# ArchimedesOne: Query Processing over Probabilistic Knowledge Bases

Xiaofeng Zhou
xiaofeng@cise.ufl.edu

Yang Chen
yang@cise.ufl.edu

Daisy Zhe Wang
daisyw@cise.ufl.edu

Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611-1906

## ABSTRACT

Knowledge bases are becoming increasingly important in structuring and representing information from the web. Meanwhile, web-scale information poses significant scalability and quality challenges to knowledge base systems. To address these challenges, we develop a probabilistic knowledge base system, ARCHIMEDESONE, by scaling up the knowledge expansion and statistical inference algorithms. We design a web interface for users to query and update large knowledge bases.

In this paper, we demonstrate the ARCHIMEDESONE system to showcase its efficient query and inference engines. The demonstration serves two purposes: 1) to provide an interface for users to interact with ARCHIMEDESONE through load, search, and update queries; and 2) to validate our approaches of *knowledge expansion* by applying inference rules in batches using relational operations and *query-driven inference* by focusing computation on the query facts. We compare ARCHIMEDESONE with state-of-the-art approaches using two knowledge bases: NELL-sports with 4.5 million facts and Reverb-Sherlock with 15 million facts.

## 1. INTRODUCTION

Recent development in information extraction and data management systems arouses elevating efforts in constructing large knowledge bases (KBs). These knowledge bases store information in a structured format, facilitating efficient processing and querying. Examples of these knowledge bases include DBpedia, DeepDive, Freebase, Google Knowledge Graph, Knowledge Vault, NELL, OpenIE, ProBase, ProbKB, and YAGO. They store structured information about real-world people, places, organizations, etc, paving the way for the *semantic web* [1] and *semantic search* [4] movement that revolutionizes keyword matching for search.

With the prevalence of web information, these extracted knowledge bases are becoming prohibitively large and are continuously expanding in scale. As of this writing, Freebase has 388 million facts; Reverb has 15 million facts. Despite their scales, the knowledge bases are often incomplete or uncertain due to limitations of human knowledge and the probabilistic nature of information extraction algorithms [3]. In this paper, we present the ARCHIMEDESONE system to manage large-scale automatically harnessed knowledge. ARCHIMEDESONE addresses the *incompleteness* and *uncertainty* challenges in knowledge management: First, knowledge bases contain only subsets of the global knowledge due to limitations of the construction methods–human collaboration and extraction algorithms. Second, machine-constructed knowledge bases contain uncertain or noisy information extracted by probabilistic information extraction algorithms.

ARCHIMEDESONE addresses these challenges by performing the knowledge expansion and query-driven inference tasks:

**Knowledge expansion.** Derive missing and implicit facts using first-order inference rules.

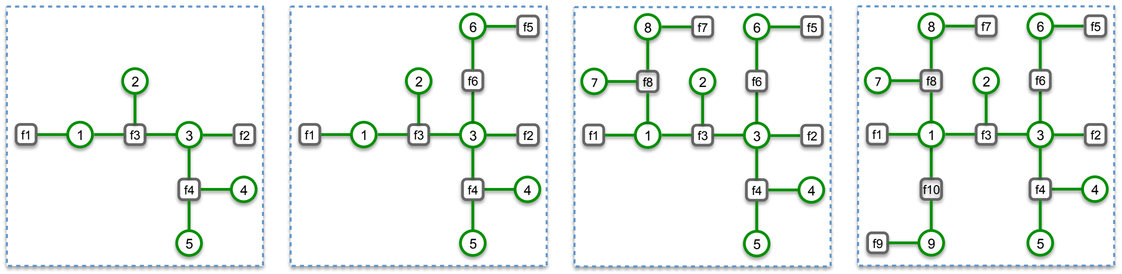
**Query-driven inference.** Compute and update marginal probabilities of query results.

To efficiently perform knowledge expansion, ARCHIMEDESONE utilizes a novel relational model from our work, ProbKB [3], that stores inference rules in relational tables and applies them in batches using SQL queries. It improves Tuffy [6] by storing rules with the same structure in one table. Thus, a single SQL query applies all rules in one rules table in batches, and the number of SQL queries is substantially reduced for large rule sets: for the Sherlock-Reverb dataset, we use 6 queries to apply 30,912 inference rules. Consequently, we improve performance by more than 200 times over Tuffy [3]. To efficiently support inference, ARCHIMEDESONE focuses computation on query nodes to avoid re-computation over the entire knowledge base for each query. Query-driven inference achieves one order of magnitude of speedup for an average query.

