Calculating Error Bars on Inferences from Web Data

Calculating Error Bars on Inferences from Web Data

Kwabena Nuamah, Alan Bundy
School of Informatics, University of Edinburgh, United Kingdom
Email: k.nuamah@ed.ac.uk, a.bundy@ed.ac.uk

***Abstract*—In this work, we explore uncertainty in automated question answering over real-valued data from knowledge bases on the Internet. We argue that the coefficient of variation (*cov*) is an intuitive and general form in which to express this uncertainty, with the added advantage that it can be calculated exactly and efficiently. The large amounts of data on the Internet presents a good opportunity to answer queries that go beyond simply looking up facts and returning them. However, such data is often vague and noisy. For discrete results, e.g., stating that a particular city is the capital of a particular country, probabilities are a natural way to assign uncertainty to answers. For continuous variables or quantities that are typically treated as continuous (such as populations of countries), probabilities are uninformative, being infinitesimal For instance, the probability that the population of India is exactly equal to last census count is effectively zero. Our aim is to capture uncertainty in these estimates in an intuitive, uniform, and computationally efficient way. We present initial efforts at automating the inference process over real-valued web data while accounting for some of the typical sources of uncertainty: noisy data and errors from inference operations. Having considered several problem domains and query types, we find that approximating all continuous random variables with Gaussian distributions, and communicating uncertainties to users as coefficients of variation. Our experiments show that the estimates of uncertainty derived by our method are well-calibrated and correlate with the actual deviations from the true answer. An immediate benefit of our approach is that our inference framework can attach credible intervals[1] to real-valued answers that it infers. This conveys to a user the plausible magnitudes of the error in the answer, a meaningful measure of uncertainty compared to ranking scores provided in other question answering systems.**

***Index Terms*—Query Answering; Credible Intervals; Uncertainty; Bayesian Inference; Coefficient of Variation**

I. Introduction

As the world wide web and web-based data sources grow, so does the promise that they will allow us to ask virtually any question and receive a reliable and appropriate answer. However, as web-based knowledge bases (KBs) grow, they become increasingly difficult for humans to assess, curate, and correct. They vary in quality, and most contain at least some imprecise or erroneous records. It is thus essential that automated question answering (QA) systems accommodate uncertainty in the accuracy of 'facts' recovered from KBs, to produce more accurate answers, and to communicate uncertainty to end users, allowing them to distinguish between high-quality, high-precision answers and what are essentially guesses.

Most QA systems that rely on web KBs fail to address the uncertainty that comes with such noisy sources or missing values that have to be inferred. Many more (as discussed in section VII) fail to acknowledge the uncertainty in the inference methods used. Such systems avoid the uncertainty problem and resort instead to ranking candidate answers without providing information about the quality of that ranking or relative confidences. This creates difficulty in interpreting the ranking, and sidesteps the challenge of informing the user about any errors in the answer returned.

Our hypothesis is that *credible intervals are an appropriate way to express uncertainty when inferring real-valued answers from a QA system and can be accurately estimated when the inference process incorporates relevant probabilistic measures as a core component of its internal representation*. Visually, credible intervals can be represented as *error bars*, making it easily interpretable.

To achieve this, we build on the Rich Inference Framework (RIF) [1] which provides a flexible representation to dynamically curate data from different sources with different representations into the same inference tree to find answers to queries. RIF is built on the idea of solving complex problems by composing solutions to sub-problems from several smaller operations (programs or algorithms) recursively.

In this work, we estimate the credible intervals on real-valued answers inferred in RIF, tracking and propagating variance estimates throughout the inference process. We evaluate our method using data from web KBs and show that it provides useful information about the accuracies of answers, as shown by a positive correlation between the size of the credible intervals that RIF assigns to its answer and the actual errors between inferred answers and the true values.

II. Overview

Question answering over continuous, real-valued facts from web KBs usually involves information that is inherently vague. For instance, consider the sentence:

$$Q : population(UK, 2011) = 63M$$

that states that the population of the UK in 2011 is 63 million. Not only is there a vagueness about who counts in

[1]We will use symmetric 68.27 percent credible intervals for the remainder of this paper, corresponding to 1 standard deviation from the mean in a standarized Gaussian, but note that this contains sufficient information to estimate arbitrary posterior probabilities under our assumption of normality.

the population (given, say, the constant movement of people, or those in the process of being born), but with our current technology it's impossible to be accurate to a single person. Therefore, the probability of the assertion is,

$$p(Q) \approx 0$$

So, a probabilistic logic [2] approach, where we assign a probability to that assertion is not a good technology in this domain. What makes more sense is the probability that the answer lies in some range:

$$population(UK, 2011) \in \{n | 63M - 0.32M < n < 64M + 0.32M\}$$

But for our purposes of providing a meaningful answer to the user, it makes even more sense to invert this and say, for a fixed probability, what range the answer lies in. This is usually understood as the credible interval on an answer.

Rather than calculating the probability of assertions in the KB or the inferences made (e.g., in probabilistic logic), we deal with uncertainty by considering the variances in the data observed when inferring answers. To formalise these notions, we need to commit to a statistical model. We assume that the data retrieved from the KB are normally distributed. We assume that the answer is the mean of this distribution and the variance accounts for the uncertainty.

The assumption of normality makes it straightforward to update uncertainty estimates incrementally, assigning a prior variance to a KB's data values and then updating the variance as data is retrieved. This is particularly useful in cases where a dataset is not fully observed and estimations are done online rather than in batch. Given that a knowledge base contains data of different kinds, it is inappropriate to use an unnormalized value as a prior variance attributed to the KB's data since the variance is scaled by the magnitude of the mean. Instead, we use a measure of relative variability known as the *coefficient of variation* (*cov*) calculated as the ratio of the standard deviation to the mean. The *cov* is often expressed as a percentage.

For instance, attaching a prior *cov* of 0.5% to real-valued data records in a KB means that if $63M$ is retrieved from the KB for the query $population(UK, 2011)$, then the size of the credible interval associated with this value is $\pm 0.32M$. This means that we can store our estimates of errors in the KB's data as a *cov*, and, for a property for which we do not yet have a prior variance, use the *cov* to calculate one once the specific quantity being measured is known. From there on, we can calculate our posterior variances using sequential estimation of the means and variances. We explain this further in section IV.

