KGEval: Accuracy Estimation of Automatically Constructed Knowledge Graphs

Prakhar Ojha
Indian Institute of Science
prakhar@iisc.ac.in

Partha Talukdar
Indian Institute of Science
ppt@iisc.ac.in

Abstract

Automatic construction of large knowledge graphs (KG) by mining web-scale text datasets has received considerable attention recently. Estimating accuracy of such automatically constructed KGs is a challenging problem due to their size and diversity and has largely been ignored in prior research. In this work, we try to fill this gap by proposing KGEval. KGEval uses *coupling constraints* to bind facts and crowdsource those few that can *infer* large parts of the graph. We demonstrate that the objective optimized by KGEval is submodular and NP-hard, allowing guarantees for our approximation algorithm. Through experiments on real-world datasets, we demonstrate that KGEval best estimates KG accuracy compared to other baselines, while requiring significantly lesser number of human evaluations.

1 Introduction

Automatic construction of Knowledge Graphs (KGs) from Web documents has received significant interest over the last few years, resulting in the development of several large KGs consisting of hundreds of predicates (e.g., *isCity*, *stadiumLocatedInCity(Stadium, City)*) and millions of their instances called beliefs (e.g., *(Joe Luis Arena, stadiumLocatedInCity, Detroit)*). Examples of such KGs include NELL (Mitchell et al., 2015), Knowledge-Vault (Dong et al., 2014) etc.

Due to imperfections in the automatic KG construction process, many incorrect beliefs are also found in these KGs. Knowing accuracy for each predicate in the KG can provide targeted feedback for improvement and highlight its strengths from weaknesses, whereas overall accuracy of a KG can quantify the effectiveness of its construction-process. Knowing accuracy at predicate-level granularity is immensely helpful for Question-Answering (QA) systems that integrate opinions from multiple KGs (Samadi et al., 2015). For such systems, being aware that a particular KG is more accurate than others in a certain domain, say sports, helps in restricting the search over relevant and accurate subsets of KGs, thereby improving QA-precision and response time. In comparison to the large body of recent work focused on construction of KGs, the important problem of accuracy estimation of such large KGs is unexplored – we address this gap in this paper.

True accuracy of a predicate may be estimated by aggregating human judgments on correctness of each and every belief in the predicate[1]. Even though crowdsourcing marketplaces such as Amazon Mechanical Turk (AMT) provide a convenient way to collect human judgments, accumulating such judgments at the scale of larges KGs is prohibitively expensive. We shall refer to the task of manually classifying a single belief as true or false as a Belief Evaluation Task (BET). Thus, the crucial problem is: *How can we select a subset of beliefs to evaluate which will best estimate the true (but unknown) accuracy of KG and its predicates?*

A naive and popular approach is to evaluate randomly sampled subset of beliefs from the KG. Since random sampling ignores relational-couplings present among the beliefs, it usually results in oversampling and poor accuracy estimates. Let us motivate this through an example.

[1] Note that belief evaluation can not be completely automated and will require human-judgment. If an algorithm could accurately predict correctness of a belief, then it may as well be used during KG construction rather than during evaluation.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1741–1750
Copenhagen, Denmark, September 7–11, 2017.

Motivating example: We motivate efficient accuracy estimation through the KG fragment shown in Figure 1. Here, each belief is an edge-triple in the graph, for example (*RedWings, isA, SportsTeam*). There are six correct and two incorrect beliefs (the two incident on *Taj Mahal*), resulting in an overall accuracy of $75\%(= 6/8)$ which we would like to estimate. Additionally, we would also like to estimate accuracies of the predicates: *homeStadiumOf*, *homeCity*, *stadiumLocatedInCity*, *cityInState* and *isA*.

We now demonstrate how evaluation labels of beliefs are constrained by each other. *Type consistency* is one such coupling constraint. For instance, we know from KG ontology that the *homeStadiumOf* predicate connects an entity from *Stadium* category to another entity in *Sports Team* category. Now, if *(Joe Louis Arena, homeStadiumOf, Red Wings)* is evaluated to be correct, then from these *type constraints* we can infer that *(Joe Louis Arena, isA, Stadium)* and *(Red Wings, isA, Sports Team)* are also correct. Similarly, by evaluating *(Taj Mahal, isA, State)* as false, we can infer that *(Detroit, cityInState, TajMahal)* is incorrect.

Additionally, we have *Horn-clause coupling constraints* (Mitchell et al., 2015; Lao et al., 2011), such as *homeStadiumOf(x, y)* $\wedge$ *homeCity(y, z)* $\rightarrow$ *stadiumLocatedInCity(x, z)*. By evaluating *(Red Wings, homeCity, Detroit)* and applying this horn-clause to the already evaluated facts mentioned above, we infer that *(Joe Louis Arena, stadiumLocatedInCity, Detroit)* is also correct. We explore generalized forms of these constraints in Section 3.1.

Thus, evaluating only three beliefs, and exploiting constraints among them, we exactly estimate the *overall* true accuracy as 75% and also cover all predicates. In contrast, the empirical accuracy by randomly evaluating three beliefs, averaged over 5 trials, is 66.7%.

Our contributions in this paper are the following: (1). Systematic study into the important problem of evaluation of automatically constructed Knowledge Graphs. (2). A novel crowdsourcing-based system KGEval to estimate accuracy of large knowledge graphs (KGs) by exploiting dependencies among beliefs for more accurate and faster KG accuracy estimation. (3). Extensive experiments on real-world KGs to demonstrate KGEval's effectiveness and also evaluate its robustness and scalability.

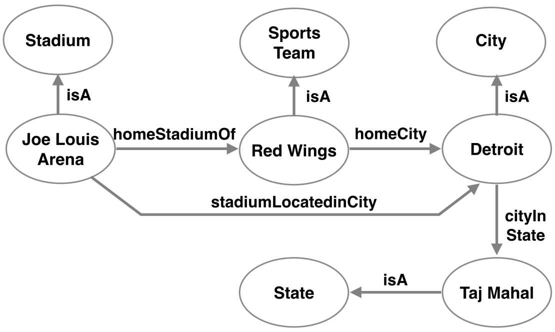

Figure 1: Sample knowledge-graph (KG) fragment that is consistent but has erroneous beliefs.

