Graph-Based Wrong IsA Relation Detection in a Large-Scale Lexical Taxonomy

Jiaqing Liang
Shanghai Key Laboratory of Data Science,
School of Computer Science, Fudan University
l.j.q.light@gmail.com

Yanghua Xiao*
Shanghai Key Laboratory of Data Science,
School of Computer Science, Fudan University
Xiaoi Research, Shanghai, China
shawyh@fudan.edu.cn

Yi Zhang
School of Computer Science,
Fudan University
z_yi11@fudan.edu.cn

Seung-won Hwang
Yonsei University
seungwonh@yonsei.ac.kr

Haixun Wang
Facebook, USA
haixun@gmail.com

Abstract

Knowledge base(KB) plays an important role in artificial intelligence. Much effort has been taken to both manually and automatically construct web-scale knowledge bases. Comparing with manually constructed KBs, automatically constructed KB is broader but with more noises. In this paper, we study the problem of improving the quality for automatically constructed web-scale knowledge bases, in particular, *lexical taxonomies* of isA relationships. We find that these taxonomies usually contain cycles, which are often introduced by incorrect isA relations. Inspired by this observation, we introduce two kinds of models to detect incorrect isA relations from cycles. The first one eliminates cycles by extracting directed acyclic graphs, and the other one eliminates cycles by grouping nodes into different levels. We implement our models on Probase, a state-of-the-art, automatically constructed, web-scale taxonomy. After processing tens of millions of relations, our models eliminate 74 thousand wrong relations with 91% accuracy.

Introduction

Machine intelligence relies on a variety of knowledge bases, which are constructed manually or automatically. Examples of manually constructed knowledge bases include WordNet (Miller 1995) and Cyc (Lenat and Guha 1989), and examples of automatically constructed ones include KnowItAll (Etzioni et al. 2004), NELL (Mitchell et al. 2015), and Probase (Wu et al. 2012). Manually constructed knowledge bases are highly precise, but are limited in scale, while automatically constructed ones have high coverage but relatively low accuracy.

*Correspondence author. This paper was supported by National Key Basic Research Program of China under No.2015CB358800, by the National NSFC (No.61472085, U1509213), by Shanghai Municipal Science and Technology Commission foundation key project under No.15JC1400900, by Shanghai Municipal Science and Technology project under No.16511102102. Hwang was supported by IITP grant funded by the Korea government (MSIP; No. B0101-16-0307) and Microsoft Research.

The goal of this paper is to design algorithms to detect and eliminate errors in automatically constructed knowledge bases. In particular, we focus on *lexical taxonomies*, an important type of knowledge base consisting mainly of isA relations, such as `apple` *isA* `fruit`, where the "isA" refers to the *hyponymy* relation. Taxonomies are important because they map instances to concepts, thus enabling the machine to obtain the ability of generalization and specialization when understanding the text. Consequently, taxonomies, especially those machine-generated ones with a larger coverage, have been widely used in various text understanding tasks. Thus, detecting and eliminating errors in taxonomies are essential to improve machine intelligence.

In this work, we use a state-of-the-art, data-driven taxonomy, Probase (Wu et al. 2012), as an example for taxonomy cleansing. Although we focus on Probase, our solutions are applicable to other data-driven taxonomies. Probase contains 16 million isA relations, which are automatically extracted from 1.7 billion web pages by using mainly the Hearst syntactic patterns (Hearst 1992). In Probase, each isA relation is associated with a frequency observed in the web corpus. The accuracy of Probase is reported to be 92% (Wu et al. 2012), which is lower than WordNet. Table 1 shows some errors in Probase. Most of the errors are caused by errors in the corpus, or mistakes made by information extraction algorithms. For example, a typo (**as** should be **an**) in the following sentence `"... make Paris such` **`as`** `exciting city"` leads to the extraction of `exciting city` *isA* `Paris` by algorithms that use the `such as` pattern for extraction.

Entity	*isA*	Concept	Entity	*isA*	Concept
exciting city	*isA*	paris	battery	*isA*	fuel cell
automobile	*isA*	lead acid battery	cause	*isA*	tsunami
music video	*isA*	youtube video	sweet	*isA*	glucose
world cup	*isA*	football	grape	*isA*	purple
college	*isA*	basketball	juice	*isA*	tomato

Table 1: Examples of incorrect isA relations in Probase

To address the problem, we need to first detect the sus-

picious isA relations in the taxonomy. There are two naive approaches for this problem.