We propagate variance estimates through the derivation in our inference tree, from the child nodes to the parents in closed-form. Some inference operations, such as regression, also introduce uncertainties which are inherited during propagation of variances. We prefer to store the variances as *cov* during the inference process because (1) they are dimensionless and therefore uniformly represented throughout the inference tree; and (2) they provide a more intuitive basis to compare uncertainties across different scales. But, since we have the means in the nodes of the tree, we are able to convert the *cov*s back to variances for statistical calculations during propagation. The variance at the root of the inference tree is then used to inform the user about the uncertainty in the answer derived.

We explore and implement our ideas in the Rich Inference Framework (RIF) for query answering over real-valued data, so we first provide a brief overview of RIF in the next section.

III. Rich Inference Framework

A. Overview

The objective of RIF is to infer answers to queries when the required data that answers the query is not stored in the available knowledge bases. RIF uses a graph-based algorithm that recursively decomposes queries, eventually grounding out in either stored facts or previously cached answers. RIF focuses on the compositional application of aggregate functions (aggregates), including prediction, as well as the decomposition rules that determine the relevance of aggregates at different stages in inference process. These decomposition rules extend the RIF tree when lookups in knowledge bases fail to instantiate the unknowns (variables) in nodes.

Humans are reasonably good at inferring answers to questions even when the answer is not stored in our memory or in a knowledge base. In contrast, QA systems, although designed to use inference of varying kinds, tend to focus on the task of efficiently retrieving facts that are pre-stored in a knowledge base. Most perform minimal or no inference operations on real-valued data even when these existing facts answer the question.

Consider the question *"What will be the UK population in 2021?"*. A QA system will typically attempt (unsuccessfully) to find the pre-stored fact ⟨*UK,population,p,2021*⟩ where the quad represents ⟨*subject, predicate, object, time*⟩. Unless the KB stores forecasts in addition to the facts, it will not find this, and so will give up and return no answer. Humans, however, are able to answer this kind of question by using other readily available information. In the example above, we could look up the population values for past census years, and then extrapolate to determine the population in 2021 using regression. A similar technique can be employed to interpolate missing facts. For example when the year is in the past, but no data was observed in the KBs. RIF is motivated by these kinds of inference.

B. Representation

A frame is a set of attribute-value pairs that specified an entity and its properties as well as the operations to be performed on it during inference. Attribute names include, but are not limited to, *subject, property, object, time* and *cov* (for uncertainty) and can be dynamically added to during inference. Although some object-level information (e.g. time) about facts may exist as triples in the database, the representation of facts as triples limits the inclusion of information that is useful for

Lookup Query: What was the urban population of Africa in 2010?

```
{op:"value",ov:"?x",s:"Ghana",p:"urban population",o:"?x",t:2010}
```

Aggregate Query: What was the average gdp of countries in Europe in 2005?

```
{op:"avg",ov:"?x",s:"$y",p:"gdp",o:"?x",t:2005,
 "$y":[{p:"is_a",o:"country"},{p:"located_at",o:"Europe"}]
}
```

Nested Query: Who is the leader of the country with the highest male population in Africa in 1992?

```
{op:"value",ov:"?x",s:"$y",p:"leader",o:"?x",
  {op:"max",ov:"?y",s:"$z",p:"male_population",o:"?y",t:1992,
    "$z":[{p:"is_a",o:"country"},{p:"located_at",o:"Africa"}]}
}
```

Prediction Query: What will be the urban population of Africa in 2023?

```
{op:"value",ov:"?x",s:"Ghana",p:"urban population",o:"?x",t:2023}
```

Key op=operation, ov=operation variable, s=subject, p=property, o=object, t=time

Fig. 1. Examples of RIF queries.

reasoning, unless we resort to techniques such as reification[2] as used in RDF [3]. RIF's frame, therefore, augments the triple representation found in relational and graph databases.

Interoperability with web KBs having different representations, and RIF's dynamic curation of facts inform our choice of JSON (JavaScript Object Notation) [4] as an appropriate representation to serialize RIF frames. For instance, graph databases (including Linked-data [5] KBs) represent data as triples. Reified forms of these graph subsets for a given entity can be represented as key-value pairs. Traditional relational databases management systems (RDBMS) use tables (relations) whereas NoSQL databases use a document-based (key-value) representation.

RIF queries (examples in fig 1) are also expressed as frames. Variables are denoted by a $ or ? prefix. These represent the unknowns in a frame that have to be instantiated to successfully answer a query. Since RIF returns a specific answer, the ? prefix indicates the variable in a query that is returned as the answer. A variable can also be used in the operation variable (ov) attribute of a frame. This indicates the variable to which the frame's inference operation is applied.

C. Composable Inference Operations

RIF finds answers to queries by applying operations to data in a compositional and recursive manner. An inference operation (1) determines how a RIF frame should be decomposed, and (2) specifies how the child frames of the decomposed frame should be combined in order to instantiate variables in parent frames. The aggregate functions used for (2) are shown in table I. That is, operations transform values of frames attributes and propagate them to their children or parent frames, depending on the type of operation being performed. Figure 2 shows an illustration of the RIF tree that results from these operations.

[2]https://www.w3.org/TR/rdf11-mt/

Which country will have the largest population in Africa in 2021?

```
{op:"max",ov:"$y",s:"?x",p:"population",o:"$y",t:"2021",
 "$y":[{p:"is_a",o:"country"},{p:"located_at",o:"Africa"}]}
```

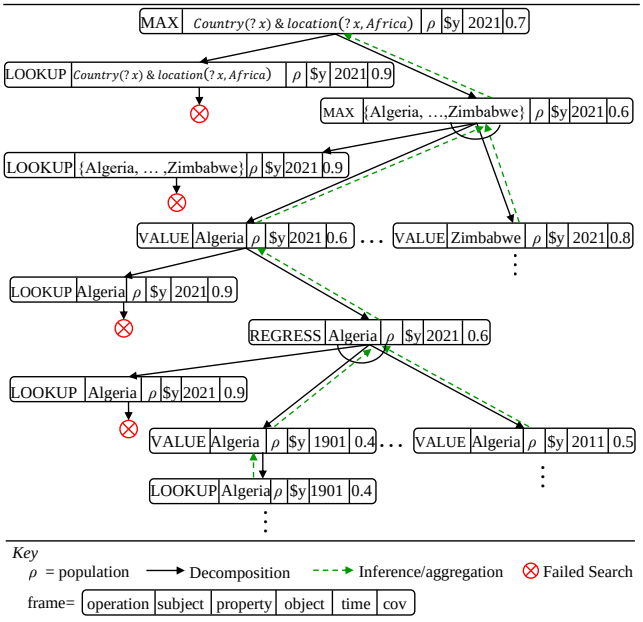

Fig. 2. An example showing a RIF tree.