All the data and code used in the paper are available at `http://talukdar.net/mall-lab/KGEval`

2 Overview and Problem Statement

2.1 KGEval: Overview

We try to estimate correctness of as many beliefs as possible while evaluating only a subset of them through crowdsourcing. KGEval achieves this goal using an iterative algorithm which alternates between the following two stages:

- **Control Mechanism** (Section 3.4): In this step, KGEval selects the belief which is to be evaluated next using crowdsourcing.
- **Inference Mechanism** (Section 3.3): Coupling constraints are applied over evaluated beliefs to automatically estimate correctness of additional beliefs.

This iterative process is repeated until there are no more beliefs to be evaluated. Single iteration of KGEval over the KG fragment from Figure 1 is shown in Figure 2 where, belief *(John Louis Arena, homeStadiumOf, Red Wings)* is selected and evaluated by crowdsourcing. Subsequently, the inference mechanism uses type coupling constraints to infer *(JL Arena, isA, Stadium)* and *(R. Wings, isA, Team)* also as true. Next, we formalize the notations used in this paper.

2.2 Notations and Problem Statement

We are given a KG with n beliefs. Evaluating a single belief as true or false forms a Belief Evaluation Task (BET). Coupling constraints are derived by determining relationships among BETs, which we further discuss in Section 3.1. Notations are also summarized in Table 1.

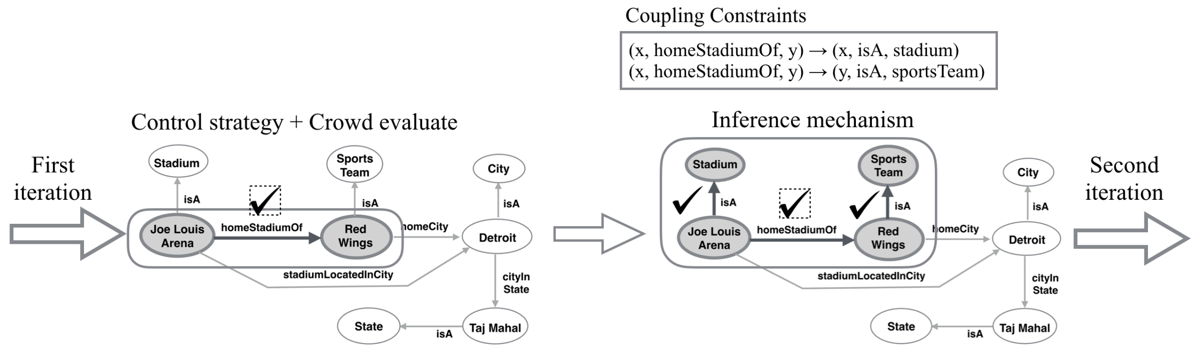

Figure 2: Demonstration of one iteration of KGEval. Control mechanism selects a belief whose correctness is evaluated from crowd. In the above example, *(J.L. Arena, homeStadiumOf, Red Wings)* is crowd-evaluated to be true (indicated by tick with dotted square). (Section 2.1 and Section 3).

Symbol	Description
$\mathcal{H} = \{h_1, \ldots, h_n\}$	Set of all n Belief Evaluation Tasks (BETs)
$c(h) \in \mathbb{R}_+$	Cost of labeling h from crowd
$\mathcal{C} = \{(\mathcal{C}_i, \theta_i)\}$	Coupling constraints $\mathcal{C}_i$ with weights $\theta_i \in \mathbb{R}_+$
$t(h) \in \{0, 1\}$	True label of h
$l(h) \in \{0, 1\}$	Estimated label of h
$\mathcal{H}_i = \mathrm{Dom}(\mathcal{C}_i)$	$\mathcal{H}_i \subseteq \mathcal{H}$ which participate in $\mathcal{C}_i$
$G = (\mathcal{H} \cup \mathcal{C}, \mathcal{E})$	Evaluation Coupling Graph, $e \in \mathcal{E}$ between $\mathcal{H}_j$ and $\mathcal{C}_j$ denotes $\mathcal{H}_j \in \mathrm{Dom}(\mathcal{C}_j)$.
$\mathcal{Q} \subseteq \mathcal{H}$	BETs evaluated using crowd
$\mathcal{I}(G, \mathcal{Q}) \subseteq \mathcal{H}$	Inferable set for evidence $\mathcal{Q}$:
$\Phi(\mathcal{Q}) = \frac{1}{\lvert\mathcal{Q}\rvert} \sum_{h \in \mathcal{Q}} t(h)$	True accuracy of evaluated BETs $\mathcal{Q}$

Table 1: Summary of notations used (Section 2.2).

Inference algorithm helps us work out evaluation labels of other BETs using constraints $\mathcal{C}$. For a set of already evaluated BETs $\mathcal{Q} \subseteq \mathcal{H}$, we define *inferable set* $\mathcal{I}(G, \mathcal{Q}) \subseteq \mathcal{H}$ as BETs whose evaluation labels can be deduced by the inference algorithm. We calculate the average true accuracy of a given set of evaluated BETs $\mathcal{Q} \subseteq \mathcal{I}(G, \mathcal{Q}) \subseteq \mathcal{H}$ by $\Phi(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{h \in \mathcal{Q}} t(h)$.

KGEval aims to sample and crowdsource a BET set $\mathcal{Q}$ with the largest inferable set, and solves the optimization problem:

$$\arg\max_{\mathcal{Q} \subseteq \mathcal{H}} \left|\mathcal{I}(G, \mathcal{Q})\right| \tag{1}$$

3 KGEval: Method Details

In this section, we describe various components of KGEval.

3.1 Coupling Constraints

The evaluation labels of beliefs are often dependent on each other due to rich relational structure of KGs. In this work, we derive coupling constraints $\mathcal{C}$ from the KG ontology and link-prediction algorithms, such as PRA (Lao et al., 2011) over NELL and AMIE (Galárraga et al., 2013) over Yago. These rules are jointly learned over entire KG with millions of facts and are assumed true.