ARCHIMEDESONE models uncertain knowledge using Markov logic networks (MLNs) [8]. An MLN consists of weighted first-order clauses to represent uncertain facts and rules. It determines a ground factor graph defining the probability distribution over the base and inferred facts. To answer user queries requires probabilistic inference over MLNs in two steps: grounding (constructing the ground factor graph by applying MLN rules) and inference (computing marginal probabilities). ARCHIMEDESONE efficiently performs these steps by knowledge expansion and query-driven inference. Comparing with the state-of-the-art inference algorithms [7], ARCHIMEDESONE scales to large MLNs.

In this paper, we present the ARCHIMEDESONE probabilistic knowledge base system. We develop a web interface for users to interact with ARCHIMEDESONE through load, search, and update queries, demonstrating its support for real-time queries by performing efficient knowledge expansion and query-driven inference. We implement ARCHIMEDESONE on UDA-GIST [5], a relational database system extended to support common analytics algorithms including MCMC and MC-SAT [7]. The demonstration highlights ARCHIMEDESONE's performance and quality.

*Proceedings of the VLDB Endowment,* Vol. 9, No. 13
Copyright 2016 VLDB Endowment 2150-8097/16/09.



(a) Factor graph at $t_1$ (b) Factor graph at $t_2$ (c) Factor graph at $t_3$ (d) Factor graph at $t_4$

ground predicate table

| Predicate ID | Ground Predicate |
|---|---|
| 1 | Obama *isBornInState* Hawaii |
| 2 | Hawaii *isAStateOf* USA |
| 3 | Obama *isBornInCountry* USA |
| 4 | Obama *hasACitizenshipOf* USA |
| 5 | Obama *isEligibleToBePresidentOf* USA |
| 6 | Obama *isBornInCountry* Kenya |
| 7 | Star-Bulletin *isLocatedAt* Hawaii |
| 8 | Star-Bulletin *advertiseTheBirthOf* Obama |
| 9 | Obama *hasBirthCertificateIn* Hawaii |
| - | - |

factor table

| Factor ID | Clause | $p(t_1)$ | $p(t_2)$ | $p(t_3)$ | $p(t_4)$ |
|---|---|---|---|---|---|
| $f_1$ | 1 | 1.00 | 0.70 | | |
| $f_2$ | 3 | 1.00 | 0.70 | | |
| $f_3$ | $1 \wedge 2 \rightarrow 3$ | 1.00 | | | |
| $f_4$ | $3 \wedge 4 \rightarrow 5$ | 0.95 | | | |
| $f_5$ | 6 | - | 0.30 | | |
| $f_6$ | $6 \rightarrow \neg 3$ | - | 1.00 | | |
| $f_7$ | 8 | - | | 0.95 | |
| $f_8$ | $7 \wedge 8 \rightarrow 1$ | - | | 1.00 | |
| $f_9$ | 9 | - | | | 0.95 |
| $f_{10}$ | $9 \rightarrow 1$ | - | | | 1.00 |

**Figure 1: Probabilistic knowledge base of Barack Obama citizenship conspiracy theories. (a)-(d) Ground factor graphs at $t_1$ to $t_4$.**

## 2. PROBABILISTIC KNOWLEDGE BASES

A probabilistic knowledge base is a knowledge base that supports uncertain facts and rules. The support for uncertainty is essential for representing automatically constructed knowledge bases since they contain facts and rules mined by probabilistic information extraction algorithms. In Figure 1, we show a probabilistic knowledge base of Barack Obama citizenship conspiracy theories constructed from a Wikipedia page of the events [9].

**Example 1.** Before Obama's presidential campaign, denoted by timestamp $t_1$, public information showed that he was born in Hawaii. At timestamp $t_2$, anonymous emails questioned Obama's birth place and indicated that *"Obama isBornInCountry Kenya."* Jim Geraghty of the National Review Online sparked further speculation. This information led to a contradiction about Obama's birthplace. As described in Figure 1, *"Obama isBornInCountry Kenya"* is extracted with a confidence value of 0.3 and *"Obama isBornInCountry USA"* is extracted with a confidence value of 0.7. At the next timestamp $t_3$, the local newspaper Star-Bulletin advertised the birth place of Obama. Finally, the tempest died away with the release of Obama's birth certificate from the Hawaii Department of Health.