Frame decomposition is the process of creating new child frames from a frame, *f*, by transforming attributes values and variables in *f*. The *Frame Normalisation* rule first converts all frames to RIF Normal Form prior to variable instantiation. That is, compound frames containing sub-queries (nested frames) are unpacked into simpler child frames without sub-queries. For normalized frames, a decomposition is triggered whenever a KB lookup fails to return any results. The objective of decomposing a frame is to increase the chances of grounding its variables when the KBs are searched. Three decomposition rules determine how we extend and explore the RIF tree. These are: (1) lookup decomposition, (2) temporal decomposition and (3) geospatial decomposition.

1) Lookup Decomposition: Lookup decomposition is the base case rule for finding matching facts from knowledge bases. One of the problems that arises in open-domain query answering using web data is the expression of the same concept in multiple ways. This decomposition finds synonyms of terms in the frame to increase the chances of finding a match in the KB. We use the Jaro-Winkler edit distance [6] as the criteria for selecting the best matches between terms in queries to those in the KB as it performs slightly better than other similar edit distance measure. We then formulate the appropriate queries to retrieve data from the KB.

2) Temporal Decomposition: Temporal decomposition is based on the intuition that if the query requests data for a specified date and that data point is not available in the KBs, then we can take advantage of the data observed for different dates and draw inferences from it for the date originally required in the query using regression. In this project, we limit

TABLE I
RIF AGGREGATES

Operation	Description
VALUE	Default aggregate. Identity function on frame's value
SUM	Add values of frames of numeric type
AVG	Mean value of child frames
MEDIAN	Median value of child frames
MAX	Maximum value of child frames
MIN	Minimum value of child frames
COMP	Obtain a set by list comprehension
GT	'Greater Than' function to compare two frames
LT	'Less Than' function to compare two frames
EQ	Check if the value of two frames are equal
REGRESS	Regression function from child frames

temporal decomposition to the 'year' level of granularity for data/time values.

3) Geospatial Decomposition: Our use of geospatial decomposition in RIF is based on the ideas of merology and merotopology [7] that create a spatial logic for regions and connections. This type of decomposition is applied to attributes in frames that represent geo-spatial concepts. We use a part-hood rule based on the idea that an entity can be partitioned into its sub-parts such that a problem can decomposed into smaller problems for the sub-parts which could be easier to solve and then aggregated to solve for the whole entity. For example, if an entity such as a continent can be partitioned into the countries that it is composed of, then we can solve a problem for its countries first, and then aggregate to get an answer for the continent.

RIF can be extended to other domains by adding the appropriate inference operations. For instance, RIF was extended to the energy domain in (Markov, 2017) [8] by creating inference operations that determined how to answer queries such as "*What proportion of the UK will need to be covered by solar panels in order to supply enough energy to meet the entirety of the UK's energy demand in 1999?"*.

In the sections that follow, we explore in more detail the estimation of uncertainty in answers inferred by RIF.

IV. UNCERTAINTY IN RIF

A. Overview

Uncertainty is unavoidable when answering queries over heterogeneous data sources. So providing an uncertain answer is acceptable in such cases as long as the user is informed about the degree of uncertainty and the uncertainty measure is a tight and accurate estimate. We define uncertainty as the possible deviation of a fact or an inferred answer from its true value. As a result, we consider credible intervals an appropriate representation for uncertainties attached to an answer. We also limit our discussion to the uncertainty of real-valued data records and answers. Subsequently, we mean *real-valued data* when we refer to data in KBs, and *real-valued answers* when we refer to answers from RIF. We focus on two main forms of uncertainty in RIF:

- *KB uncertainty*: Imprecision of KB data values due to errors.
- *Inference operation uncertainty*: Due to approximations resulting from the execution of inference operations. This is currently limited to regression.

Our aim in calculating uncertainty is to estimate the error in an answer value computed by RIF by assessing the accuracy of the data records in a KB. We assume that each KB has noise that is reflected as a deviation from the true value of the facts they contain by assigning a prior variance (expressed as *cov*) on the data in the KB. As we retrieve more data from a KB, we update the initial variance by computing the posterior variance given the new data seen. Thus, the posterior variance attached to data values of a KB is conditional on data observed from all other KBs that RIF finds data from.

The KB's data variances, when incorporated into RIF, together with the errors that are introduced from RIF's aggregate functions, reflect the error in the computed answer. This approach means focusing not only on the precision of individual data records in a KB, but also to have a notion of the precision of data values in the KB as a whole such that we can infer the error in a retrieved value even if we have never previously seen data for that property of an entity.

This general application of KB error to its constituent data leads us to use the Coefficient of Variation (*cov*) (section IV-B) as a normalized form of the error since a KB contains facts of different types, with magnitudes on different scales. We track parameters of the distribution that allows us to make useful statements about uncertainty, such as credible intervals, about the answers inferred. The sequential update of the KB's prior error is similar to the techniques of sequential estimation using Bayesian methods. We therefore adapt these established techniques to our method in subsequent sections.

B. Coefficient of Variation and credible intervals

We compute uncertainty in the answer from the posterior variance and we normalize it with the mean so that the variation between the inferred and actual for different data items in the RIF tree is not affected by the magnitudes of the data values. The normalized standard deviation is called the *cov* and it is calculated as:

$$cov = \frac{\sigma}{\mu} \tag{1}$$

This is a measure of relative variability and is often expressed as a percentage of the mean. For our use in RIF, this provides four advantages compared with the use of the raw variance or probability values between 0 and 1 used in other inference systems.

(1) Since the *cov* is calculated from the parameters of the Gaussian distribution, it allows us to express the uncertainty in an answer as a credible interval based on the standard deviation of the distribution. Whereas most QAs systems that deal with uncertainty are concerned with discrete assertions, we are interested in statements about continuous or effectively continuous variables. For example, in the domains socio-economic development, many questions deal with populations, GDPs, and other quantities where the probability of any particular

real value will be approximately zero, and all but useless in describing the uncertainty of the value. Credible intervals, however, give a sense of how far an inferred value could be from the truth and provide a meaningful representation of uncertainty for the datasets we work with. By storing *cov*s and posterior means and assuming normality, we retain sufficient statistics for propagating uncertainty up inference trees.