We use conjunction-form first-order-logic rules and refer to them as *Horn clauses*. Examples of a few such coupling constraints are shown below.

$\mathcal{C}_2$: *(x, homeStadiumOf, y) → (y, isA, sportsTeam)*
$\mathcal{C}_5$: *(x, homeStadiumOf, y) ∧ (y, homeCity, z) →*
(x, stadiumLocatedInCity, z)

Each coupling constraint $\mathcal{C}_i$ operates over $\mathcal{H}_i \subseteq \mathcal{H}$ to the left of its arrow and infers label of the BET on the right of its arrow. $\mathcal{C}_2$ enforces *type consistency* and $\mathcal{C}_5$ is an instance of PRA path. These constraints have also been successfully employed earlier during knowledge extraction (Mitchell et al., 2015) and integration (Pujara et al., 2013). Note that the constraints are *directional* and inference propagates in forward direction.

3.2 Evaluation Coupling Graph (ECG)

To combine all beliefs and constraints at a common place, for all $\mathcal{H}$ and $\mathcal{C}$, we construct a graph with two types of nodes: (1) a node for each BET $h \in \mathcal{H}$, and (2) a node for each constraint $\mathcal{C}_i \in \mathcal{C}$. Each $\mathcal{C}_i$ node is connected to all h nodes that participate in it. We call this graph the *Evaluation Coupling Graph* (*ECG*), represented as $G = (\mathcal{H} \cup \mathcal{C}, \mathcal{E})$ with set of edges $\mathcal{E} = \{(\mathcal{C}_i, h) \mid h \in \mathrm{Dom}(\mathcal{C}_i)\ \forall \mathcal{C}_i \in \mathcal{C}\}$. Note that ECG is a bipartite *factor graph* (Kschischang et al., 2001) with h as variable-nodes and $\mathcal{C}_i$ as factor-nodes.

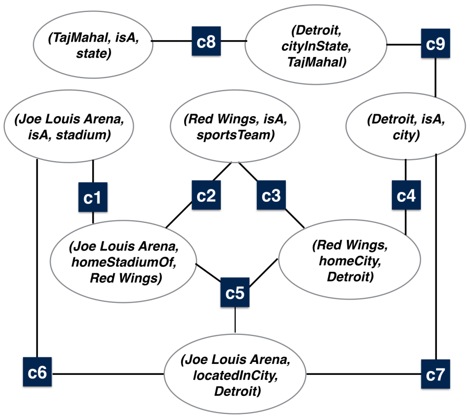
(TajMahal, isA, state)
c8
(Detroit, cityInState, TajMahal)
c9
(Joe Louis Arena, isA, stadium)
(Red Wings, isA, sportsTeam)
(Detroit, isA, city)
c1
c2
c3
c4
(Joe Louis Arena, homeStadiumOf, Red Wings)
(Red Wings, homeCity, Detroit)
c5
c6
(Joe Louis Arena, locatedInCity, Detroit)
c7

Figure 3: Evaluation Coupling Graph (ECG) constructed for example in Figure 1. (Section 3.2)

Figure 3 shows ECG constructed out of the motivating example in Figure 1 with $|\mathcal{C}| = 8$ and separate nodes for each of the edges (beliefs or BETs) in KG. We pose the KG evaluation problem as classification of BET nodes in the ECG by allotting them a label of 1 or 0 to represent true or false respectively.

3.3 Inference Mechanism

Inference mechanism helps propagate true/false labels of evaluated beliefs to other non-evaluated beliefs using available coupling constraints (Bragg et al., 2013). We use Probabilistic Soft Logic (PSL), (Broecheler et al., 2010) as our inference engine.Below we briefly describe the internal workings of our inference engine for accuracy estimation problem.

Potential function ψ_j is defined for each $\mathcal{C}_j$ using Lukaseiwicz t-norm and it depicts how satisfactorily constraint $\mathcal{C}_j$ is satisfied. For example, $\mathcal{C}_5$ mentioned earlier is transformed from first-order logical form to a real valued number by

$$\psi_j(\mathcal{C}_5) = \big(\max\{0, h_x + h_y - 1 - h_w\}\big)^2 \quad (2)$$

where $\mathcal{C}_5 = h_x \wedge h_y \rightarrow h_w$, where h_x denotes the evaluation score $\in [0,1]$ associated with the BETs.

The probability distribution over label assignment is so structured such that labels which satisfy more coupling constraints are more probable. Probability of any label assignment $\Omega(\mathcal{H}) \in \{0,1\}^{|\mathcal{H}|}$ over BETs in G is given by

$$\mathbb{P}(\Omega(\mathcal{H})) = \frac{1}{Z}\exp\Big[-\sum_{j=1}^{|\mathcal{C}|}\theta_j\psi_j(\mathcal{C}_j)\Big] \quad (3)$$

where Z is the normalizing constant and ψ_j corresponds to potential function acting over BETs $h \in \mathrm{Dom}(\mathcal{C}_j)$. Final assignment of $\Omega(\mathcal{H})_{PSL} \in \{1,0\}^{|\mathcal{H}|}$ labels is obtained by solving the *maximum a-posteriori (MAP)* optimization problem

$$\Omega(\mathcal{H})_{PSL} = \underset{\Omega(\mathcal{H})}{\arg\max}\ \mathbb{P}\Big(\Omega(\mathcal{H})\Big)$$

We denote by $M_{PSL}(h,\gamma) \in [0,1]$ the PSL-estimated score for label $\gamma \in \{1,0\}$ on BET h in the optimization above.

Inferable Set using PSL: We define estimated label for each BET h as shown below.

$$l(h) = \begin{cases} 1 \text{ if } M_{PSL}(h,1) \geq \tau \\ 0 \text{ if } M_{PSL}(h,0) \geq \tau \\ \varnothing \text{ otherwise} \end{cases}$$

where threshold τ is system hyper-parameter. Inferable set is composed of BETs for which inference algorithm (PSL) confidently propagates labels.

$$\mathcal{I}(G,\mathcal{Q}) = \{h \in \mathcal{H} \mid l(h) \neq \varnothing\}$$

Note that two BET nodes from ECG can interact with varying strengths through different constraint nodes; this multi-relational structure requires soft probabilistic propagation.