- **Use frequency.** Relations with a low frequency may be suspicious, because the relations introduced by typos or algorithmic extraction problems are seldom observed in the corpora. However, frequency information of isA relations itself follows a power-law distribution with a long tail, which implies that most relations with or without errors both have a low frequency. For example, about 7 million edges have frequency 1 in Probase. But our sample test shows that 78% of them are correct. Thus, we can not simply identify all relations with a low frequency as suspicious edges.
- **Use external knowledge.** Another method is employing external knowledge bases available to eliminate the conflicts and improve the quality of the taxonomy. However, some knowledge bases such as Probase have many specific concepts which do not exist in many other knowledge bases. As a result, the membership relations of an instance to a specific concept will be missing in external knowledge bases. For example, Probase has 2.7 million concepts while Yago has only 0.48 million types and DBpedia has only 700 types. Due to the huge gap between the concept coverage of Probase and external knowledge bases, it is impossible to use them to find conflicts in Probase.

In this paper, we propose to use structural information. The key difference between manually and automatically constructed taxonomies is whether it is a *directed acyclic graph* (DAG), i.e. there are no cycles in the taxonomy. Figure 1 illustrates a DAG taxonomy where many specific entities, such as `iphone 6`, `nexus 5`, `shanghai` are placed in the lower level while more abstract concepts, such as `thing`, `concept`, `object` are in the higher level.

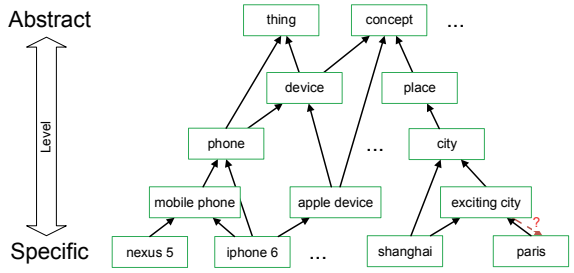

Figure 1: An ideal taxonomy example

Main sources of cycles in Probase are:

- **Ambiguity**: A word or phrase may have multiple senses, which are not differentiated in Probase. As a result, `word` may refer to Microsoft Word, which is a `software` (as an entity), or its literal meaning as an abstract concept. Hence, both `word` *isA* `software` and `software` *isA* `word` are correct and exist in Probase. These two relations constitute a cycle.
- **Wrong isA relations**: A wrong isA relation may be extracted, such as `exciting city` *isA* `paris`, which causes a cycle (as shown in Figure 2).

Cycles are important sources of locating suspicious relations. We sampled 100 of the entire Probase cycles of size 2 and 3, respectively, and compared them with a *null model* of randomly sampling 100 subgraphs of size 2 and 3 (with or without cycles) respectively. Then we manually judge whether each subgraph contains wrong isA relations. We report the $z-$score, which shows the degree of deviation between the samples from cycles and null models. The result is shown in Table 2. We can see that most cycles contain wrong isA relations, which is statistically significant since the corresponding random substructure tends to have a smaller number of wrong isA relations and the z-score is sufficiently large. We also illustrate examples of cycles with wrong relations in Figure 2.

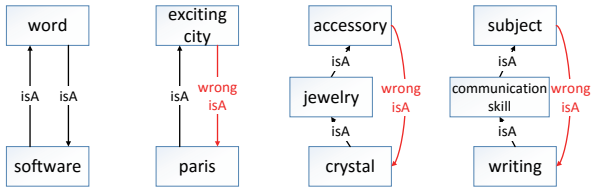

Figure 2: Cycle examples

Size	Have error	Null model	z-score	p-value
2	97%	15%	22.96	<0.0001
3	96%	24%	16.86	<0.0001

Table 2: Cycle statistics in Probase

Inspired by the above observation, we use cycle elimination methods to identify wrong isA relations. Though this problem has been studied before, existing solutions cannot be applied due to the following challenges:

1. First, *enumerating all cycles in a graph is computationally hard.* The number of cycles in a graph is exponential in the worst cases. A brute-force enumerating method for detecting all cycles in a graph is computationally prohibitive on web-scale data-driven taxonomies.
2. Second, *not all the isA relations in cycles are wrong*. We thus need a metric to quantify *trustworthiness* to determine which relation is wrong.