(2) A probability value $\{p|0 \leq p \leq 1\}$ shows how probable the answer is, but does not indicate by how much the answer could be wrong. Many QA systems including IBM's Watson [9], [10] and Microsoft's AskMSR [11], [12] return a probability value or a ranking score attached to their answers. We believe that the semantics of uncertainty in RIF and its representation offers a better interpretation of the accuracy of an answer than a generic score used to rank the candidate answers. That is, we can attribute meaning to RIF's *cov*, whereas a ranking score cannot be easily interpreted relative to the answer that a user sees. Also, since the *cov* is expressed as a percentage of the answer (i.e., the mean value), it easily explains how much error is, potentially, in an answer. So an answer with a higher *cov* is more uncertain than an answer with a lower *cov* and by turning this into a variance, the user can see and upper and lower bound of the answer.

(3) The normalization of the variance as *cov* allows us to compare and propagate variance in the inference tree regardless of the magnitude of the means of the inferred data values. For instance, if the RIF tree has frames containing data for *birth rate* and other frames for *population*, the *cov* allows us to uniformly express the uncertainty of these frames in the same RIF tree, even with data on different scales. This also means that we can compare the credible intervals of answers to queries in a uniform way irrespective of the magnitude of the answers. Raw variance, scaled by the magnitudes of the answer, do not lend themselves to easy comparisons of uncertainties of different answers.

(4) We assign priors to new variance estimates for a given KB heuristically, combining previous *cov* estimates from that KB with new mean estimates. That is, we use the *cov* assigned to a KB's data values as the prior variance of the data values in a KB when we encounter a property for which we have not previously retrieved data. Section IV and the worked example in table II explains this. This approach is preferable to using a full probabilistic model relying on a posterior distribution over raw variances, given that in practice uncertainty and error for individual KB entries tends to scale with the expected value, and these expected values can vary by orders of magnitude.

The *cov* attribute in a RIF frame is dedicated to uncertainty and is propagated with other frame attributes during inference. Our goal is to make RIF update its belief about the uncertainty of facts given what it has previously observed from knowledge bases in a justified way. In the sections that follow, we look at how we calculate uncertainty for real-valued facts.

For real-valued answers, we communicate uncertainty to the user by multiplying *cov* by the answer to obtain the standard deviation from the answer. For example, for a query that returns the answer $27,650,000$ with a *cov* of 15%, implying a .68 probability that the estimate is within $\pm 15\%$ of the correct answer. In raw terms, the interval includes values within $4,147,500$ of the posterior mean.

To be clear, the *cov* is simply a dimensionless, standardized representation of the standard deviation, and our approach does not work well when means are close to zero or when then sign of the true answer is unknown. In practice, this is not a significant problem in our target domain, but we hope to explore this further in future work.

V. Calculating Uncertainty

A. Background: Bayesian Inference and Sequential Estimation

Probabilistic reasoning deals with uncertainty over possible worlds for given random variables. The conditional (posterior) probability of a random variable X given another random variable Y is expressed as:

$$P(x|y) = P(x,y)/P(y) \tag{2}$$

where $P(x,y)$ is the *joint probability* distribution of X and Y, and $P(y)$ is prior probability of Y.

Bayes' Rule is one of the fundamental elements of probabilistic reasoning and is expressed as:

$$P(x|y) \propto P(y|x)P(x) \tag{3}$$

$P(y|x)$ is called the likelihood distribution and $P(x)$ is called the prior distribution.

When a dataset is not fully observed, the approximate posterior distribution of the dataset can be estimated by sequentially updating the posterior distribution in light of new data. The prior parameters of the distribution and the likelihood distribution of the new data values observed are used to estimate the posterior parameters. Conjugate priors simplify the calculations of posteriors. A key factor in choosing a distribution as a conjugate prior is the similarity of its functional form to the likelihood function. We define the two main probability distributions used as likelihood distributions in our work and their respective conjugate priors.

The Gaussian distribution (also referred to as the normal distribution) is a common statistical distribution for continuous random variables. For a random variable X, its Gaussian probability density function is defined by two parameters; a mean μ, and a variance σ^2 in the form:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{1}{2\sigma^2}(x-\mu)^2} \tag{4}$$

The Gamma distribution, parameterized by the shape a and rate b, is commonly used as the prior for the Gaussian distribution.

$$Gam(\lambda|a,b) = \tfrac{1}{\Gamma(a)} b^a \lambda^{a-1} exp(-b\lambda) \tag{5}$$

$$\text{and} \quad \Gamma(x) = \int_o^\infty u^{x-1} e^{-u} du \ , \ \Gamma(1) = 1$$

B. KB Uncertainty for rvd

Our method for calculating uncertainties is based on sequentially estimating Gaussian distributions as new observations, e.g., KB entries (which may be subject to additive noise or rounding error), become available. It is also motivated by ideas from the Expectation-Maximization (EM) algorithm [13]. These techniques use prior distributions based on previously seen data, and then update the priors to get posterior distributions as new data is retrieved. For convenience, we recapitulate formulae for sequentially updating a Gaussian distribution as new data arrive, following [14].

Suppose that for a particular leaf node in the RIF tree, we require data value *f*. Let $s^{(1)}, s^{(2)}, ..., s^{(n)}$ denote KBs 1 through n, and $x_j^{(i)}$ represent the j^{th} fact from KB $s^{(i)}$. We assume that all real-valued observations of a specified property from the KBs are independent and normally distributed with mean, μ, and variance, σ^2. Given that the KB's entire data records is not fully observed by RIF, the normal distribution is a reasonable assumption to make for the data records for each of the real-valued properties in the KB. Although this assumption may not necessarily hold in every case, it allows us to work with existing probabilistic techniques in a reasonable way.

We use a random variable $X^i = \{x_1^{(i)}, x_2^{(i)}, ..., x_j^{(i)}\}$ to represent real-valued facts that have been retrieved for a given property from $s^{(i)}$. Each $x_j^{(i)}$ is normally distributed as,

$$x_j^{(i)} \sim \mathcal{N}(\mu^{(i)}, \sigma^{2(i)}) \tag{6}$$

We refer to the inverse of the variance as *precision* (λ). That is, $\lambda = 1/\sigma^2$.