To answer the query *"Obama isBornInCountry USA,"* we need to re-compute its probability based on all current evidence since previously computed probabilities are outdated with the emergence of new evidence. With the data described in Figure 1, the query *"Obama isBornInCountry USA"* returns probabilities 1.00, 0.84, 0.90 and 0.97 in the four snapshots. $\square$

We formally define a probabilistic knowledge base as a database of facts and a set of probabilistic first-order formulae, represented by a Markov logic network.

### 2.1 Markov Logic Networks

Markov logic networks represent uncertain facts and rules by weighted first-order clauses. Essentially, an MLN is a set of weighted first-order formulae $\{(F_i, W_i)\}$, the weight $W_i$ indicating how likely the formula $F_i$ is true. For example, the following weighted formulae form an MLN representing knowledge about Barack Obama's birth place and rules to expand the knowledge:

0.70 isBornInCountry(Barack Obama, USA)
1.00 isBornInState$(x, z)$, isAStateOf$(z, y) \rightarrow$ isBornInCountry$(x, y)$
1.00 isBornInCountry$(z, x) \wedge$ isBornInCountry$(z, y) \rightarrow x = y$

They state a fact that Barack Obama was born in the USA, a rule that if a person $x$ was born in state $z$ and the state $z$ is located in country $y$, then the person $x$ was born in country $y$, and a constraint that a person was born in only one country. The weights 0.70 and 1.00 specify how strong the rules are; stronger rules are more likely satisfied by the knowledge base.

### 2.2 Inference

An MLN can be viewed as a template for constructing ground factor graphs. A *factor graph* is a set of factors $\Phi = \{\phi_1, \ldots, \phi_N\}$, where each factor $\phi_i$ is a function $\phi_i(\mathbf{X}_i)$ over a random vector $\mathbf{X}_i$ indicating the causal relationships among the random variables in $\mathbf{X}_i$. In Figures 1(a)-(d), we show ground factor graphs for the knowledge base at timestamps $t_1$ to $t_4$. Each factor represents a ground rule–e.g., factor $f_3$ represents the rule isBornInState(Obama, Hawaii), isAStateOf(Hawaii, USA) $\rightarrow$ isBornInCountry(Obama, USA). The process of constructing the ground factor graph from an MLN is called *grounding* [8].

In a factor graph $\Phi = \{\phi_1, \ldots, \phi_N\}$, the factors together determine a joint probability distribution over the random vector $\mathbf{X}$ consisting of all the random variables in the factor graph:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_i \phi_i(\mathbf{X}_i) = \frac{1}{Z} \exp\left(\sum_i W_i n_i(\mathbf{x})\right), \quad (1)$$

where $n_i(\mathbf{x})$ is the number of true groundings of rule $F_i$ in $\mathbf{x}$, $W_i$ is its weight, and $Z$ is the partition function, i.e., normalization constant. ARCHIMEDESONE answers user queries by computing $P(X = x)$, the marginal distribution of a query node $X$ defined by (1). This is called *marginal inference* of probabilistic graphical models. Exact inference in MLNs is intractable [8], and state-of-the-art approaches use sampling algorithms including MCMC and MC-SAT [8, 7, 6]. ARCHIMEDESONE uses a query-driven approach to focus MCMC sampling on the query nodes. As illustrated in Figure 1, when we update the factor graph, the marginal probabilities it defines also get updated. Query-driven sampling handles these updates by avoiding re-computation over the entire graph.

## 3. SYSTEM OVERVIEW

ARCHIMEDESONE models facts, rules, and the factor graph in relational tables. This relational model enables ARCHIMEDESONE to efficiently perform knowledge expansion and query-driven inference using join queries in the UDA-GIST in-database analytics framework [5]. Users interact with ARCHIMEDESONE through load, search, and update queries on a web user interface. The ARCHIMEDESONE architecture is shown in Figure 2.

