Ontology-Aware Partitioning for Knowledge Graph Identification

Jay Pujara, Hui Miao, Lise Getoor
Dept of Computer Science
University of Maryland
College Park, MD 20742
{jay,hui,getoor}@cs.umd.edu

William W. Cohen
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
wcohen@cs.cmu.edu

ABSTRACT

Knowledge graphs provide a powerful representation of entities and the relationships between them, but automatically constructing such graphs from noisy extractions presents numerous challenges. Knowledge graph identification (KGI) is a technique for knowledge graph construction that jointly reasons about entities, attributes and relations in the presence of uncertain inputs and ontological constraints. Although knowledge graph identification shows promise scaling to knowledge graphs built from millions of extractions, increasingly powerful extraction engines may soon require knowledge graphs built from billions of extractions. One tool for scaling is partitioning extractions to allow reasoning to occur in parallel. We explore approaches which leverage ontological information and distributional information in partitioning. We compare these techniques with hash-based approaches, and show that using a richer partitioning model that incorporates the ontology graph and distribution of extractions provides superior results. Our results demonstrate that partitioning can result in order-of-magnitude speedups without reducing model performance.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence

Keywords

knowledge graphs identification; partitioning probabilistic graphical models

1. INTRODUCTION

The web is a vast repository of knowledge, and copious research has sought to extract structured data at web-scale [4, 5, 13] and populate knowledge bases using this abundance of data [6, 1]. As these techniques grow more sophisticated, increasing attention is being devoted to considering the relationships between these structured extractions, and representing both the extractions and relationships as a *knowledge graph* [16]. In conjunction with the knowledge graph representation, current research [7, 14] jointly reasons about millions of web-scale extractions, while leveraging ontological information, to produce a consistent knowledge graph.

Jointly reasoning about noisy extractions from the web has shown promise for uncovering the true knowledge graph, but whether such approaches will successfully scale to billions of extractions remains an open question. We analyze the scalability of knowledge graph identification (KGI) [14], a method which operates on uncertain extractions and performs entity resolution, collective classification, and link prediction mediated by ontological constraints to produce a knowledge graph. While knowledge graph identification is capable of inferring a complete knowledge graph using millions of extractions in a few hours, we explore mechanisms for distributing the computation of the knowledge graph across machines to perform KGI in minutes, while preserving high precision and recall.

In this work, we provide an overview of knowledge graph identification implemented using probabilistic soft logic (PSL). We build on this framework, exploring methods for partitioning extractions while retaining the benefits of joint reasoning. In experiments, we contrast a hash-based approach that partitions the output of an information extraction system directly to more sophisticated techniques that leverage the ontology and distribution of extractions. We introduce a novel partitioning mechanism that balances the contribution of different types of ontological information and the frequency of different relations and labels in the data. Our contribution is an ontology-aware approach to partitioning extractions that reduces the time necessary for knowledge graph identification dramatically with no significant impact on model performance.

AKBC'13, Oct. 27–28, 2013, San Francisco, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2411-3/13/10 ...$15.00.
http://dx.doi.org/10.1145/2509558.2509562.

2. KNOWLEDGE GRAPH IDENTIFICATION

Candidate facts from an information extraction system can be represented as an *extraction graph* where entities are nodes, categories are labels associated with each node, and relations are directed edges between the nodes. Unfortunately, the output from an information extraction system is often incorrect; the graph constructed from it has spurious and missing nodes and edges, and missing or inaccurate node labels. Our approach, *knowledge graph identification* (KGI) [14], performs collective classification, link prediction, and entity resolution in the presence of rich ontological information and multiple sources of uncertain information, ultimately producing a better knowledge graph.

Unlike earlier work on graph identification [12] we use a very different probabilistic framework, PSL [3, 9]. PSL allows us to apply global constraints instead of relying only on local features. PSL models are expressed using a set of universally-quantified logical rules that, when combined with data such as the noisy extractions and ontological information, define a probability distribution over possible knowledge graphs. In the case of KGI, we introduce rules to relate uncertain extractions to the true relations and labels in the knowledge graph, pool these facts across co-referent entities, and constrain relations and labels with rules that use ontological information such as domain and range constraints, mutual exclusion relationships. We explain how these components of KGI map to PSL rules, motivating these rules with examples of challenges found in a real-world information extraction system, the Never-Ending Language Learner (NELL) [4].

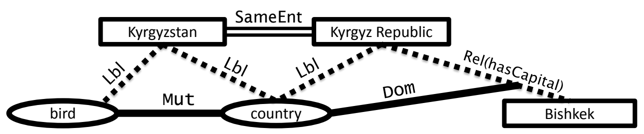

Figure 1: An illustration of the example showing how knowledge graph identification can resolve conflicting information in a knowledge graph. Entities are shown in rectangles, dotted lines represent uncertain information, solid lines show ontological constraints and double lines represent co-referent entities found with entity resolution.

2.1 Representation of Uncertain Extractions

NELL produces candidate extractions from a web corpus which often contains noise. Candidate extractions from the NELL corpus include labels such as bird(kyrgyzstan) and country(kyrghyzstan) as well as relations such as locatedIn(kyrghyzstan, Russia), locatedIn(kyrgz republic, Asia), locatedIn(kyrghyzstan, US), and locatedIn(kyrgyzstan, Kazakhstan). These extractions can contain mistakes that include variations in spelling and inconsistencies in relationships, and NELL assigns confidence values to each extraction.

In PSL, we represent these candidate extractions with predicates CANDLBL and CANDREL , e.g. CANDLBL(kyrgyzstan, bird) and CANDREL(kyrgz republic, Asia, locatedIn). NELL has multiple extractors that generate candidates, and we can use different predicates for each extractor to capture the confidence of that extractor. For a given extractor T, we introduce predicates CANDREL_T and CANDLBL_T to represent the candidates extracted by T. We relate these candidates to the facts that we wish to infer, LBL and REL, using the following rules:

$$\text{CANDREL}_T(E_1, E_2, R) \overset{w_{CR-T}}{\Rightarrow} \text{REL}(E_1, E_2, R)$$

$$\text{CANDLBL}_T(E, L) \overset{w_{CL-T}}{\Rightarrow} \text{LBL}(E, L)$$

PSL uses *soft logic*, so we can represent noisy extractions by translating confidences into real-valued truth assignments in the $[0, 1]$ range. For example, if NELL extracts the relation locatedIn(kyrgz republic,Asia) and assigns it a confidence value of 0.9, we would assign the atom CANDREL(kyrgyzstan,Asia,locatedIn) a soft-truth value of 0.9. Similarly, our output values for unknown facts are in the $[0, 1]$ range allow us to trade-off precision and recall by using a truth threshold. By learning the weights of these rules, w_{CL_T} and w_{CR_T}, our model combines multiple sources of information to label nodes and predict links.

2.2 Reasoning About Co-Referent Entities

While the previous PSL rules provide the building blocks of predicting links and labels using uncertain information, KGI employs entity resolution to pool information across co-referent entities. In the example above, many different forms for the country Kyrgystan appear: kyrgyzstan , kyrghyzstan , and kyrgz republic. A key component of KGI is identifying possibly co-referent entities and determining the similarity of these entities. We use the SAMEENT predicate to capture the similarity of two entities. While any similarity metric can be used, we compute the similarity of entities using a process of mapping each entity to the YAGO knowledge base [17], extracting a set of Wikipedia articles for each entity and then computing the Jaccard index of possibly co-referent entities. We incorporate this information about co-referent entities by constraining the labels and relations of these entities through PSL rules:

$$\text{SAMEENT}(E_1, E_2) \wedge \text{LBL}(E_1, L) \overset{w_{EL}}{\Rightarrow} \text{LBL}(E_2, L)$$

$$\text{SAMEENT}(E_1, E_2) \wedge \text{REL}(E_1, E, R) \overset{w_{ER}}{\Rightarrow} \text{REL}(E_2, E, R)$$

$$\text{SAMEENT}(E_1, E_2) \wedge \text{REL}(E, E_1, R) \overset{w_{ER}}{\Rightarrow} \text{REL}(E, E_2, R)$$

These rules define an equivalence class of entities, such that all entities related by the SAMEENT predicate must have the same labels and relations. The soft-truth value of the SAMEENT, derived from our similarity function, mediates the strength of these rules. When two entities are very similar, they will have a high truth value for SAMEENT, so any label assigned to the first entity will also be assigned to the second entity. On the other hand, if the similarity score for two entities is low, the truth values of their respective labels and relations will not be strongly constrained.

2.3 Incorporating Ontological Information

Although entity resolution allows us to relate extractions that refer to the same entity, knowledge graphs can employ ontological information to specify rich relationships between many facts. Our ontological constraints are based on the logical formulation proposed in [7]. Each type of ontological relation is represented as a predicate, and these predicates represent ontological knowledge of the relationships between labels and relations. For example, the constraints DOM(hasCapital, country) and RNG(hasCapital, city) specify that the relation hasCapital is a mapping from entities with label country to entities with label city. The constraint MUT(country, bird) specifies that the labels country and bird are mutually exclusive, so that an entity cannot have both the labels country and bird. We similarly use constraints for subsumption of labels (SUB) and inversely-related functions (INV). To use this ontological knowledge, we introduce rules relating each ontological relation to the predicates representing our knowledge graph. We specify seven types of ontological constraints in our experiments:

$$\begin{array}{llll}
\text{DOM}(R, L) & \wedge\ \text{REL}(E_1, E_2, R) & \overset{w_O}{\Rightarrow} & \text{LBL}(E_1, L) \\
\text{RNG}(R, L) & \wedge\ \text{REL}(E_1, E_2, R) & \overset{w_O}{\Rightarrow} & \text{LBL}(E_2, L) \\
\text{INV}(R, S) & \wedge\ \text{REL}(E_1, E_2, R) & \overset{w_O}{\Rightarrow} & \text{REL}(E_2, E_1, S) \\
\text{SUB}(L, P) & \wedge\ \text{LBL}(E, L) & \overset{w_O}{\Rightarrow} & \text{LBL}(E, P) \\
\text{RSUB}(R, S) & \wedge\ \text{REL}(E_1, E_2, R) & \overset{w_O}{\Rightarrow} & \text{REL}(E_1, E_2, S) \\
\text{MUT}(L_1, L_2) & \wedge\ \text{LBL}(E, L_1) & \overset{w_O}{\Rightarrow} & \neg\text{LBL}(E, L_2) \\
\text{RMUT}(R, S) & \wedge\ \text{REL}(E_1, E_2, R) & \overset{w_O}{\Rightarrow} & \neg\text{REL}(E_1, E_2, S)
\end{array}$$

2.4 Putting It All Together

Inferring a knowledge graph becomes challenging as we consider the many interactions between the uncertain extractions that we encounter in KGI (Figure 1). For example, NELL's ontology includes the constraint that the attributes bird and country are mutually exclusive. While extractor confidences may not be able to resolve which of these two labels is more likely to apply to kyrgyzstan, reasoning collectively using entity resolution and ontological constraints can provide a solution. For example, NELL is highly confident that kyrgz republic has a capital city, Bishkek. The NELL ontology specifies that the domain of the relation hasCapital has label country. Entity resolution allows us to infer that kyrgz republic refers to the same entity as kyrgyzstan. Deciding whether Kyrgyzstan is a bird or a country now involves a prediction where we include the confidence values of the corresponding bird and country facts from co-referent entities, as well as collective features from ontological constraints of these co-referent entities, such as the confidence values of the hasCapital relations.

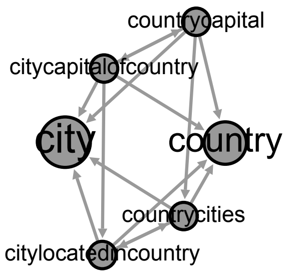

Figure 2: A small subset of the ontology graph. Generating clusters in the ontology graph allows us to partition the extractions effectively.

To accomplish this collective reasoning, we use PSL to define a joint probability distribution over knowledge graphs. The universally-quantified rules described are a PSL model and provide the basis for defining this probability distribution. In a PSL program, $\mathbf{\Pi}$, this model is *grounded* by substituting values from NELL's noisy extractions into the rule template. For example, the rule:

$$\text{DOM}(R, L) \quad \wedge \ \text{REL}(E_1, E_2, R) \quad \overset{w_O}{\Rightarrow} \ \text{LBL}(E_1, L)$$

can be grounded by substituting atoms from NELL, DOM(`hasCapital`, `country`), REL(`kyrgyzstan`, `Bishkek`, `hasCapital`), and LBL(`kyrgyzstan`, `country`), into the rule template. We refer to the collection of ground rules in the program as R.

Unlike Boolean logic where each grounding would have a binary truth value, our choice of soft logic requires a different definition of truth value. We refer to an assignment of soft-truth values to atoms as an interpretation, which, in this case, corresponds to a possible knowledge graph, and use the *Lukasiewicz t-norm* and *co-norm* to determine the truth values of logical formulas under an interpretation. This t-norm defines a relaxation of logical connectives as follows:

$$p \wedge q = \max(0, p + q - 1)$$
$$p \vee q = \min(1, p + q)$$
$$\neg p = 1 - p$$

With this definition, for each possible knowledge graph, G, we can assign a truth value $T_r(G)$ to every grounding $r \in R$ and define a *distance to satisfaction*, $\phi_r(G) = 1 - T_r(G)$ for each grounding. The probability distribution over knowledge graphs, $P_{\mathbf{\Pi}}(G)$ can now be defined using a weighted combination of the distance to satisfaction of ground rules in the PSL program:

$$P_{\mathbf{\Pi}}(G) = \frac{1}{Z} \exp \left[-\sum_{r \in R} w_r \phi_r(G)^p \right]$$

where $p \in 1, 2$ specifies a linear or quadratic combination and Z is a normalization constant.

Most probable explanation (MPE) inference corresponds to identifying a graph that maximizes $P_{\mathbf{\Pi}}(G)$, which can then be mapped to a set of labels and relations that comprises the true knowledge graph. In our work, we choose a soft-truth threshold and determine the true entities, labels and relations by using those atoms whose truth value exceeds the threshold. MPE inference can be formulated as convex optimization in PSL, and using the alternating direction method of multipliers (ADMM), which scales linearly with the number of ground rules in the PSL program[2].

3. PARTITIONING EXTRACTIONS

Our current approach for knowledge graph identification easily scale to millions of extractions. Yet the combination of more powerful extraction techniques and the vast information available on the Internet suggest that the true scale of KGI is billions of extractions or more. Horizontal partitioning, or splitting data into multiple sets which are processed independently, is non-trivial for joint inference problems such as KGI. Here we consider preliminary approaches for horizontally partitioning the extractions used for KGI and introduce ontology-aware partitioning that maintains the performance of KGI while drastically reducing the running time.

Many appealing strategies for partitioning extractions involve partitioning the extractions directly. For example, generating a min-cut of the graph of noisy extractions would provide a straightforward approach to partitioning KGI. However, a number of issues make such approaches intractable. First, such a partitioning requires partitioning a graph with billions of edges, which presents a substantial scalability challenge. Second, partitioning extractions directly does not preserve the ontological relationships that form a key ingredient for generating a consistent knowledge base.

Instead of partitioning the extractions, our approach partitions the labels and relations in the ontology. The ontology is many orders of magnitude smaller than the extractions and, as demonstrated in the PSL model for KGI, provides many important constraints for joint reasoning. We partition the ontology by representing ontological information as a graph, with labels and relations as vertices and the ontological constraints between them as edges. For example, the ontological relation DOM(`cityCapitalOfCountry`, `country`) would be converted to an edge of type DOM between the relation vertex `cityCapitalOfCountry` and the label vertex `country`. We show a small subset of the ontological graph for NELL in Figure 2.

Given an ontology graph, many graph clustering techniques, such as edge min-cut, can be used to partition the relations and labels present in the ontology. Using these clusters of relations and labels, we can create a corresponding partition of the extractions of specific instances of these relations and labels. One potential problem is that some relations and labels occur more frequently in the data than others. For example, the extractor may have far more extractions about the label `city` than the label `bird`, resulting in partitions that are unbalanced. Unbalanced partitions pose a problem when inference is run in parallel across partitions, as the time of inference will be proportional to the size of the largest partition. By balancing partitions, we can produce the quickest overall inference. We address unbalanced partitions by considering an approach that weights each vertex, corresponding to a relation or label, by its frequency in the extractions and then constraining the graph cut to produce clusters that have equal vertex weights.

Another possible problem with simply partitioning the ontology graph is that ontological information is unbalanced. For example, in NELL's ontology there are nearly 50K RMUT constraints and only 418 DOM constraints. Many of the mutually-exclusive relations are not present in the extractions for the same pair of entities, while domain constraints are relevant for every extracted relation. We consider an approach that weights the edges, which correspond to ontological constraints, by choosing weights that are inversely proportional to the number of ontological constraints of that type. Using this weighting scheme, the aggregate weights of all constraints of each type will be equal, so that DOM and RMUT constraints have the same aggregate weight.

While the choice of partitioning technique can influence running time, the number of partitions used in inference can also impact the computational performance of KGI. Joint inference without partitioning preserves all dependencies between extractions, but has a correspondingly complex model and cannot benefit from parallelism. Using a large number of partitions increases parallelism and improves the speed of inference, but necessarily involves losing dependencies which may reduce the quality of the inference results. We explore the speed-quality tradeoff between the number of partitions and the quality of the inference results.

4. EXPERIMENTAL RESULTS

We evaluate different partitioning strategies for our KGI model with data from iteration 165 of the NELL system, which contains nearly 80K ontological relationships, 1.7M candidate facts, and

Table 1: Comparing different partitioning techniques, we find that partitioning extractions with an ontology-based approach that weights vertices with the frequency of respective labels and relations in the data preserves model quality and reduces inference speed

Technique	AUC	Time (min.)
`NELL` (no KGI)	0.765	-
`baseline`	0.780	**31**
`Onto-EqEdg-NoVtx`	0.788	42
`Onto-EqEdg-WtVtx`	**0.791**	**31**
`Onto-WtEdg-WtVtx`	0.790	31
`No-Partitioning`	0.794	97

440K previously promoted facts which we represent as a separate, noisy source. While PSL supports weight learning, our experiments use a simpler setting where all source weights are set to be equal ($\forall T : w_{CL-T} = w_{CR-T} = 1$) , while entity resolution rules and ontology rules are given higher weights ($w_{ER} = w_{EL} = 10; w_O = 100$). We assess the quality of our inference results using a manually-labeled evaluation set with 4.5K extractions [7] and measuring the running time on a 16-core Xeon X5550 CPU at 2.67GHz with 78GB of RAM. We provide documentation, code and datasets to replicate our results on GitHub[1]. To partition the ontology, we use the METIS graph partitioning package[8]. In all cases, the time for partitioning the ontology graph was less than a minute. Our experiments consider two aspects of partitioning extractions for KGI: the partitioning technique and the number of partitions.

4.1 Partitioning Techniques

We compare four techniques for partitioning knowledge graphs with inference on the full set of extractions. The first, a `baseline`, randomly assigns each extraction to a partition. While such an approach balances partitions, it does not actively try to maintain the dependencies between extractions that KGI uses. The second approach `Onto-EqEdg-NoVtx` formulates an ontology graph where each edge (corresponding to an ontological constraint) has equal weight. The ontology graph is partitioned using a p-way balanced min-cut, where the objective function minimizes the communication cost defined by the sum of adjacent edge weights. The third approach, `Onto-EqEdg-WtVtx` equally weights each edge but assigns weights to each vertex (relation or label) based on the frequency of that relation or label in the extraction data. The ontology graph is partitioned using a minimum edge cut with a constraint that each cluster has the same aggregate vertex weight. The fourth approach, `Onto-WtEdg-WtVtx` weights vertices by frequency, and also assigns a weight to each edge. The edge weights are set to be inversely proportional to the frequency of the respective type of ontological information. This formulation was chosen to give each type of ontological information an equal representation in the ontology graph.

For each partitioning algorithm, we generate 6 disjoint clusters of labels and relations. We use these 6 clusters of labels and relations to produce 6 corresponding partitions from the extraction data, where each partition is limited to the relations and labels in the corresponding cluster. For each of the 6 partitions generated we perform inference on each partition independently and combine the results of this distributed inference, averaging truth values when the same fact appears in the output of multiple partitions. We compute the area under the precision-recall curve (AUC) for each technique, and report the running time of the slowest partition.

As shown in Table 1, inference over the full knowledge graph `No-Partitioning` takes 97 minutes and provides an improvement over NELL's default strategy for promoting candidates. The `baseline` strategy of randomly partitioning extractions dramatically reduces inference time, but produces a considerable drop in the AUC. By using an ontology-aware partitioning method `Onto-EqEdg-NoVtx`, we improve the AUC over the baseline, achieving parity with the full joint inference problem, but the running time increases significantly relative to the baseline. Using vertex weights that reflect the data distribution, `Onto-EqEdg-WtVtx` reduces the inference time and improves AUC. However including weights based on the ontological frequency, `Onto-WtEdg-WtVtx`, does not improve our results.

[1]`https://github.com/linqs/KnowledgeGraphIdentification`

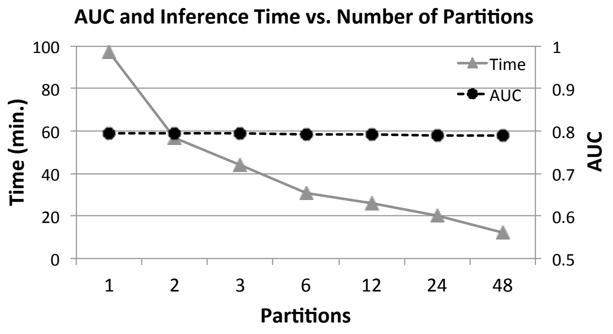

Figure 3: Increasing the number of partitions used in inference can dramatically reduce the time for inference with relatively modest loss in quality, as measured by AUC

4.2 Number of Partitions

The number of partitions used for joint inference can have an impact on the speed and quality of the inference task in KGI. We consider the best-performing partitioning technique from our experiments in the previous subsection, `Onto-EqEdg-WtVtx`, and evaluate performance for 1, 2, 3, 6, 12, 24, and 48 partitions. Figure 3 shows the trade-off between inference speed and quality for the NELL dataset. Our results show that partitioning can dramatically reduce inference time from 97 minutes with a single partition to 12 minutes with 48 partitions. Surprisingly, there is little degradation in inference quality as measured by AUC, which ranges from .794 with a single partition to .788 with 48 partitions. The quality for 48 partitions remains higher than the baseline strategy from the previous section, which had an AUC of .780 from randomly partitioning the data into 6 partitions. We observe a sublinear speedup, which is potentially related to caching effects or computational overhead in the implementation of our model.

5. CONCLUSION

Knowledge graphs present a growing scalability challenge: reasoning collectively about potentially billions of interrelated facts. We consider new methods for scaling *knowledge graph identification*, jointly inferring a knowledge graph from the noisy output of an information extraction system through a combined process of determining co-referent entities, predicting relational links, collectively classifying entity labels, and enforcing ontological constraints. Our work demonstrates that appropriately leveraging the ontology to partition extractions can retain many of the benefits of joint approaches to knowledge graph construction while improving running time considerably. In future work, we look forward to investigating increasingly sophisticated approaches to partitioning extractions for knowledge graph identification, possibly improving results further using multiple partitionings of the same data, as in [15], or implementing KGI on powerful graph computation platforms such as GraphLab [10] or Pregel [11].

Acknowledgments.

This work was partially supported by NSF CAREER grant 0746930 and NSF grants IIS1218488 and CCF0937094. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] J. Artiles and J. Mayfield, editors. *Workshop on Knowledge Base Population*, 2012.

[2] S. H. Bach, M. Broecheler, L. Getoor, and D. P. O'Leary. Scaling MPE Inference for Constrained Continuous Markov Random Fields with Consensus Optimization. In *NIPS*, 2012.

[3] M. Broecheler, L. Mihalkova, and L. Getoor. Probabilistic Similarity Logic. In *UAI*, 2010.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, 2010.

[5] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open Information Extraction from the Web. *Communications of the ACM*, 51(12), 2008.

[6] H. Ji, R. Grishman, and H. Dang. Overview of the Knowledge Base Population Track. In *Text Analysis Conference*, 2011.

[7] S. Jiang, D. Lowd, and D. Dou. Learning to Refine an Automatically Extracted Knowledge Base Using Markov Logic. In *ICDM*, 2012.

[8] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1), 1998.

[9] A. Kimmig, S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. A Short Introduction to Probabilistic Soft Logic. In *NIPS Workshop on Probabilistic Programming*, 2012.

[10] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. GraphLab: A New Parallel Framework for Machine Learning. In *UAI*, 2010.

[11] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *ACM SIGMOD International Conference on Management of data*, 2010.

[12] G. M. Namata, S. Kok, and L. Getoor. Collective Graph Identification. In *KDD*, 2011.

[13] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and Searching the World Wide Web of Facts-Step One: the One-million Fact Extraction Challenge. In *AAAI*, 2006.

[14] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Knowledge graph identification. In *ISWC*, 2013.

[15] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. In *ACL*, 2011.

[16] A. Singhal. Introducing the Knowledge Graph: Things, Not Strings, 2012. Official Blog (of Google).

[17] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW*, 2007.