A Bayesian treatment allows us to introduce priors over the parameters in equation (6) . Our aim is to determine the posterior mean (the assumed true value) and *cov* of a fact given the facts retrieved from KBs. The sequential Bayesian estimation for the Gaussian distribution proceeds in two steps. First, we estimate the mean by assuming a known variance. Next, we do the reverse by assuming that the mean is known and then estimate the variance. The posterior means and variances are then used to calculate the *cov*. Figure 3 shows how we use prior distributions of KB's data.

***STEP 1**: Assume Variance, Estimate Mean:* We begin by using the *cov* associated with the KB to calculate our known variance, σ^2, of the underlying Gaussian distribution of the true value is known. Our goal in this first step is to infer the mean of this distribution. We obtain the 'assumed' variance from the prior *cov* as follows:

$$\sigma^2 = \sum_{i=1}^{k} (cov^{(i)} \times \mu_{ML})^2 \tag{7}$$

where $cov^{(i)}$ is the *cov* associated with KB s_i and μ_{ML} is the maximum likelihood mean of the subset of all data retrieved from KB $s^{(i)}$, where $i \in \{1, \ldots, k\}$.

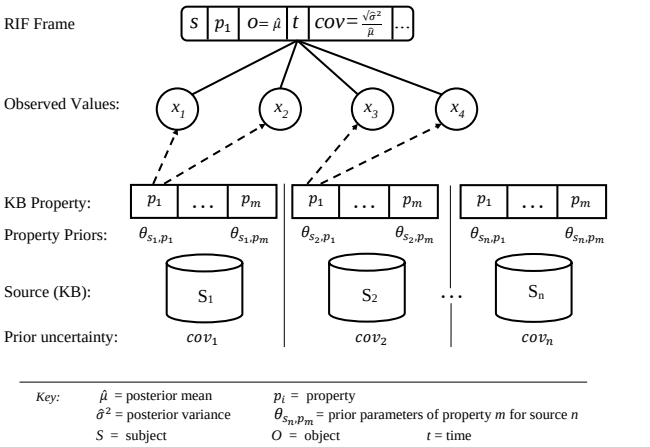

Fig. 3. Prior parameter for data records in KBs and their use in calculating the posterior mean and variance for RIF frame attribute.

Our aim is to find the parameters of the posterior distribution $p(\mu|X)$ given by:

$$\begin{aligned} p(\mu|X) &\propto p(X|\mu)p(\mu) \\ p(\mu|X) &= \mathcal{N}(\mu|\mu_N, \sigma_N^2) \end{aligned} \tag{8}$$

where the prior is given by:

$$p(\mu) = \mathcal{N}(\mu|\mu_0, \sigma_0^2) \tag{9}$$

We use an 'improper' prior μ_0 for each KB by taking the maximum likelihood (ML) mean of the observations from that KB. Ordinarily, the prior mean will be obtained from prior knowledge of the specific data that is retrieved. However, in our case, the dataset in a KB is so diverse that it is impossible to find an existing prior mean. Hence, we use the ML mean of the data that RIF retrieves from a KB as its prior. This prior is close enough to the actual mean and is a better prior than a random value since we usually do not have a prior mean of the data value when we encounter it the first time in a KB. The parameters μ_N and σ_N^2 of the posterior distribution of the mean are given by:

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML} \tag{10}$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2} \tag{11}$$

where $N = |x_{i,1}, ..., x_{i,n}|$ is the number of observations from the KBs in the current iteration.

***STEP 2**: Assume Mean, Estimate Variance:* In the second step, we estimate the uncertainty associated with the distribution of the true value. We assume that the mean is known (we use equation 10 as the 'assumed' mean), so we infer the variance (precision).

Following a similar process as above, our posterior distribution of precision is given by:

$$p(\lambda|X) \propto p(X|\lambda)p(\lambda) \tag{12}$$

and the likelihood function with the given mean μ_N is expressed as:

$$p(X|\lambda) = \prod_{n=1}^{N} \mathcal{N}(x_n|\lambda^{-1})$$
$$= \frac{\lambda^{N/2}}{(2\pi)^{(1/2)}} exp\Big\{ -\frac{\lambda}{2}\sum_{n=1}^{N}(x_n - \mu_N)^2 \Big\} \quad (13)$$

The Gamma distribution (equation 5 in section V-A) is used as the prior for the likelihood function since the likelihood is a function of the product of a power of λ and the exponent of a linear function of λ.

$$\lambda^{(i)} \sim Gam(a^{(i)}, b^{(i)})$$

For each source $s^{(i)}$, we calculate $\lambda^{(i)}$ by using priors on its parameters, i.e., $a_0^{(i)}$ and $b_0^{(i)}$. We obtain the posterior precision by updating the parameters of the Gamma distribution given the new observations as follows:

$$a_N^{(i)} = a_0^{(i)} + N/2 \quad (14)$$
$$b_N^{(i)} = b_0^{(i)} + \frac{1}{2}\sum_{j=1}^{n}(x_j - \mu_N)^2 = b_0^{(i)} + \frac{1}{2}\sigma_{ML}^2$$

where σ_{ML}^2 is the variance of retrieved data from all sources.

The estimate of the posterior precision is given by the expectation of the precision:

$$\lambda_N = \mathbb{E}[\lambda_N] = \frac{a_N}{b_N} \quad (15)$$

Finally, our posterior cov_N is calculated using equations (10) and (15) as:

$$cov_N = \frac{\sqrt{\lambda_N^{-1}}}{\mu_N} \quad (16)$$

The posterior precision is propagated back in the RIF tree and the posterior *cov* is saved and will be used as prior variance in future observations from the KB. Table II shows a worked example for the query *"What is the UK population in 2013"* that follows the above sequence of steps. The posterior mean (line 7) gives us the data value from each KB given the prior variance on data from the KBs from which the value is retrieved. We propagate the posterior *cov*s (line 14) through the RIF tree.

C. Similarities and Difference between RIF the EM Algorithm

RIF's sequential estimation of uncertainty and the vanilla EM algorithm for clustering data are conceptually similar, but different in their objectives and calculations. In RIF, we estimate the posterior mean and variance and assign to frame variables using the prior variances about facts in the web KBs. EM, in the context of clustering, on the other hand deals with the iterative estimation of the maximum likelihood of observed data by estimating the parameters of the distributions and the cluster assignments. We can also map RIF's uncertainty calculation steps to that of the expectation step (E-step) and maximization step (M-step) of the EM algorithm.