To overcome the above challenges, we explore two kinds of methods and propose an efficient solution for each model. The first one aims to extract a DAG from the given graph, for which we propose an efficient method (*maximal feedback arc set: MFAS*) to minimize the trustworthy metric of the removed edges. And the alternative one models the problem as assigning levels (an integer) to each node in the taxonomy so that a specific concept or instance has a low level and an abstract concept has a high level. Thus, we can eliminate the edges from high level nodes (abstract concepts) to low level nodes (specific entities) as wrong isA relations. In summary, we made the following contributions:

- First, we show that cycles are good indicators to find wrong isA relations in data-driven taxonomies.

- Second, we propose graph based models as well as their algorithmic solutions to find wrong isA relations in cycles.
- Third, we validate our solutions by processing a real-life web-scale taxonomy.

DAG Decomposition based Model

An ideal taxonomy is cycle-free and bears a structure of DAG as discussed in the previous section. This motivates us to model our problem by the DAG decomposition framework (See Def 1). Hence, the identification of wrong isA relations is equivalent to the identification of an acyclic subgraph.

Definition 1 (DAG Decomposition) *Given a directed graph $G(V, E)$, find a subgraph $D(V, E_D)$ of G such that D is acyclic and $q(E_R)$ is minimized, where $E_R = E \setminus E_D$ and $q(E_R)$ is a penalty function of E_R.*

Since E_D and E_R are complementary to each other, we can just focus on the evaluation of E_R for simplicity. In general, we hope that most members in E_R are truly wrong isA relations without false positives. Thus, a general principle to define $q(E_R)$ is the sum of the trustworthiness of each member in E_R. As a result, minimizing $q(E_R)$ is the appropriate objective for finding most suspicious isA relations without removing the highly reliable edges.

MFAS Model and Algorithm

One way to define $q(E_R)$ is, when $w(e)$ is the trustworthiness of an edge e, $q(E_R) = \sum_{e \in E_R} w(e)$. That is , the sum of the trustworthy score of all edges in E_R. The formalized problem is in Def 2. This model is a *weighted minimum feedback arc set problem* (weighted MFAS problem), which is a classical NP-Hard problem (Even et al. 1998).

Definition 2 (MFAS) *Given a weighted directed graph $G(V, E)$, where each edge is associated with a weight $w(e)$, find an edge subset $E_R \subseteq E$ such that (1) $D(V, E \setminus E_R)$ is a DAG and (2) the weight sum of E_R (i.e., $\sum_{e \in E_R} w(e)$) is minimized.*

Since the problem is NP-Hard, we focus on an efficient approximate algorithm. Specifically, we propose a greedy algorithm: *repeatedly find a cycle and remove all edges with the lowest weight until there are no cycles in the remaining graph.* Obviously, the resulting graph must be a DAG. However, too many edges might have been removed so that the accumulated weight of the removed edges is far away from the optimal. Demetrescu et al. (Demetrescu and Finocchi 2003) proposed a subtle heuristic to improve it. The algorithm has two steps. The first step is same as the basic greedy strategy. In the second step, it checks the edges removed one by one in the descending order of edge weight. For each edge removed, it tries to add the edge back to the graph and judges whether a cycle is created by the addition. If not, the edge is added back. Clearly, this algorithm removes fewer edges to generate a DAG than the basic greedy algorithm. Actually, it was proven that this improved algorithm achieves a λ-approximation, where λ is the length of the longest cycle in the graph. The time complexity of this algorithm is $O(nm)$ (Demetrescu and Finocchi 2003), where n is the number of the nodes and m is the number of edges in the graph.

w_f range	Accuracy
1	78%
2-10	86%
11-100	94%
> 100	100%

Table 3: Effectiveness of w_f

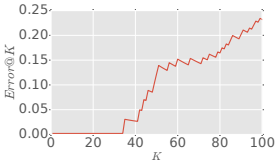

Figure 3: Effectiveness of P_h

Trustworthiness Metric

Our next step is defining edge weights to quantify trustworthiness. First, we present a basic metric that uses frequency for isA relations in Probase. Then we propose how to improve this metric.

Basic Metric: Frequency Recall that each isA relation X *isA* Y in Probase is associated with a *frequency* that the isA relation is observed in corpus. We use $w_f(e)$ to denote the frequency of edge e. Intuitively, a larger weight implies higher trustworthiness. Example 1 illustrates this. Our empirical study shows that w_f is effective in characterizing trustworthiness of isA relations. We randomly sample 50 edges from some frequency ranges respectively in Probase, and then manually judge their correctness. Table 3 shows that a larger frequency implies higher trustworthiness.