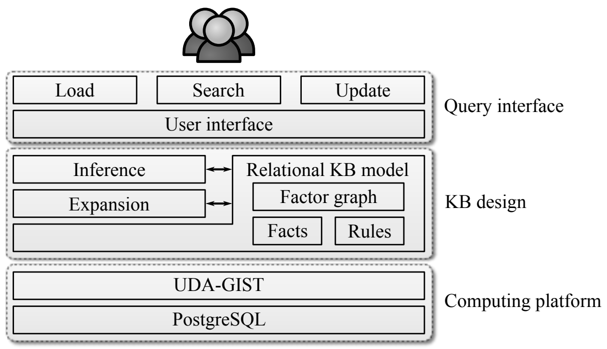


**Figure 2: Users interact with ARCHIMEDESONE through a web user interface to retrieve results from knowledge expansion and query-driven inference, implemented in UDA-GIST.**

### 3.1 Knowledge Expansion

In ARCHIMEDESONE, we represent a knowledge base as relational tables. This relational model is first introduced by ProbKB [3] and proves efficient in rule mining [2] by applying inference rules in batches using join queries. The main challenge with inference rules is that they have flexible structures. To adapt for their structures, we utilize structural equivalence to divide rules into equivalent classes so that each equivalent class has a fixed table format.

In particular, we call two first-order clauses *structurally equivalent* if they differ only in entities, types, and predicates. To illustrate, the following rules are structurally equivalent:

isBornInState$(x, z)$, isAStateOf$(z, y) \to$ isBornInCountry$(x, y)$
isBornInCity$(x, z)$, isACityOf$(z, y) \to$ isBornInState$(x, y)$

It is verifiable that the structural equivalence relation is an equivalence relation. Thus, first-order clauses can be divided into equivalent classes accordingly. According to the definition, each rule is identified by the differing entities, types, or predicates. Thus, each equivalent class can be stored in a fixed-column table, with the columns being entities, types, and predicates.

Based on the relational model, we express the knowledge expansion algorithm as join queries between the facts and rules tables, one join for each rules table. Our experiments show that applying rules in batches results in a 200-300 times of speedup over the state-of-the-art approaches [3]. The result of knowledge expansion is a ground factor graph $\Phi = \{\phi_1, \ldots, \phi_N\}$, where each factor $\phi_i(\mathbf{X}_i)$ represents a ground rule. The factor graph is modeled by a relational table, the columns storing predicate IDs of variables $X \in \mathbf{X}$ and weights of the factors. Performing inference on this factor graph yields marginal probabilities of the query facts.

### 3.2 Query-Driven Inference

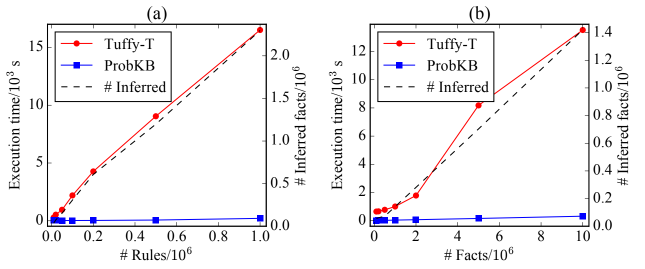
ARCHIMEDESONE uses query-driven inference to speed up MLN inference algorithms by focusing computation on the query facts. The query-driven inference algorithm is designed with the UDA-GIST analytics framework [5] to achieve efficient inference in a relational database system. Furthermore, we use $K$-hop approximation to focus computation on the query facts.

**UDA-GIST.** We implement query-driven inference using the UDA-GIST in-database analytics framework [5]. UDA-GIST utilizes User Defined Aggregates (UDAs) and extends it with a new operator, General Iterative State Transition (GIST), that performs iterative transitions of computation over a large state. UDA-GIST combined extends relational database systems by allowing users to define UDAs and GISTs capable of expressing complex analytics algorithms including MCMC and MC-SAT.

**$K$-hop approximation.** To achieve real-time response, we approximate inference by extracting $K$-hop sub-networks of the ground factor graph, consisting of nodes within $K$ hops from the query nodes. The $K$-hop approximation is based on the observation that neighbors of the query nodes have more influence than distant nodes. To achieve real-time response, we use an additional network limit parameter to control the expansion of $K$-hop sub-networks as $K$ increases. In Section 3.3, we achieve an 18 times of speedup compared to inference over the entire factor graph by choosing $K = 2$, with an acceptable error of 0.04 in probabilities.

### 3.3 Experiments

We evaluate ARCHIMEDESONE using the Reverb-Sherlock Wikipedia knowledge base [3] with 407,247 facts and 30,912 first-order inference rules and a synthetic knowledge base with varying numbers of facts and rules ranging from 10K to 10M.