TABLE II
KB UNCERTAINTY WORKED EXAMPLE: *"UK population in 2013"*

#	STEP	KB s_1	KB s_2
1	Prior *cov*	1.0	1.0
2	Retrieved data	63900000, 64100000	64000000,63800000, 64200000,63500000
	Step 1: Estimating the posterior mean value of the observation, using the *cov* as the prior variance		
3	$\mu_0 = \mu_{ML}^{(i)}$	64000000	63875000
4	μ_{ML}	63916667	
5	$\sigma_0^2 = (cov \times \mu_{ML})^2$	4.096×10^{15}	4.080×10^{15}
6	σ^2	1×10^{10}	6.6875×10^{10}
7	**Posterior Mean**, μ_N (from eqn 10)	6.3917×10^7	6.3917×10^7
	Step 2: Estimating the posterior *cov*		
8	a_0	1.0	1.0
9	b_0 (set to μ_0)	64000000	63875000
10	σ_{ML}	1.694×10^{10}	7.028×10^{10}
11	a_N (eqn 14)	2.0	3.0
12	b_N (eqn 14)	8.536×10^9	3.520×10^{10}
13	λ_N (eqn 15)	2.343×10^{-10}	8.552×10^{-11}
14	**Posterior *cov***, $cov_N = \frac{\sqrt{\lambda_N^{-1}}}{\mu_N}$	1.002×10^{-3} $= 0.0010\%$	1.695×10^{-3} $= 0.0017\%$

In the E-step, the current values of the parameters of the cluster are used to estimate the posterior probability of each cluster's data points given the observations. In RIF, the priors of the hyperparameters are used to estimate the posterior mean and variance of the observation. In the M-step, the current values of the posteriors are used to re-estimate the parameters of the clusters and re-assign data to clusters such that it maximizes the expected log-likelihood calculated in the E-step. In RIF, we update the hyperparameters (priors) using the posterior mean and variance calculated for the current observations.

D. Inference Operation Uncertainty

Aggregate functions in RIF may contribute to uncertainty in the final answer. We categorize aggregate functions into two types: (1) Exact aggregates and (2) Approximating aggregates

1) Exact Aggregate Functions: These functions perform non-approximating operations on nodes and do not generalize from their child nodes. Examples of such aggregates are MAX, MIN, GT, LT, LOOKUP, VALUES and AVG. Since these aggregates do not approximate their returned values, they do not introduce errors during up-propagation and hence, do not contribute any uncertainty to the final answer. Exact aggregates, however, propagate the uncertainties of their child frames up the RIF tree.

2) Approximating Aggregate Functions: These are aggregates, such as the GP (Gaussian Process Regression) [15], [16] that generate functions from their child nodes in the inference tree and infer new values from them. They take as inputs RIF frames that contain either *values* representing facts, or functions inferred by other methods. Given that these aggregates extrapolate from functions they generate, or

simply reuse previously generated functions, they are likely to introduce errors in the answers returned to their parent nodes.

A GP generates data located through some domain such that any finite subset of the range follows a multivariate Gaussian distribution. The GP gives a distribution over the underlying function that the retrieved data represents. For each independent variable, we can sample the GP for the function. As we generate several samples from this distribution, we observe variances for different independent variable points selected on the function, which results in different function curves. The point variance gives the uncertainty that the GP aggregate function contributes in RIF.

E. Propagating Uncertainty

RIF propagates the uncertainty from its leaf nodes, through intermediate aggregate function nodes, to the root node of the inference tree to obtain the uncertainty of the answer inferred. In addition to the uncertainty that the aggregate functions introduce, the child nodes of the aggregate functions may also contain uncertainties calculated from further down the inference tree. We therefore have to combine uncertainty values to obtain the uncertainty that is propagated from a given node to its parent node.

Given the frame's *cov* and mean value, the variance is known and can be propagated in closed-form up the RIF tree. For example, suppose the aggregate function I sums up two nodes A and B. If A and B have variances σ_a^2 and σ_b^2 respectively, then the variance of combined aggregate function:

$$\sigma_{A+B}^2 = \sigma_A^2 + \sigma_B^2$$

We limit our propagation to inference operations such as SUM, AVG, VALUE and GP (regression) for which there are well understood closed-form combination of the variances of Gaussian distributions. To further simplify our model, we assume conditional independence of the data values of child nodes of a specified node in the inference tree given the inference strategy applied.

VI. Evaluation

Our goal in this evaluation was to demonstrate, using queries over actual web KBs, that the credible intervals that RIF attached to the answers it predicted were properly calibrated and consistent with the actual deviation of the inferred answers from the true answers. In addition to our argument in section IV-B that the *cov* (and in effect, the credible intervals) is appropriate for expressing uncertainty when reasoning over real-valued data, these experiments test our probabilistic method for estimating the *cov*.

We evaluated this hypothesis using real-valued data from the World Bank[3] (with data on over 15,000 different country indicators) and Wikidata [17]. Other supporting KBs used include: DBPedia [18] GeoNames[4], ConceptNet [19] and WordNet [20]. Existing test sets for evaluating query answering systems focus on aspects of the inference process that differ from our objectives in RIF. We were interested in queries that, not only find relevant facts, but also infer non-trivial answers by combining them. We therefore looked at the kinds of questions that are usually asked about demographics and other country development indicators from open data sources such as the World Bank and created a list of test queries to evaluate RIF.

[3]http://databank.worldbank.org, (api endpoint: http://api.worldbank.org)
[4]http://geonames.org

TABLE III
INFERRED ANSWERS FOR THE TRUE GHANA POPULATION VALUE OF $26,962,563$ IN 2014 FOR MULTIPLE YEARS' DATA HELD OUT.