3.4 Control Mechanism

Control mechanism selects the BET to be crowd-evaluated at every iteration. We first present the following two theorems involving KGEval's optimization in Equation (1). Please refer Appendix for proofs of both theorems.

Theorem 1. [Submodularity] *The function optimized by KGEval (Equation (1)) using the PSL-based inference mechanism is submodular (Lovász, 1983).*

The proof follows from the fact that all pairs of BETs satisfy the regularity condition (Jegelka and Bilmes, 2011; Kolmogorov and Zabih, 2004), further used by a proven conjecture (Mossel and Roch, 2007).

Theorem 2. [NP-Hardness] *The problem of selecting optimal solution in KGEval's optimization (Equation (1)) is NP-Hard.*

Proof follows by reducing NP-complete Set-cover Problem (SCP) to selecting $\mathcal{Q}$ which covers $\mathcal{I}(G,\mathcal{Q})$.

Justification for Greedy Strategy: From Theorem 1 and 2, we observe that the function optimized by KGEval is NP-hard and submodular.

Algorithm 1 KGEval: Accuracy Estimation of Knowledge Graphs

```
Require: $\mathcal{H}$: BETs, $\mathcal{C}$: coupling constraints, $\mathbb{B}$:
    assigned budget, $\mathcal{S}$: seed set, $c(h)$: BET cost
 1: $G = \textsc{BuildECG}(\mathcal{H}, \mathcal{C})$
 2: $B_r = \mathbb{B}$
 3: $\mathcal{Q}_0 = \mathcal{S}, t = 1$
 4: repeat
 5:     $h^* = \arg\max_{h \in \mathcal{H}-\mathcal{Q}} |\mathcal{I}(G, \mathcal{Q}_{t-1} \cup \{h\})|$
 6:     $\textsc{CrowdEvaluate}(h^*)$
 7:     $\textsc{RunInference}(\mathcal{Q}_{t-1} \cup h^*)$
 8:     $\mathcal{Q}_t = \mathcal{I}(G, \mathcal{Q}_{t-1} \cup \{h^*\})$
 9:     $B_r = B_r - c(h^*)$
10:     $\mathcal{Q} = \mathcal{Q} \cup \mathcal{Q}_t$
11:     if $\mathcal{Q} \equiv \mathcal{H}$ then
12:         EXIT
13:     end if
14:     $\mathrm{Acc_t} = \frac{1}{|\mathcal{Q}|} \sum_{h \in \mathcal{Q}} l(h)$
15:     $t = t + 1$
16: until CONVERGENCE
17: return $\mathrm{Acc_t}$
```

Results from (Nemhauser et al., 1978) prove that greedy hill-climbing algorithms solve such maximization problem within an approximation factor of $(1-1/e) \approx 63\%$ of the optimal solution. Hence we iteratively select the next BET which gives the greatest increase in size of inferable set.

We acknowledge the importance of crowdsourcing aspects such as label-aggregation, worker's quality estimation etc. Appendix A.1 presents a mechanism to handle noisy crowd workers under limited budget.

3.5 Bringing it All Together

Algorithm 1 presents KGEval. In Lines 1-3, we build the Evaluation Coupling Graph $G = (\mathcal{H} \cup \mathcal{C}, \mathcal{E})$ and use the labels of seed set $\mathcal{S}$ to initialize G. In lines 4-16, we repetitively run our inference mechanism, until either the accuracy estimates have converged, or all the BETs are covered. In each iteration, the BET with the largest inferable set is identified and evaluated using crowdsourcing (Lines 5-6). The new inferable set $\mathcal{Q}_t$ is estimated. These automatically annotated nodes are added to $\mathcal{Q}$ (Lines 7-10).

Convergence: In this paper, we define convergence whenever the variance of sequence of accuracy estimates [$\mathrm{Acc_{t-k}}, \ldots, \mathrm{Acc_{t-1}}, \mathrm{Acc_t}$] is less than α. We set $k = 9$ and $\alpha = 0.002$ for our experiments.

4 Experiments

To assess the effectiveness of KGEval, we ask the following questions: (1).How effective is KGEval in estimating KG accuracy, both at predicate-level and at overall KG-level? (Section 4.3). (2). What is the importance of coupling constraints on its performance? (Section 4.4). (3). And lastly, how robust is KGEval to estimating accuracy of KGs with varying quality? (Section 4.5).

4.1 Model Description

Evaluation set	$\mathcal{H}_N$	$\mathcal{H}_Y$
#BETs	1860	1386
#Constraints	$\lvert\mathcal{C}_N\rvert = 130$	$\lvert\mathcal{C}_Y\rvert = 28$
#Predicates	18	16
Gold Acc.	91.34%	99.20%

Table 2: Details of BET subsets used for accuracy evaluation. (Section 4.1.2).

4.1.1 Setup

Datasets: For experiments, we consider two KGs: NELL and Yago2. From NELL (NELL), we choose a sub-graph of sports related beliefs **NELL-sports**, mostly pertaining to athletes, coaches, teams, leagues, stadiums etc. We construct coupling constraints set $\mathcal{C}_\mathbf{N}$ using top-ranked PRA inference rules for available predicate-relations (Lao et al., 2011). The confidence score returned by PRA are used as weights θ_i. We use NELL-ontology's predicate-signatures to get information for *type* constraints. Please note that PSL is capable of handling weighted constraints and also learn their weights (relative importance). So, it is not critical to provide absolutely correct constraints. We also select **YAGO2-sample** (YAGO) , which unlike NELL-sports, is not domain specific. We use AMIE horn clauses (Galárraga et al., 2013) to construct multi-relational coupling constraints $\mathcal{C}_\mathbf{Y}$. For each $\mathcal{C}_i$, the score returned by AMIE is used as rule weight θ_i. Table 2 reports the statistics of datasets used, their true accuracy and number of coupling constraints. Obtaining gold-labels for millions of facts is non-trivial and expensive as crowdsourcing over full KG incurs significant cost.

