRIs to be mnemonic and short. In cases in which vocabularies have evolved over time the date at which an IRI has been issued or minted has sometimes been included as part of that IRI for versioning purposes.

## 6 Conclusion & future work

In this paper we have shown that Semantic Web data contains social meaning. Specifically, we have quantitatively shown that the social meaning encoded in IRI names significantly coincides with the formal meaning of IRI-denoted resources.

We believe that such quantitative knowledge about encoded social meaning in Semantic Web names is important for the design of future tools and methods. For instance, ontology alignment tools already use string similarity metrics between class and property names in order to establish concept alignments [18]. The Ontology Alignment Evaluation Initiative (OAEI) contains specific cases in which concept names are (consistently) altered [7]. The analytical techniques provided in this paper can be used to predict *a priori* whether or not such techniques will be effective on a given dataset. Specifically, datasets in the upper right quadrant of Figure 2 are unlikely to yield to those techniques.

Similarly, we claim that social meaning should be taken into account when designing reasoners. [14] already showed how the names of IRIs could be used effectively as a measure for semantic distance in order to find coherent subsets of information. This is a clear case where social meaning is used to support reasoning with formal meaning. Our analysis in the current paper has shown that such a combination of social meaning and formal meaning is a fruitful avenue to pursue.

## References

1. Beek, W., Rietveld, L., Bazoobandi, H., Wielemaker, J., Schlobach, S.: LOD laundromat: A uniform way of publishing other people's dirty data. In: The Semantic Web–ISWC 2014, pp. 213–228. Springer (2014)
2. Berners-Lee, T.: Cool URIs don't change (1998), `http://www.w3.org/Provider/Style/URI.html.en`
3. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifier: Generic syntax (January 2005), `http://www.rfc-editor.org/info/rfc3986`
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience (2006)
5. Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 concepts and abstract syntax (2014)
6. Ding, L., Finin, T.: Characterizing the Semantic Web on the web. In: The Semantic Web–ISWC 2006, pp. 242–257. Springer (2006)
7. Dragisic, Z., Eckert, K., Euzenat, J., Faria, D., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., et al.: Results of the Ontology Alignment Evaluation Initiative 2014. In: Proceedings of the 9th International Conference on Ontology Matching. vol. 1317, pp. 61–104 (2014)
8. Duerst, M., Suignard, M.: Internationalized Resource Identifiers (January 2005), `http://www.rfc-editor.org/info/rfc3987`
9. Farrugia, J.: Model-theoretic Semantics for the Web. In: Proc. of the 12th Int. Conf. on WWW. pp. 29–38. ACM (2003)
10. Gottron, T., Knauf, M., Scheglmann, S., Scherp, A.: A systematic investigation of explicit and implicit schema information on the Linked Open Data Cloud. In: Proceedings of ESWC. pp. 228–242 (2013)
11. Halpern, H.: Social Semantics: The Search for Meaning on the Web. Springer (2013)
12. Halpin, H., Thompson, H.: Social meaning on the web: From wittgenstein to search engines. IEEE Intelligent Systems 24(6), 27–31 (2009)
13. Hayes, P.J., Patel-Schneider, P.F.: RDF 1.1 semantics (2014)
14. Huang, Z., van Harmelen, F.: Using semantic distances for reasoning with inconsistent ontologies. In: ISWC Proc. LNCS, vol. 5318, pp. 178–194. Springer (2008)
15. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: A document-oriented lookup index for Open Linked Data. International Journal of Metadata, Semantics and Ontologies 3(1), 37–52 (2008)
16. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. The Annals of Probability 25(2), 855–900 (April 1997)
17. Sauermann, L., Cyganiak, R.: Cool URIs for the Semantic Web (2006)
18. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: International Semantic Web Conference. pp. 624–637. Springer (2005)
19. Theoharis, Y., Tzitzikas, Y., Kotzinos, D., Christophides, V.: On Graph Features of Semantic Web Schemas. IEEE Transactions on Knowledge and Data Engineering 20(5), 692–702 (2008)
20. Théraux, O.: Common HTTP Implementation Problems (January 2003), `http://www.w3.org/TR/chips/`