Year(s) Held Out	Inferred Answer	*cov*	|error|
2014	26,962,104	0.0008%	0.0017%
2013, 2014	26,956,515	0.0002%	0.0022%
2012, 2013, 2014	26,957,935	18.53%	0.017%
2011, 2012, 2013, 2014	27,057,470	47.12%	0.35%

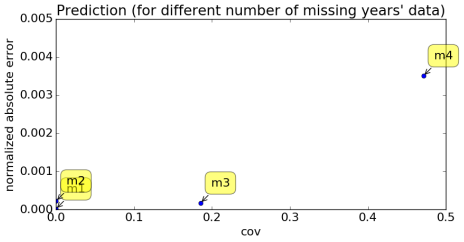

Fig. 4. Prediction *cov* and estimation error plot
A plot of the *cov* of the inferred answer against the normalized ratio of the deviation to the actual answer for prediction of the query *"What is population of Ghana in 2014"*. Different years are held out: $m1 = 2014$, $m2 = 2014, 2013$, $m3 = 2014, 2013, 2012$ and $m4 = 2014, 2013, 2012, 2011$

A. Experiment 1: Prediction

In this experiment, we tested the predictive capabilities of RIF and its ability to attach the appropriate *cov* to the returned answers. Although we can pose queries that predict values for a future date, it is difficult to evaluate when there is no ground truth to compare to. It is worth noting that existing QA benchmark datasets are not well for evaluating RIF's predictive capabilities. We therefore simulated answering prediction queries by holding out all facts after a given time (year). Specifically, this approach simulated a QA environment that assumed that the current year was 2005 and then asked questions about 2014. We asked the query:
{*operation:'value', operation-variable:'?x', subject:'Ghana', property:'population', object:'?x', time:'2014'*} .
We held out all facts after 2005. We used a prior *cov* of 1.0 for all KBs' data values. This assumes the KBs to contain data that is uncertain, with their properties having a standard deviation that is equal to the mean values of the respective properties. The results of this experiment are shown in table III and figure 4.

We observed that, for regression, RIF provides an adequate measure of uncertainty that is well-calibrated such that it is consistent with the absolute errors between the predicted

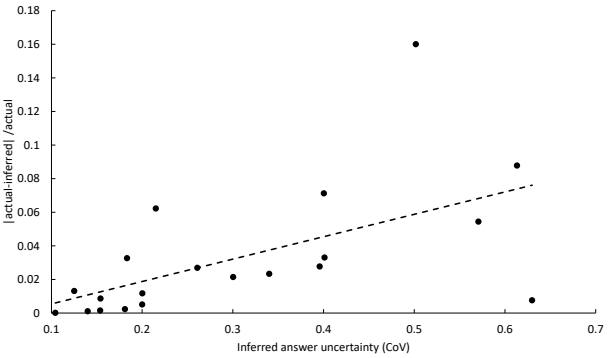

Fig. 5. *cov* and estimation error plot.
A plot of the *cov* of the inferred answer against the normalized ratio of the deviation to the actual answer shows the positive correlation between the two.

answer and the true data value in the KB. Figure 4 shows a positive correlation between the number of missing years' data, the *cov* and the normalized absolute error (|error|) between the held out value and the inferred answer. This also confirms that as number of missing data values needed to calculate an answer increase, the uncertainty in the inferred answer also increases, resulting in a larger *cov*.

B. Experiment 2: Correctness of cov

To check the accuracy of the *cov*s estimated, we generated a set of 60 queries using property terms from the country indicators in the World Bank dataset. Indicators included *population, birth rate, gdp, unemployment, agricultural land area, access to electricity, urban and rural population, cereal export, labour force, energy consumption, arable land*, and *fertility rate*. We used 40 of these queries as a training set during the development of RIF and put the remaining 20 away for these experiments. We plotted the *cov*s against the actual relative errors calculated from the difference between the estimated answer and the held out value from the KB. We used a prior *cov* of 1.0 for all KBs' data values. Similar to the experiment 1, we hid the answers stored in the KB from RIF to force it to infer from other values in the KB.

We observed a positive correlation between the credible interval sizes (*cov*) estimated by RIF and the normalized error between the true value and the answer that was inferred (figure 5). RIF's *cov* increased with the errors in RIF's answers compared to the correct answer in the KB. This shows that RIF's *cov* is adequately captures the intervals within which the truth values of query targets inferred fall. We also observe that the although we started with a uniform prior *cov* of 1.0, individual *cov* values increased as the actual absolute errors between RIF's answer and the hidden true value increased.

VII. Related Work

Probabilistic logic [2] and fuzzy logic [21], [22] are commonly used for reasoning about uncertainty in logic. In classical logic, a sentence can be either *true* or *false* in all possible worlds (models or assignments to variables in the sentence). In the actual world, the sentence can be modelled to account for uncertainty such that it is world w_1 with some probability p and in another world w_2 with probability $p_2 = 1 - p_1$. Probabilistic logic focuses on such uncertain reasoning by a process of probabilistic entailment that formally calculates bounds on the probability of a sentence given a base set of sentences that constitutes its belief about the possible worlds. Also, given a set of sentences and their associated probabilities, probabilistic logic can be used to make new assertions about the world with some posterior probability attached to these assertions. These logics are, however, built for expressing uncertainty about logical formulae, not the specific values within them. They are, therefore, not suited to the kinds of higher order reasoning with real-valued data required in query answering where uncertainty of the values are of interest. More importantly, in RIF, we are unable to capture the variances required to compute our *cov*s. As a result, using these logics (and related ones such as Bayesian Logics Programs [23]), we are unable to express the uncertainty of the answer value.

Although many of the QA systems in the literature exhibit strengths in information retrieval tasks, most fail to attribute an appropriate measure of uncertainty beyond ranking of the answer candidates. Others attribute probabilities to the facts retrieved or inferred. In Ko et.al. [24], evidence from sources (e.g. gazetters and search engines) are used to rank and merge answers. The authors used multiple answering agents to extract candidate answers and developed a unified probabilistic framework to combine multiple evidence to address challenges in ranking and answer merging for factoid questions and questions that return lists. Limitations included the inability of the QA system to scale well to open/multi-domain and multi-knowledge base QA as we do in RIF. DeepQA [9], developed for IBM's Watson, uses a processing pipeline architecture that applies several algorithms that analyse evidence along different dimensions. The evidence evaluation results in confidence values that are used to rank answers.

Out of 20 other QA systems based on SPARQL (including PowerAqua [25], ANGIE [26], ISOFT [27] and Intui3 [28]) and others based on natural language processing and on neural networks, we found 9 with no measure of uncertainty, 7 with a ranking score of the quality of the answers, and 4 with a probability measure of uncertainty.

In databases, probabilistic techniques are also used to deal with uncertainty. The Trio database system [29] manages data, accuracy and lineage of the data, with focus on the inexactness of the data contained. Its is capable of storing uncertain or incomplete data and running queries over them to return answers that may also be inexact, using probabilistic methods to compute these. Trio, however, requires well-defined databases with uncertainty values assigned to the records stored. Work in approximate query processing (AQP) surveyed in [30] surveyed a number of techniques used in AQP over the past few years. These include dynamic sampling [31] and sampling with error estimation [32] where only a subset of the relations

in the database are sampled and used. Such sampling errors are used as uncertainty measures returned with the answers. These database systems, have not been used in web KBs and so most QA systems are unable to leverage the uncertainty values associated with the data they contain.