Size of evaluation set: NELL-sport consists of $23,422$ beliefs with $13,290$ unique entities and 53 unique predicates. Whereas YAGO-sample has $31,720$ beliefs, with unique $32,103$ entities and 17 predicates. In order to calculate accuracy, we require gold evaluation of all beliefs in the evaluation set. Since obtaining gold evaluation of the entire (or large subsets of) NELL and Yago2 KGs will be prohibitively expensive, we take subset of these KGs for evaluation. (KGEval) consists of datasets used, their crowdsourced labels, coupling constraints and code for inference/control.

Initialization: Algorithm 1 requires initial seed set $\mathcal{S}$ which we generate by randomly evaluating $|\mathcal{S}| = 50$ beliefs from $\mathcal{H}$. To maintain fairness, all baselines start from $\mathcal{S}$. For asserting true (or false) value for beliefs, we set a high soft label confidence threshold at $\tau = 0.8$ (see Section 3.3).

4.1.2 Crowdsourcing of BETs

To compare KGEval predictions against human evaluations, we evaluate all BETs $\{\mathcal{H}_N \cup \mathcal{H}_Y\}$ on AMT. For the ease of workers, we translate each *entity-relation-entity* belief into human readable format before posting to crowd.

We published BETs on AMT under 'classification project' category. We hired AMT recognized master workers for high quality labels and paid \$0.01 per BET. To compare between 'master' and 'noisy' workers, we correlated their labels individually to expert labels on random subset and observed that master workers were better correlated (93%) as compared to three non-masters (89%). Consequently we consider votes of master workers for $\{\mathcal{H}_N \cup \mathcal{H}_Y\}$ as gold labels, which we would like our inference algorithm to be able to predict. As the average turnaround time for AMT tasks runs into a few minutes (Dupuis et al., 2013), KGEval is effectively real-time within such turnaround time range.

4.1.3 Performance Evaluation Metrics

Performance of various methods are evaluated using the following two metrics. To capture accuracy at the predicate level, we define $\Delta_{predicate}$ as the average of difference between gold and estimated accuracy of each of the R predicates in KG.

$$\Delta_{predicate} = \frac{1}{|R|}\left(\sum_{\forall r \in R}\left|\Phi(\mathcal{H}_r) - \frac{1}{|\mathcal{H}_r|}\sum_{\forall h \in \mathcal{H}_r} l(h)\right|\right)$$

We define $\Delta_{overall}$ as the difference between gold and estimated accuracy over the entire evaluation set.

$$\Delta_{overall} = \left|\Phi(\mathcal{H}) - \frac{1}{|\mathcal{H}|}\sum_{\forall h \in \mathcal{H}} l(h)\right|$$

Above, $\Phi(\mathcal{H})$ is the overall gold accuracy, $\Phi(\mathcal{H}_r)$ is the gold accuracy of predicate r and $l(h)$ is the label assigned by the currently evaluated method. $\Delta_{overall}$ treats entire KG as a single bag of BETs whereas $\Delta_{predicate}$ segregates beliefs based on their type of predicate-relation. For both metrics, lower is better.

4.2 Baseline Methods

Since accuracy estimation of large multi-relational KGs is a relatively unexplored problem, there are no well established baselines for this task (apart from random sampling). We present below the baselines which we compared against KGEval.

Random: Randomly sample a BET $h \in \mathcal{H}$ without replacement and crowdsource for its correctness. Selection of every subsequent BET is independent of previous selections.

Max-Degree: Sort the BETs in decreasing order of their degrees in ECG and select them from top for evaluation; this method favors selection of more centrally connected BETs first.

Independent Cascade: This method is based on contagion transmission model where nodes only infect their immediate neighbors (Kempe et al., 2003). At every time iteration t, we choose a BET which is not evaluated yet, crowdsource for its label and let it propagate its evaluation label in ECG.

KGEval: Method proposed in Algorithm 1.

4.3 Effectiveness of KGEval

Experimental results of all methods comparing $\Delta_{overall}$ and $\Delta_{predicate}$ at convergence, are presented in Table 3. We observe that KGEval is able to achieve the best estimate across both datasets and metrics. Due to the significant positive bias in $\mathcal{H}_Y$ (see Table 2), all methods do fairly well as per $\Delta_{overall}$ on this dataset, even though KGEval still outperforms others. Also, KGEval is able to estimate KG accuracy most closely while utilizing least number of crowd-evaluated queries. This clearly demonstrates KGEval's effectiveness.

Nodes in coupling graph with higher degrees are those which participate in large number of constraints. In real KGs, such facts tend to be correct as they interact with several other facts. Hence, MaxDegree overestimates the accuracy by propagating True label. In contrast, Random samples True and False labels in unbiased fashion.

Predicate-level Analysis: Here, we consider the top two baselines from Table 3, viz., Random and

NELL sports dataset ($\mathcal{H}_N$)			
Method	$\Delta_{predicate}$ (%)	$\Delta_{overall}$ (%)	# Queries
Random	4.9	1.3	623
Max-Deg	7.7	2.9	1370
Ind-Casc	9.8	0.8	232
KGEval	**3.6**	**0.5**	**140**
Yago dataset ($\mathcal{H}_Y$)			
Random	1.3	0.3	513
Max-Deg	1.7	0.5	550
Ind-Casc	1.1	0.7	649
KGEval	**0.7**	**0.1**	**204**

Table 3: $\Delta_{predicate}(\%)$ and $\Delta_{overall}(\%)$ estimates (lower is better) of various methods with number of crowd-evaluated queries (BET evaluations) to reach the $\Delta_{overall}$ converged estimate. (See Section 4.3)