Example 1 *The frequency of* `china` *isA* `country` *in Probase is 10,723, which means that there are 10,723 sentences containing this isA relation. In contrast, the frequency of* `exciting city` *isA* `paris` *is only one. Obviously, the former is much more trustworthy than the latter.*

Improved Metric: Using Hyponym Numbers However, the above metric has a significant weakness: it is less discriminative for edges with low frequency. Among the 7 million edges with frequency 1, only a small part of them is wrong. Hence, we need to integrate more signals besides frequency.

A simple observation on data-driven taxonomies is that abstract concepts always have many hyponyms (words of more specific meaning), but specific concepts or entities always have few or no hyponyms. Example 2 illustrates these facts.

Example 2 (Hyponyms) *In Probase,* `concept` *has 18,832 hyponyms. In contrast, a more specific concept like* `exciting city` *has 30 hyponyms such as* `paris`, `london`, `shanghai`. *Moreover, most specific entities such as* `shanghai` *have no hyponyms.*

Thus, given an isA relation X *isA* Y, Y in general is more abstract than X and Y's hyponym number is supposed to be larger than X's. The larger the difference is, the more trustworthy the edge is. More formally, let $hypo(X)$ be the number of hyponyms of X, we define P_h to represent our belief about the correctness of an isA relation:

$$P_h(X \textit{ isA } Y) = \log\left(1 + \frac{hypo(Y)}{hypo(X)}\right) \quad (1)$$

where the larger P_h is, the more likely the isA relation is true. Note that P_h is undefined when $hypo(X) = 0$. However, we will show in the next subsection that we only need to handle nodes in strong connected components in the taxonomy, which always have a $hypo$ value larger than 0.

We multiply P_h with w_f to derive a new metric w_{fh} in Eq 2. Clearly, P_h plays a role of a modified factor. We use it to discriminate the edges with the same frequency. Example 3 illustrates the effectiveness of the new metric.

$$w_{fh}(e) = w_f(e) \times P_h(e) \quad (2)$$

Example 3 (Effectiveness of w_{fh}) *e_1=`juice` isA `tomato` and e_2=`tomato` isA `traditional food` in Probase both have frequency 1. However, $P_h(e_1) = 0.34$ and $P_h(e_2) = 1.85$. ($hypo($`juice`$) = 173$, $hypo($`tomato`$) = 69$, $hypo($`traditional food`$) = 372$). P_h score indicates that e_2 is more trustworthy than e_1. Hence, P_h is helpful in the discrimination of edges with the same weight.*

We further use a statistical study to show the effectiveness of P_h. We randomly sample 100 edges with frequency 1, and manually judge their correctness. We sort them by P_h in descending order, and then measure $Error@K$ (the proportion of errors in the top-K results) for each K. The result, shown in Figure 3, reveals a clear correlation between $Error@K$ and K. In general, a higher P_h leads to a higher precision. When $K \leq 34$, almost 100% precision is achieved. The fact strongly proves P_h's effectiveness.

Level Assignment based Model

An alternative way to solve the problem is clustering nodes into levels, such that correct edges only exist from specific nodes to abstract nodes, which ensures a DAG.

Preliminary of Level Assignment

An ideal taxonomy can be arranged level by level, as shown in Figure 1. More generally, a directed graph G has a level-wise layout once a level is assigned for each node. We define *level assignment* (LA) function for nodes in a directed graph G (see Def 3). For each LA function, we can construct a *unique acyclic* subgraph of G, which consists of all edges from *low level nodes* to *high level nodes* (see Def 4 and Lemma 1). Hence, *once we find a level assignment for the directed graph G, we derive a unique DAG, which implies a unique remaining edge set*. Thus, alternatively, we can find wrong isA relations by looking for a good level assignment function. Then our model is: *Given a taxonomy $G(V, E)$, find a level assignment function l with minimal $q(E_R)$, where $E_R = E \setminus E_D$ and $D(V, E_D)$ is the acyclic subgraph implied by l.* This model is closely related to the DAG based model. The LA based model actually can be considered as an additional step to specify a DAG.

Definition 3 (Level Assignment (LA)) *Given a directed graph $G(V, E)$, node level assignment function $l : V \to \mathcal{N}$ is a mapping from V to $\mathcal{N}$ such that each $l(v)$ is a non-negative integer representing the level of v.*

Definition 4 (Subgraph implied by LA) *Given a directed graph $G(V, E)$ and an LA function l defined on G, we refer to the graph $D(V, E \setminus E_R)$ with $E_R = \{(x, y)|(x, y) \in E, l(x) \geq l(y)\}$ as the* subgraph of G implied by l.