VIII. Conclusion

When answering queries with data from the Web KBs, errors from source KBs as well as inference operations lead to errors in the final answers. In this work, we demonstrate that it is possible to estimate meaningful credible intervals for answers inferred in RIF by adapting existing probabilistic techniques to our calculations. Our use of the *cov*, the ratio of the standard deviation to mean value, allows RIF to track relevant parameters of the data distributions from which to say useful things about the precision of the answers returned.

Our method allows RIF to calculate uncertainty in real-valued answers as credible intervals, providing simple, interpretable 'error bars' on its answers. This is a useful contribution given the limitations of current QA systems' use of ranking techniques or discrete probabilities to inform the user about the uncertainty in the answers that they infer. Although useful in expressing how probable an answer is, such methods do not clearly show by how much the answer could be wrong.

We show that credible intervals, expressed using *cov* around a posterior mean, are an appropriate way to represent uncertainty in a large class of QA problems since they: (1) offer a simple and intuitive way to express uncertainties about continuous quantities, (2) support incremental and closed-form inferences when composing a variety of inference operations; and (3) are is dimensionless and makes it easy to compare the uncertainty of different answers about widely varying quantities. Our experiments showed that inferred *cov*s were positively correlated with absolute errors, suggesting that the overall model is well-calibrated.

In the future, we hope to weaken some of the assumptions of the distribution of the KB's data records and independence that underpins our current methods. We also plan to deal with uncertainty for non-real-valued answers in RIF. Finally, we will work on generalizing our methods to a widely variety of inference operations and sources of error such as model misspecification.

Acknowledgement

We thank Christopher Lucas for assistance with methodology and for comments that greatly improved the paper.

References

[1] K. Nuamah, A. Bundy, and C. Lucas, "Functional inferences over heterogeneous data," in *International Conference on Web Reasoning and Rule Systems*. Springer, 2016, pp. 159–166.

[2] N. J. Nilsson, "Probabilistic logic," *Artificial intelligence*, vol. 28, no. 1, pp. 71–87, 1986.

[3] D. Beckett and B. McBride, "Rdf/xml syntax specification (revised)," *W3C recommendation*, vol. 10, 2004.

[4] T. Bray, "The javascript object notation (json) data interchange format," 2017.

[5] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *Semantic services, interoperability and web applications: emerging concepts*, pp. 205–227, 2009.

[6] W. E. Winkler, "The state of record linkage and current research problems," in *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[7] A. G. Cohn and S. M. Hazarika, "Qualitative spatial representation and reasoning: An overview," *Fundamenta informaticae*, vol. 46, no. 1-2, pp. 1–29, 2001.

[8] K. Markov, "Extending the Rich Inference Framework in the Energy Domain," 2017, 4th Year Project Report Computer Science, School of Informatics, University of Edinburgh.

[9] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*, "Building watson: An overview of the deepqa project," *AI magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[10] J. W. Murdock, J. Fan, A. Lally, H. Shima, and B. Boguraev, "Textual evidence gathering and analysis," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 8–1, 2012.

[11] E. Brill, J. J. Lin, M. Banko, S. T. Dumais, A. Y. Ng *et al.*, "Data-intensive question answering." in *TREC*, vol. 56, 2001, p. 90.

[12] M. Banko, E. Brill, S. Dumais, J. Lin, and M. Way, "AskMSR: Question answering using the worldwide Web," *Proceedings of 2002 AAAI Spring Symposium on Mining Answers*, pp. 1–2, 2002.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[14] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[15] C. K. Williams and C. E. Rasmussen, "Gaussian processes for regression," *Advances in neural information processing systems*, pp. 514–520, 1996.

[16] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[17] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, pp. 78–85, 2014.

[18] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," *The semantic web*, pp. 722–735, 2007.

[19] R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," in *The Peoples Web Meets NLP*. Springer, 2013, pp. 161–176.

[20] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[21] R. E. Bellman and L. A. Zadeh, "Local and fuzzy logics," in *Modern uses of multiple-valued logic*. Springer, 1977, pp. 103–165.

[22] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy sets and systems*, vol. 90, no. 2, pp. 111–127, 1997.

[23] K. Kersting and L. De Raedt, "Basic principles of learning bayesian logic programs," in *Institute for Computer Science, University of Freiburg*. Citeseer, 2002.

[24] J. Ko, L. Si, and E. Nyberg, "Combining evidence with a probabilistic framework for answer ranking and answer merging in question answering," *Information Processing & Management*, vol. 46, no. 5, pp. 541–554, Sep. 2010.

[25] V. Lopez, M. Fernández, E. Motta, and N. Stieler, "PowerAqua: Supporting users in querying and exploring the Semantic Web," *Semantic Web*, vol. 3, no. 3, pp. 249–265, 2012.

[26] N. Preda, G. Kasneci, F. M. Suchanek, T. Neumann, W. Yuan, and G. Weikum, "Active knowledge: dynamically enriching rdf knowledge bases by web services," pp. 399–410, 2010.

[27] K. Xu, S. Zhang, Y. Feng, and D. Zhao, "Answering natural language questions via phrasal semantic parsing," in *Natural Language Processing and Chinese Computing*. Springer, 2014, pp. 333–344.

[28] C. Dima, "Answering natural language questions with intui3." in *CLEF (Working Notes)*, 2014, pp. 1201–1211.

[29] J. Widom, "Trio: A system for integrated management of data, accuracy, and lineage," Stanford InfoLab, Tech. Rep., 2004.

[30] S. Chaudhuri, B. Ding, and S. Kandula, "Approximate query processing: No silver bullet," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 511–519.

[31] B. Babcock, S. Chaudhuri, and G. Das, "Dynamic sample selection for approximate query processing," in *Proceedings of the 2003 ACM*

SIGMOD international conference on Management of data. ACM, 2003, pp. 539–550.

[32] S. Agarwal, H. Milner, A. Kleiner, A. Talwalkar, M. Jordan, S. Madden, B. Mozafari, and I. Stoica, "Knowing when you're wrong: building fast and reliable approximate query processing systems," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 481–492.