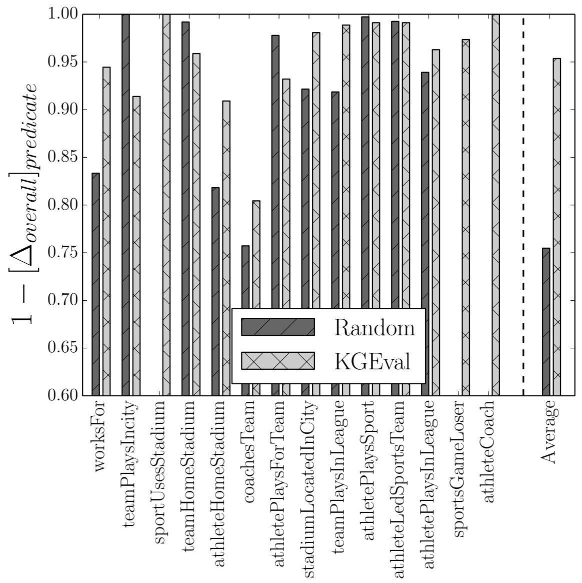

Figure 4: Comparing $(1 - [\Delta_{overall}]_{predicate})$ of individual predicates (higher is better) in $\mathcal{H}_N$ between KGEval and Random, the two top performing systems in Table 3. (see Section 4.3)

KGEval, and compare performance on the $\mathcal{H}_N$ dataset. We use $(1 - [\Delta_{overall}]_{predicate})$ as the metric, which means $\Delta_{overall}$ computed over individual predicates. Here, we are interested in evaluating how well the two methods have estimated per-predicate accuracy when KGEval's $\Delta_{overall}$ has converged. Comparison of per-predicate performances of the two methods is shown in Figure 4. Observe that KGEval significantly outperforms Random baseline. Its advantage lies in exploiting the coupling constraints among beliefs, where evaluating a belief from certain predicate helps infer beliefs from other predicates.

Constraint Set	Iterations to Convergence	$\Delta_{overall}(\%)$
$\mathcal{C}$	**140**	**0.5**
$\mathcal{C} - \mathcal{C}_{b3}$	259	0.9
$\mathcal{C} - \mathcal{C}_{b3} - \mathcal{C}_{b2}$	335	1.1

Table 4: Performance of KGEval with ablated constraint sets. Additional constraints help in better estimation with lesser iterations.(see Section 4.4)

4.4 Importance of Coupling Constraints

This paper is largely motivated by the thesis – *exploiting richer relational couplings among BETs may result in faster and more accurate evaluations*. To evaluate this thesis, we successively ablated Horn clause coupling constraints of body-length 2 and 3 from $\mathcal{C}_N$.

We observe that with the full (non-ablated) constraint set $\mathcal{C}_N$, KGEval takes least number of crowd evaluations of BETs to convergence, while providing best accuracy estimate. Whereas with ablated constraint sets, KGEval takes up to 2.4x more crowd-evaluation queries for convergence; thus validating our thesis.

4.5 Additional Experiments

Other Baselines along with Inference: In order to evaluate how Random and Max-degree perform in conjunction with inference mechanism, we replaced KGEval's greedy control mechanism in Line 5 of Algorithm 1 with these two control mechanisms. In our experiments, we observed that both *Random+inference* and *Max-degree+inference* are able to estimate accuracy more accurately than their control-only variants. Secondly, even though the accuracies estimated by *Random+inference* and *Max-degree+inference* were comparable to that of KGEval, they required larger number of crowd-evaluation queries – 1.2x and 1.35x more, respectively. This shows effectiveness of greedy mechanism.

Rate of Coverage: In case of large KGs with scarce budget, it is imperative to have a mechanism which covers greater parts of KG with lesser number of crowdsource queries. Figure 5 shows the fraction of total beliefs whose evaluations were automatically inferred by different methods as a function of number of crowd-evaluated beliefs. We observe that KGEval infers evaluation for the largest number of BETs at each supervision level.

Robustness to Noise: In order to test robustness of the methods in estimating accuracies

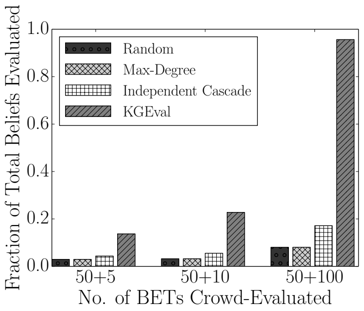

Figure 5: Fraction of total beliefs whose evaluation where automatically inferred by different methods for varying number of crowd-evaluated queries (x-axis) in $\mathcal{H}_N$.

NELL sports + 5% noise ($\mathcal{H}_{N5}$)		
Method	$\Delta_{overall}$ (%)	# Queries
Random	1.8	563
Max-Degree	4.2	1249
Ind-Cascade	1.2	370
KGEval	**0.8**	**106**
NELL sports + 10% noise ($\mathcal{H}_{N10}$)		
Random	1.8	728
Max-Degree	6.2	1501
Ind-Cascade	1.2	406
KGEval	**0.2**	**115**

Table 5: Accuracy estimate (higher is better) over entire KG by various baselines in the presence of noise.

of KGs with different gold accuracies, we artificially added noise to $\mathcal{H}_N$ by flipping a fixed fraction of edges, otherwise following the same evaluation procedure as in Section 3.5. We analyze $\Delta_{overall}$ (and not $\Delta_{predicate}$) because flipping edges in KG distorts predicate-relations dependencies and present in Table 5. We evaluated all the methods and observed that while performance of other methods degraded significantly with diminishing KG quality (more noise), KGEval was significantly robust to noise.

Scalability comparisons with MLN: Markov Logic Networks (Richardson and Domingos, 2006) can serve as a candidate for Inference Mechanism. We compared the runtime performance of KGEval with PSL and MLN as inference engines. While PSL took 320 seconds to complete one iteration, the MLN implementation (PyMLN) could not finish grounding the rules even after 7 hours. This justifies our choice of PSL as the inference engine for KGEval.

5 Related Work

Even though Knowledge Graph (KG) construction is an active area of research, we are not aware of any previous research which systematically studies the important problem of estimating accuracy of such automatically constructed KGs. Random sampling has traditionally been the most preferred way for large-scale KG accuracy estimation (Mitchell et al., 2015).