Lemma 1 (Acyclicity) *For a directed graph $G(V, E)$ and an LA function l defined on G, the subgraph implied by l is acyclic.*

The benefit of modeling our problem by LA is that we can use the level information to infer the wrong isA relations directly. Since we use all *forward* edges x *isA* y such that $l(x) < l(y)$ to construct the DAG, all the *backward* edges x *isA* y such that $l(x) \geq l(y)$ will be returned as errors. A good LA ensures that an abstract concept has a high level and an entity (or a specific concept) has a low level. Hence, any edges from high level nodes to low level nodes tend to be wrong isA edges. We use topological sorting as a baseline to find a good level assignment. To adjust topological sorting for a directed graph that might have cycles, we remove the node with minimum degree instead of the node with zero degree in each loop.

Agony Model

The baseline based on topological sorting does not consider the weight information of isA relations. Next, we incorporate the weights to derive a better level assignment.

Penalty Metric Recall that our basic idea is using an LA l to identify all the *backward* edges as the wrong isA relations. In general, we hope to minimize the number of false positives. That means an isA relation of higher *trustworthiness* should be penalized more if it is identified as the backward edge by the LA. We resort to such penalties to help find the best level assignment. We consider the following principles to define the penalty. We illustrate these principles in Example 4.

1. First, the more errors incurred, the higher the penalty is.
2. Second, the more reliable the edge is in terms of other signals (such as edge weight), the higher the penalty is.

Example 4 (Penalty Metric) *`apple` isA `fruit` has a very high weight in Probase, which is a strong evidence that `apple` is more specific than fruit. If an LA gives apple an abstract level, this LA should be punished. The more penalty should be given if 1) the level difference between `apple` and `fruit` in this LA is more significant; 2) the weight of this edge is larger.*

The simplest way to measure how much error is incurred for a correct isA relation x *isA* y is the level difference, defined as follows:

$$d(x, y) = l(x) - l(y) + 1. \quad (3)$$

A positive $d(x, y)$ means that x is more abstract than y . We add 1 to $d(x, y)$ because we still have to penalize an edge x *isA* y when $l(x) = l(y)$. To reflect the second principle, we reuse the two metrics adopted in the previous section. We still use $w(e)$ to represent one of them. By combining the above two factors, the final penalty for an edge x *isA* y is:

$$penalty(x, y) = d(x, y)w(x, y) \quad (4)$$

This metric clearly punishes more for a bad assignment on a reliable isA relation.

Problem Model Given the penalty metric, our problem is reduced to finding an LA such that the penalty sum over all edges in E_R is minimized. More formally, our problem is:

$$\arg\min_{l} \sum_{(x,y)\in E_R} d(x,y)w(x,y) \quad (5)$$

Obviously, this optimization objective prohibits the highly reliable edges to be backward edges. The objective function can be transformed into the sum over all edges in E when we assign zero value to $d(x,y)$ for all forward edges:

$$\sum_{(x,y)\in E_R} d(x,y)w(x,y) = \sum_{(x,y)\in E} \max\{0, d(x,y)\}w(x,y) \quad (6)$$

Definition 5 *Given a weighted directed graph $G(V,E)$, and edge weight $w(e)$ for each $e \in E$, find an level assignment function l^* on G such that*

$$\sum_{(x,y)\in E} \max\{0, d(x,y)\}w(x,y)$$

is minimized, where $d(x,y) = l^*(x) - l^*(y) + 1$.

Given above objective functions, we formalize our problem in Def 5, which is known as the *Agony* (Gupte et al. 2011) model and is widely used to find a hierarchy from a directed graph. The good news is that Agony problem is in P and many efficient solutions are available (Tatti 2014) when the weights are natural numbers.

Algorithm It was proved that Agony problem has a dual-problem known as *circulation* problem (Tatti 2015). Hence, we first solve the circulation problem by the algorithm proposed by (Edmonds and Karp 1972). With some optimizations, the time complexity is $O(n^2m)$, where n and m are the number of nodes and edges in the graph, respectively. The algorithm is guaranteed to have an integer-based solution when the weight is integer. In our setting, we have two options for the weight: $w_f(e)$ and $w_{fh}(e)$. Clearly, $w_{fh}(e)$ is not necessarily an integer. To use $w_{fh}(e)$, we round $100 \times w_{fh}(e)$ to the nearest integer for each edge and feed the integers into the algorithm.