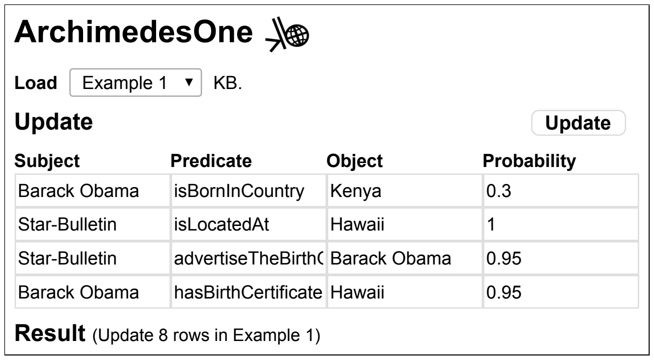
(c)

| $K$ | 1 | 2 | 5 | 10 | 15 |
|---|---|---|---|---|---|
| Runtime | 1.31 | 25.82 | 31.23 | 31.29 | 31.31 |
| Error | 0.09 | 0.04 | 0.04 | 0.04 | 0.04 |

**Figure 4: Efficiency benchmark results. (a)(b) Knowledge expansion. Tuffy-T refers to our implementation of Tuffy to support typed rules. (c) Query-driven inference.**

**Knowledge expansion.** We use Tuffy [6] as the baseline comparison. Figures 4(a)(b) compare performance of ARCHIMEDESONE with Tuffy on the synthetic knowledge base with varying numbers of facts and rules. We see that ARCHIMEDESONE achieves more than 200 times of speedup over Tuffy for $10^7$ facts. The speedup benefits from the batch application of rules with join operations supported by the relational knowledge base model.

**Query-driven inference.** We evaluate query-driven inference on the Reverb-Sherlock knowledge base by varying $K$ from 1 to 15 and setting network_limit to 1000 in the $K$-hop approximation. The result is reported in Figure 4(c). By setting $K = 2$, we achieve an 18 times of speedup with an acceptable error of 0.04 in computed probabilities. The runtime becomes stable when $K \geq 5$ as the network size reaches the network_limit.

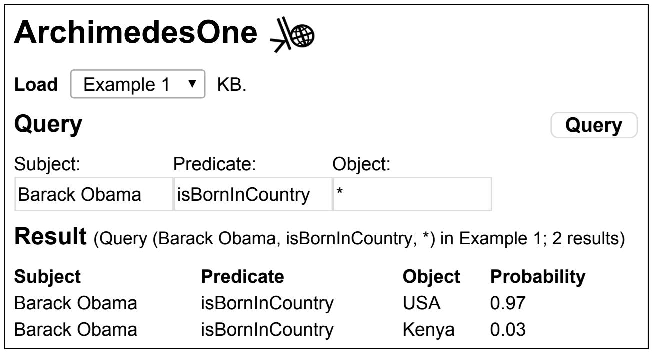


Load Example 1 KB.

Update

| Subject | Predicate | Object | Probability |
|---|---|---|---|
| Barack Obama | isBornInCountry | Kenya | 0.3 |
| Star-Bulletin | isLocatedAt | Hawaii | 1 |
| Star-Bulletin | advertiseTheBirth( | Barack Obama | 0.95 |
| Barack Obama | hasBirthCertificate | Hawaii | 0.95 |

**Result** (Update 8 rows in Example 1)



Load Example 1 KB.

Query

| Subject: | Predicate: | Object: |
|---|---|---|
| Barack Obama | isBornInCountry | * |

**Result** (Query (Barack Obama, isBornInCountry, *) in Example 1; 2 results)

| Subject | Predicate | Object | Probability |
|---|---|---|---|
| Barack Obama | isBornInCountry | USA | 0.97 |
| Barack Obama | isBornInCountry | Kenya | 0.03 |

**Figure 3: ARCHIMEDESONE user interface. (a) Updating ARCHIMEDESONE by adding facts. (b) Query results at time $t_4$. ARCHIMEDESONE determines the probabilities of Obama's birth places by aggregating current and previous evidence.**

## 4. DEMONSTRATION

We demonstrate the ARCHIMEDESONE system with a web interface to query NELL-Sports and Reverb-Sherlock KBs, featuring its knowledge expansion and query-driven inference engines.