Traditional crowdsourcing research has typically considered atomic allocation of tasks where the requester posts them *independently*. KGEval operates in a rather novel crowdsourcing setting as it exploits dependencies among its tasks (BETs or belief evaluations). Our notion of interdependence (coupling constraints) among tasks is more general and different than related ideas explored in the crowdsourcing literature before (Kolobov et al., 2013; Bragg et al., 2013; Sun et al., 2015). Even though coupling constraints have been used for KG construction (Nakashole et al., 2011; Galárraga et al., 2013; Mitchell et al., 2015), they have so far not been exploited for KG evaluation. We address this gap in this paper.

The task of knowledge corroboration (Kasneci et al., 2010) proposes probabilistic model to utilize a fixed set of basic first-order logic rules for label propagation and is closely aligned with our motivations. However, unlike KGEval, it does not try to reduce the number of queries to crowdsource or maximize coverage.

6 Conclusion

In this paper, we have initiated a systematic study into the important problem of evaluation of automatically constructed Knowledge Graphs. In order to address this challenge, we have proposed KGEval, an instance of a novel crowdsourcing paradigm where dependencies among tasks presented to humans (BETs) are exploited. To the best of our knowledge, this is the first method of its kind. We demonstrated that the objective optimized by KGEval is in fact NP-Hard and submodular, and hence allows for the application of simple greedy algorithms with guarantees. Through extensive experiments on real datasets, we demonstrated effectiveness of KGEval. We hope to extend KGEval to incorporate varying evaluation cost, and also explore more sophisticated evaluation aggregation.

Acknowledgments

This research has been supported by MHRD (Govt. of India) and in part by a gift from Google.

A Appendix

Submodularity: A real valued function f, which acts over subsets of any finite set $\mathcal{H}$, is said to be *submodular* if $\forall R, S \subset \mathcal{H}$ it fulfills

$$f(R) + f(S) \geq f(R \cup S) + f(R \cap S).$$

We call potential function ψ as pairwise *regular* if for all pairs of BETs $\{p, q\} \in \mathcal{H}$ it satisfies

Proof. **(for Theorem 1)** The additional utility, in terms of label inference, obtained by adding a BET to larger set is lesser than adding it to any smaller subset. By construction, any two BETs which share a common factor node $\mathcal{C}_j$ are encouraged to have similar labels in G.

Potential functions ψ_j of Equation (3) satisfy pairwise regularity property i.e., for all BETs $\{p, q\} \in \mathcal{H}$

$$\psi(0,1) + \psi(1,0) \geq \psi(0,0) + \psi(1,1) \tag{4}$$

where $\{1, 0\}$ represent true/false. Equivalence of *submodular* and *regular* properties are established (Kolmogorov and Zabih, 2004; Jegelka and Bilmes, 2011). Using non-negative summation property (Lovász, 1983), $\sum_{j \in C} \theta_j \psi_j$ is submodular for positive weights $\theta_j \geq 0$.

We consider a BET h to be confidently inferred when the soft score of its label assignment in $\mathcal{I}(G, \mathcal{Q})$ is greater than threshold $\tau_h \in [0, 1]$. From above we know that $\mathbb{P}(l(h)|\mathcal{Q})$ is submodular with respect to fixed initial set $\mathcal{Q}$. Although $\max$ or $\min$ of submodular functions are not submodular in general, but (Kempe et al., 2003) conjectured that global function of Equation (1) is submodular if local threshold function $\mathbb{P}(h|\mathcal{Q}) \geq \tau_h$ respected submodularity, which holds good in our case of Equation (3). This conjecture was further proved in (Mossel and Roch, 2007) and thus making our global optimization function of Equation (1) submodular. □

Proof. **(for Theorem 2)** We reduce KGEval to NP-complete Set-cover Problem (SCP) so as to select $\mathcal{Q}$ which covers $\mathcal{I}(G, \mathcal{Q})$. For the proof to remain consistent with earlier notations, we define SCP by collection of subsets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_m$ from set $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ and we want to determine if there exist k subsets whose union equals $\mathcal{H}$. We define a bipartite graph with $m + n$ nodes corresponding to $\mathcal{I}_i$'s and h_j's respectively and construct edge $(\mathcal{I}_i, h_j)$ if $h_j \in \mathcal{I}_i$. We need to find a set $\mathcal{Q}$, with cardinality k, such that $|\mathcal{I}(G, \mathcal{Q})| \geq n + k$.

Choosing our BET-set $\mathcal{Q}$ from SCP solution and further inferring evaluations of other remaining BETs using PSL will solve the problem in hand. □

A.1 Noisy Crowd Workers and Budget

Here, we provide a scheme to allot crowd workers so as to remain within specified budget and upper bound total error on accuracy estimate. We have not integrated this mechanism with Algorithm 1 to maintain its simplicity.

We resort to majority voting in our analysis and assume that crowd workers are not adversarial. So expectation over responses $r_h(u)$ for a task h with respect to multiple workers u is close to its true label $t(h)$ (Tran-Thanh et al., 2013), i.e.,

$$\left| \mathbb{E}_{u \sim \mathbb{D}(u,h)}[r_h(u)] - t(h) \right| \leq \frac{1}{2} \tag{5}$$

where $\mathbb{D}$ is joint probability distribution of workers u and tasks h.

Our key idea is that we want to be more confident about BETs h with larger inferable set (as they impact larger parts of KG) and hence allocate them more budget to post to more workers. We determine the number of workers $\{w_{h_1}, \ldots, w_{h_n}\}$ for each task such that h_t with larger inference set have higher w_{h_t}. For total budget B, we allocate

$$w_{h_t} = \left\lfloor \frac{B\, i_t\, (1-\gamma)}{c\, i_{max}} \right\rfloor$$

where i_t denotes the cardinality of inferable set $\mathcal{I}(G, \mathcal{Q} \cup h_t)$, c the cost of querying crowd worker, i_{max} the size of largest inferable set and $\gamma \in [0, 1]$ constant.