Optimization: Agony+ In the previous solution, we remove all backward edges after we got a level assignment. This method might remove too many edges that are not necessarily to be deleted to derive a DAG. The reason is that the removal of a backward edge might break another cycle. In order to reduce the number of false positives, we modify the generic algorithm in two aspects: (1) We remove backward edges (x *isA* y such that $l(x) \geq l(y)$) in the ascending order by $l(y) - l(x)$ (which is a negative or zero value), and use the edge weight to break the tie (also in ascending order); (2) For each backward edge, we only remove it when it still lies in a cycle in the remaining taxonomy. We use $l(y) - l(x)$ as the heuristic to select backward edge to remove with a high priority. The reason is that a smaller $l(y) - l(x)$ means a larger probability that x *isA* y is a wrong isA relation. The time complexity is $O(km)$, where k is the number of backward edges and m is the edge number of the graph. We refer to this improved implementation as Agony+.

Experiments

In this section, we systematically evaluate the effectiveness and efficiency of the models and solutions proposed in previous sections.

Exp 1: Results on Probase

In this experiment, we report the results of our solutions in a state-of-the-art web-scale data-driven taxonomy Probase. We use Probase's core version, which has 10,390,064 concepts or entities, and 16,285,394 isA relations. To run our solutions on a taxonomy with tens of millions of nodes is a big challenge. Fortunately, most real taxonomies including Probase can be split into a collection of SCCs (strongly connected components) and the largest SCC usually is significantly smaller than the entire taxonomy. For Probase, the largest SCC has only 0.1M nodes and 1.4M edges. Hence, we use SCC decomposition to run our solutions in acceptable time.

Metrics and Settings Specifically, we evaluate the *precision, recall, and running time* of different solutions:

- **Precision** is the proportion of the truly wrong isA relations in all detected wrong isA relations. We randomly sample 300 wrong isA relations produced by each solution, and then ask volunteers to manually judge whether they are really wrong.
- **Recall** measures how many truly wrong isA relations are found. The recall is hard to be computed because we do not know the total number of truly wrong isA relations. Hence, we use the *maximal number of truly wrong isA relations detected by all competitive solutions* as the denominator to compute the recall. Thus, we actually report *relative recall*.
- **Running time** is used to show the scalability of different solutions. We run all solutions on a server with Intel(R) Xeon(R) E5-2632 CPU and 128GB RAM.

Competitors We compare our solutions MFAS, Agony and Agony+ to the topological sorting based baseline solution. For each of our model, we have two edge trustworthiness metrics. We use #1 to denote w_f and #2 to denote w_{fh}. Baseline+ is an improved version of the topological sorting based baseline. Similar to Agony+#2, Baseline+ removes the backward edges with the optimized strategy.

Setting	Time	*# removed*	Precision	*# truly wrong*
Baseline	3min	281.1K	71.0%	199.5K
Baseline+	1.1h	260.7K	72.3%	188.5K (94.5%)
MFAS#1	1.9h	67.1K	86.0%	57.7K (28.9%)
MFAS#2	10.6h	68.7K	90.7%	62.3K (31.2%)
Agony#1	43h	89.5K	83.7%	74.9K (37.5%)
Agony#2	89h	102.3K	84.7%	86.7K (43.4%)
Agony+#1	43h	55.0K	85.7%	47.1K(23.6%)
Agony+#2	89h	74.2K	**91.3%**	67.7K(33.9%)

Table 4: Evaluation results on Probase. The *# truly wrong* is estimated by multiplying *precision* and *# removed*. And the percentage in the last column is the *relative recall*, which is *# truly wrong* of each method divided by the maximal one (199.5K).

Experiment Result & Analysis The experimental result on Probase is shown in Table 4.

- **Precision & Recall** It is easy to see that the baseline model is worst in terms of precision, although it produces more results. In real applications, we want to avoid the removal of too many correct isA relations. Hence, baseline approach in general is unusable due to its 71% precision. The comparison also reveals that w_{fh} is always better than w_f in both precision and recall. This suggests that P_h performs well in differentiating edges with the same frequency. In general our Agony+#2 model achieves the highest precision and a relatively higher recall.
- **Running Time** From Table 4, we can see that our methods in general can process web-scale taxonomies in acceptable time. The performance of MFAS model is better than that of Agony (Agony+), implying that the greedy algorithm designed for MFAS is efficient. The comparison also reveals that the computation of w_{fh} in general is slower than that of w_f. This is obvious since the computation of w_{fh} is more complicated.

Case Studies We give some wrong isA relations found by our highest precision method in Table 1. All these isA relations are obviously wrong, which will cause serious problems when we understand the entities by their concepts.

Exp 2: Effectiveness of Level Assignment

A side product of our solutions is the level assignment. In MFAS-based methods, we use topological sorting to generate the levels since their remaining graph is a DAG. Hence, we conduct this experiment to show that the levels generated by our solution on automatically constructed lexical taxonomies are correlated with that on ideal taxonomies.

We use *the levels generated from WordNet* (Miller 1995) as the ground truth. Since WordNet is a prefect DAG, we directly use topological sorting to derive the level for each synset. We randomly sample 1000 chains in WordNet, such as `neural network` $\rightarrow$ `reticulum` $\rightarrow$ `network` $\rightarrow$ `system`. For each node in the chain, we calculate their levels in Probase and WordNet respectively. Then we compute Pearson's correlation coefficient (r) between them. Finally, we report the average over 1000 sampled chains.

Setting	Pearson's r (avg $\pm$ std)
Baseline	0.568 ± 0.38
MFAS#1	0.669 ± 0.38
MFAS#2	0.651 ± 0.40
Agony#1	0.639 ± 0.42
Agony#2	$\mathbf{0.692 \pm 0.37}$
Agony+#1	0.623 ± 0.39
Agony+#2	0.684 ± 0.38

Table 5: Evaluation on level assignment

The result is shown in Table 5. It is obvious that all our models have a mean Pearson's correlation coefficient larger than 0.6, which suggests a high correlation between our level and WordNet level. Furthermore, the Agony+#2 model achieves the best.

Exp 3: Evaluation on WikiTaxonomy

In this experiment, we show the *universality* of our methods. We repeat Exp 1 on another auto-constructed taxonomy WikiTaxonomy (Ponzetto and Strube 2008), which is a taxonomy auto-constructed from Wikipedia corpus. Since the Wikipedia corpus is much clearer and smaller than free web corpus, WikiTaxonomy is more accurate but much smaller than Probase, with only about 100 thousand concepts and relations. However, it also contains cycles and wrong isA relations. The experiment setting is similar to Exp 1 except that WikiTaxonomy does not contain relation weights. Hence, we must use w_{fh} with $w_f \equiv 1$.

Setting	Time	*# result*	Truly wrong	Precision
MFAS	$\sim$1sec	108	100	92.6%
Agony	$\sim$1sec	112	102	91.1%
Agony+	$\sim$1sec	108	101	**93.5%**

Table 6: Evaluation on WikiTaxonomy

The result is shown in Table 6. It shows that WikiTaxonomy contains only hundreds of wrong isA relations and our solution can consistently achieve high precision (over 90%) on this dataset. This result sufficiently shows that our methods are effective to cleanse a wide range of data-driven taxonomies.

Related Work

Taxonomy Construction Many taxonomies have been constructed manually or automatically. Early taxonomies such as WordNet (Miller 1995) and Cyc (Lenat and Guha 1989) are constructed by human experts. They are highly precise but limited in scale, which motivated automatic construction of larger taxonomies. Existing efforts consider different sources, such as texts and tables. First, isA relations from Web corpus are extracted using Hearst (Hearst 1992) patterns and other isA patterns, generating taxonomies of billions of nodes such as Probase (Wu et al. 2012) or Google isA databases. WikiTaxonomy (Ponzetto and Strube 2008) and Yago (Suchanek, Kasneci, and Weikum 2007) are extracted from Wikipedia corpus. Alternatively, structured HTML tables (Dalvi, Cohen, and Callan 2012) or semi-supervised extractors can be trained (Kozareva and Hovy 2010) for specific domains. However, removing cycles is a common challenge for both categories, which is addressed in this work.

Conflict Resolution in Knowledge Bases In the context of manually constructed knowledge and databases, finding and resolving conflicts in data integration is important. Naiman et al. (Naiman and Ouksel 1995) identify and classify types of semantic conflicts in heterogeneous databases. Lu et al. (Lu et al. 1998) use correlation analysis and statistical regression analysis in order to detect and resolve semantic conflicts in multiple data source integration and in data mining aspect. Li et al. (Li and Ling 2004) use OWL-based method to detect several conflict types and resolve them in RDF knowledge base integrations. However, these

methods cannot be applied to noisy automatically generated taxonomies.

Hierarchy Generation in Directed Graphs Extracting a DAG subgraph from the graph, or finding an estimated hierarchy, has been studied in various contexts. One such example is studying hierarchy in social network (Clauset, Moore, and Newman 2008; Gupte et al. 2011; Henderson et al. 2012; Maiya and Berger-Wolf 2009). Estimating the high-level node, in the social network (Cherkassky and Goldberg 1999; Even et al. 1998; Jameson, Appleby, and FREEMAN 1999) is useful to find a person with influence. Some intuition and algorithms used in these works inspire our solution.

Discussion and Conclusion

As future works, we will further study the following three problems. First, how to aggregate the different models proposed in this paper to achieve a better performance. Second, some (but few) cycles detected might be reasonable. How to identify them is an interesting problem. Third, our models might have multiple solutions. Which one of them is best deserves a further study.

We studied the problem of identifying wrong relations from automatically constructed taxonomies. Our key observation is that a cycle is highly likely to contain a wrong relation. We thus abstract our problem as enumerating cycles and eliminating the relation with low trustworthy score, using two models, of extracting a DAG or estimating a hierarchy from the graph, and proposing efficient solutions, namely MFAS and Agony schemes, from each model respectively. We validate our solutions by processing real-life web-scale taxonomies.

References

Cherkassky, B. V., and Goldberg, A. V. 1999. Negative-cycle detection algorithms. *Mathematical Programming* 85(2):277–311.

Clauset, A.; Moore, C.; and Newman, M. E. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* 453(7191):98–101.

Dalvi, B. B.; Cohen, W. W.; and Callan, J. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *WSDM*, 243–252. ACM.

Demetrescu, C., and Finocchi, I. 2003. Combinatorial algorithms for feedback problems in directed graphs. *Information Processing Letters* 86(3):129–136.

Edmonds, J., and Karp, R. M. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 19(2):248–264.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Web-scale information extraction in knowitall. In *WWW*, 100–110. ACM.

Even, G.; Naor, J. S.; Schieber, B.; and Sudan, M. 1998. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica* 20(2):151–174.

Gupte, M.; Shankar, P.; Li, J.; Muthukrishnan, S.; and Iftode, L. 2011. Finding hierarchy in directed online social networks. In *WWW*, 557–566. ACM.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, 539–545. ACL.

Henderson, K.; Gallagher, B.; Eliassi-Rad, T.; Tong, H.; Basu, S.; Akoglu, L.; Koutra, D.; Faloutsos, C.; and Li, L. 2012. Rolx: structural role extraction & mining in large graphs. In *SIGKDD*, 1231–1239. ACM.

Jameson, K. A.; Appleby, M. C.; and FREEMAN, L. C. 1999. Finding an appropriate order for a hierarchy based on probabilistic dominance. *Animal Behaviour* 57(5):991–998.

Kozareva, Z., and Hovy, E. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *EMNLP*, 1110–1118. ACL.

Lenat, D. B., and Guha, R. V. 1989. *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc.

Li, C., and Ling, T. W. 2004. Owl-based semantic conflicts detection and resolution for data interoperability. In *Conceptual modeling for advanced application domains*. Springer. 266–277.

Lu, H.; Fan, W.; Goh, C. H.; Madnick, S. E.; and Cheung, D. W. 1998. *Discovering and reconciling semantic conflicts: a data mining perspective*. Springer.

Maiya, A. S., and Berger-Wolf, T. Y. 2009. Inferring the maximum likelihood hierarchy in social networks. In *CSE*, volume 4, 245–250. IEEE.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

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

Naiman, C. F., and Ouksel, A. M. 1995. A classification of semantic conflicts in heterogeneous database systems. *Journal of Organizational Computing and Electronic Commerce* 5(2):167–193.

Ponzetto, S. P., and Strube, M. 2008. Wikitaxonomy: A large scale knowledge resource. In *ECAI*, volume 178, 751–752.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *WWW*, 697–706. ACM.

Tatti, N. 2014. Faster way to agony. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 163–178.

Tatti, N. 2015. Hierachies in directed networks. In *ICDM*. IEEE.

Wu, W.; Li, H.; Wang, H.; and Zhu, K. Q. 2012. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, 481–492. ACM.