**NELL-Sports.** The NELL candidate belief dataset (up to iteration 910) contains 84.6 million facts and 1828 rules in the sports domain. We remove duplicate rules and rules with predicate "generalization" as they are beyond the sports domain. We use the 4.5 million facts after pre-processing the candidate belief dataset.

**Reverb-Sherlock.** Reverb-Sherlock is an automatically constructed knowledge base by information extraction and inductive logic programming. It is extracted from the ClueWeb, containing 15M facts and 30,912 first-order inference rules. The Reverb-Sherlock knowledge base covers general domains, including people, locations, films, food, sports, etc.

**User interface.** We develop a website for users to interact with ARCHIMEDESONE. Figure 3 shows a sample session where the user loads a knowledge base, incrementally updates the knowledge base, and queries ARCHIMEDESONE with a partial triple. ARCHIMEDESONE returns the missing values and probabilities.

### 4.1 Demo Scenarios

VLDB participants attending the ARCHIMEDESONE demonstration can query the NELL-Sports and Reverb-Sherlock KBs through its web user interface shown in Figure 3. ARCHIMEDESONE supports load, search, and update queries:

**Load.** Import a knowledge base of facts and rules into ARCHIMEDESONE and initiate the knowledge expansion task. ARCHIMEDESONE efficiently applies inference rules in batches and derives new facts from the imported knowledge base.

**Search.** Post a triple with a missing value to query, e.g., (Barack Obama, isBornInCountry, $\cdot$) asking for the birth country of Barack Obama. ARCHIMEDESONE searches in the expanded knowledge graph for the answers and performs query-driven inference to compute the probability of each answer.

**Update.** Add or update new facts or rules to ARCHIMEDESONE. In Figure 3, the user updates birth information of Barack Obama according to Example 1. Before the update, ARCHIMEDESONE would return a probability of 1.0 for *"Obama isBornInCountry USA."* After the update, the probability has changed to 0.97.

Figure 3 shows a sample session where the user queries and updates the knowledge base in Example 1. Initially, ARCHIMEDESONE knows that Barack Obama was born in Hawaii, USA. In Figure 3(a), the user updates ARCHIMEDESONE by providing additional information that Obama was born in Kenya from the anonymous emails with probability 0.3. The user also provides an additional certificate that Obama was born in Hawaii with probability 0.95. As a result, ARCHIMEDESONE initiates query-driven inference and updates the birth place of Obama to USA with probability 0.97. In a query asking for Obama's birth place in Figure 3(b), ARCHIMEDESONE returns the new information to the user.

### 4.2 Performance Comparison

In the ARCHIMEDESONE demo, we compare different algorithms for knowledge expansion and query-driven inference.

**Knowledge expansion.** ARCHIMEDESONE improves performance by storing structurally equivalent rules [3] in one relational table. Thus, a single SQL query applies one table of rules in batches. By comparing with Tuffy [6], we show that a single batch query is more efficient than an equivalent sequence of queries.

**Query-driven inference.** We compare two inference algorithms: Gibbs sampling and MC-SAT [7] with $K$-hop sub-network approximation with different $K$s and limits of the network size. We show that focusing computation on the query nodes achieves a considerable performance improvement.

**Acknowledgements.** We gratefully acknowledge the support of NSF under IIS Award # 1526753, DARPA under FA8750-12-2-0348-2 (DEFT/CUBISM), and a generous gift from Google. We also thank Dr. Milenko Petrovic and Dr. Alin Dobra for the helpful discussions on query optimization.

## 5. REFERENCES

[1] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific American*, 2001.

[2] Y. Chen, S. Goldberg, D. Z. Wang, and S. S. Johri. Ontological pathfinding: Mining first-order knowledge from large knowledge bases. In *SIGMOD*. ACM, 2016.

[3] Y. Chen and D. Z. Wang. Knowledge expansion over probabilistic knowledge bases. In *SIGMOD*. ACM, 2014.

[4] O. Etzioni. Search needs a shake-up. *Nature*, 2011.

[5] K. Li, D. Z. Wang, A. Dobra, and C. Dudley. Uda-gist: an in-database framework to unify data-parallel and state-parallel analytics. *Proceedings of the VLDB Endowment*, 2015.

[6] F. Niu, C. Ré, A. Doan, and J. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *Proceedings of the VLDB Endowment*, 2011.

[7] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI*, 2006.

[8] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 2006.

[9] Wikipedia. Barack obama citizenship conspiracy theories, 2015. Online; accessed 24-Feb-2016.