This allocation mechanism easily integrates with Algorithm 1; in (Line 8) we determine size of inferable set $i_t = |\mathcal{Q}_t|$ for task h and allocate w_h crowd workers. Budget depletion (Line 9) is modified to $B_r = B_r - w_h c(h)$. The following theorem bounds the error with such allocation scheme.

Theorem 3. [Error bounds] *The allocation scheme of redundantly posing h_t to w_{h_t} workers does not exceed the total budget B and its expected estimation error is upper bounded by $e^{-O(i_t)}$, keeping other parameters fixed. The expected estimation error over all tasks is upper bounded by $e^{-O(B)}$.*

Proof. Let $\gamma \in [0, 1]$ control the reduction in size of inferable set by $i_{t+1} = \gamma\, i_t$. By allocating w_{h_t} redundant workers for task $h_t, \forall t \in \{1 \cdots n\}$ with size of inferable set i_t, we incur total cost of

$$\begin{aligned}
\sum_{t=1}^{n} c\, w_{h_t} &= \sum_{t=1}^{n} \frac{B\, i_t\, (1-\gamma)}{c\, i_{max}} \cdot c \\
&= \left(\sum_{t=1}^{n} i_t \right) \cdot \left(\frac{B\,(1-\gamma)}{i_{max}} \right) \\
&= \left(\frac{i_{max}\,(1-\gamma^T)}{(1-\gamma)} \right) \cdot \left(\frac{B\,(1-\gamma)}{i_{max}} \right) \\
&\leq B
\end{aligned}$$

Note that the above geometric approximation helps in estimating summation $\sum_{t=1}^{n} i_t$ at iteration $t \leq n$.

Error Bounds: Here we show that the expected error of estimating of h_t for any time t decreases exponentially in the size of inferable set i_t. We use majority voting to aggregate w_{h_t} worker responses for h_t, denoted by $\hat{r}_{h_t} \in \{0, 1\}$

$$\hat{r}_{h_t} = \left\lfloor \frac{1}{w_{h_t}} \sum_{k=1}^{w_{h_t}} r_{h_t}(u_k) - \frac{1}{2} \right\rfloor + 1 \tag{6}$$

where $r_{h_t}(u_k)$ is the response by k^{th} worker for h_t. The error from aggregated response can be given by $\Delta(h_t) = |\hat{r}_{h_t} - t(h_t)|$, where $t(h_t)$ is its true label. From Equation (5) and Hoeffding-Azuma bounds over w_{h_t} i.i.d responses and error margin ε_t, we have

$$\begin{aligned}
\Delta(h_t) &= \mathbb{P}\left\{ \left| \frac{1}{w_{h_t}} \sum_{k=1}^{w_{h_t}} r_{h_t}(u_k) - \mathbb{E}(r_h(u)) \right| \geq \varepsilon_t \right\} \\
&= 2 \exp\left(-2\, \frac{B\, i_t\, (1-\gamma)}{c\, i_{max}}\, \varepsilon_t^2 \right)
\end{aligned}$$

For fixed budget B and given error margin ε_t, we have $\Delta(h_t) = e^{-O(i_t)}$. Summing up over all tasks t, by union bounds we get the total expected error from absolute truth as $\Delta(B) = \sum_{t=1}^{n} \Delta(h_t)$.

$$\begin{aligned}
\Delta(B) &\leq \sum_{t=1}^{n} 2 \exp\left(-2\, \frac{B\, i_t\, (1-\gamma)}{c\, i_{max}}\, \varepsilon_t^2 \right) \\
&\leq n \cdot 2 \exp\left(-2\, \frac{B\, i_{min}\, (1-\gamma)}{c\, i_{max}}\, \varepsilon_{min}^2 \right)
\end{aligned}$$

The accuracy estimation error will decay exponentially with increase in total budget for fixed parameters. □

References

AMT. https://www.mturk.com.

Jonathan Bragg, Daniel S Weld, et al. 2013. Crowdsourcing multi-label classification for taxonomy creation. In *HCOMP*.

Matthias Broecheler, Lilyana Mihalkova, and Lise Getoor. 2010. Probabilistic similarity logic. In *UAI*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, pages 601–610.

Marc Dupuis, Barbara Endicott-Popovsky, and Robert Crossler. 2013. An analysis of the use of amazons mechanical turk for survey research in the cloud. In *ICCSM2013-Proceedings of the International Conference on Cloud Security Management: ICCSM 2013*, page 10. Academic Conferences Limited.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422.

Stefanie Jegelka and Jeff Bilmes. 2011. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, pages 1897–1904.

Gjergji Kasneci, Jurgen Van Gael, Ralf Herbrich, and Thore Graepel. 2010. Bayesian knowledge corroboration with logical rules and user feedback. In *Machine Learning and Knowledge Discovery in Databases*, pages 1–18.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *SIGKDD*.

KGEval. http://talukdar.net/mall-lab/KGEval.

Vladimir Kolmogorov and Ramin Zabih. 2004. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159.

Andrey Kolobov, Daniel S Weld, et al. 2013. Joint crowdsourcing of multiple tasks. In *HCOMP*.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. 2001. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, pages 529–539.

László Lovász. 1983. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of AAAI*.

Elchanan Mossel and Sebastien Roch. 2007. On the submodularity of influence in social networks. In *ACM symposium on Theory of computing*, pages 128–134.

Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. 2011. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of WSDM*.

NELL. http://rtw.ml.cmu.edu/rtw/resources.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294.

PSL. http://www.psl.umiacs.umd.edu.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *The Semantic Web–ISWC 2013*, pages 542–557. Springer.

PyMLN. http://ias.cs.tum.edu/people/jain/mlns.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning*, 62(1-2):107–136.

Mehdi Samadi, Partha Talukdar, Manuela Veloso, and Tom Mitchell. 2015. Askworld: budget-sensitive query evaluation for knowledge-on-demand. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 837–843. AAAI Press.

Yuyin Sun, Adish Singla, Dieter Fox, and Andreas Krause. 2015. Building hierarchies of concepts via crowdsourcing. *arXiv preprint arXiv:1504.07302*.

Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. 2013. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *AAMAS*.

YAGO. https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga.