Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

On the Transitivity of Hypernym-Hyponym Relations in Data-Driven Lexical Taxonomies

Jiaqing Liang, Yi Zhang, Yanghua Xiao[*]
Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
l.j.q.light@gmail.com, {z_yi11, shawyh}@fudan.edu.cn

Haixun Wang
Facebook, USA
haixun@gmail.com

Wei Wang
School of Computer Science, Fudan University
weiwang1@fudan.edu.cn

Pinpin Zhu
Xiaoi Research, Shanghai Xiaoi Robot Technology Co. LTD., China.
pp@xiaoi.com

Abstract

Taxonomy is indispensable in understanding natural language. A variety of large scale, usage-based, data-driven lexical taxonomies have been constructed in recent years. Hypernym-hyponym relationship, which is considered as the backbone of lexical taxonomies can not only be used to categorize the data but also enables generalization. In particular, we focus on one of the most prominent properties of the hypernym-hyponym relationship, namely, transitivity, which has a significant implication for many applications. We show that, unlike human crafted ontologies and taxonomies, transitivity does not always hold in data-driven lexical taxonomies. We introduce a supervised approach to detect whether transitivity holds for any given pair of hypernym-hyponym relationships. Besides solving the inferencing problem, we also use the transitivity to derive new hypernym-hyponym relationships for data-driven lexical taxonomies. We conduct extensive experiments to show the effectiveness of our approach.

Introduction

Knowledge bases are playing an increasingly important role in many applications. Most knowledge bases, including WordNet (Miller 1995), Cyc (Lenat and Guha 1989), and Freebase (Bollacker et al. 2008), are manually crafted by human experts or community efforts. The coverage of manual knowledge bases, such as WordNet, is far from being complete (Sang 2007). For example, the concepts and instances below Animals and People in WordNet is quite limited (Pantel and Pennacchiotti 2006; Hovy, Kozareva, and Riloff 2009).

Much attention thus has been paid on deriving knowledge bases by automatic extraction from big corpora. The data-driven approaches produce many knowledge bases such as KnowItAll (Etzioni et al. 2004), NELL (Mitchell et al. 2015), and Probase (Wu et al. 2012). Data-driven knowledge bases in general are larger than manual knowledge bases, covering more entities, concepts as well as their relationships. For example, Freebase has thousands of types, while Probase has millions of concepts. With a larger coverage, data-driven knowledge bases are better at supporting large scale text understanding and many other tasks.

Data-driven Lexical Taxonomy

In this paper, we concentrate on a particular knowledge base: *lexical taxonomy* built by data-driven approaches. A lexical taxonomy consists of the *hypernym-hyponym* relations between terms. One term `A` is a hypernym of another term `B` if `A`'s meaning covers the meaning of `B` or much broader (Sang 2007). For example, `furniture` is a hypernym of `chair`. The opposite term for hypernym is hyponym. So `chair` is a hyponym of `furniture`. We use the expression *hyponym*(`A`, `B`) for a *hypernym-hyponym* relationship, which means `A` is a hyponym of `B`.

Hypernym-hyponym relations are backbones of text understanding. The reason hypernym-hyponym relationships hold such significance is that they enable *generalization*, which lies at the core of human cognition as well as at the core of machine inferencing for text understanding. To see this, *hyponym*(`iphone`, `smart phone`) enables machine to understand the search intent of `iphone` (i.e. `smart phone`). *hyponym*(`galaxy s4`, `smart phone`) further allows to recommend the related keyword `galaxy s4`.

Many automatically harvested lexical taxonomies such as Probase, YAGO (Suchanek, Kasneci, and Weikum 2007), WikiTaxonomy (Ponzetto and Strube 2008), are extracted from web corpora or Wikipedia by certain syntactic patterns (such as Hearst patterns (Hearst 1992)) or heuristic rules. For example, a sentence *"... famous basketball players such as Michael Jordan ... "* is considered an evidence for the claim that term `Michael Jordan` is a hyponym of term `famous basketball player`, while this sentence follows one Hearst pattern.

Problem Statement

In this paper, we focus on one of the most important properties of the hypernym-hyponym relationship: *transitivity*. For

[*]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.

human-crafted taxonomies, transitivity in a lexical taxonomy is taken for granted, that is, given *hyponym*(A, B) and *hyponym*(B, C), we know *hyponym*(A, C) (Sang 2007), as shown in Example 1. Transitivity is thus one of the cornerstones in knowledge-based inferencing, and many applications rely on transitivity (e.g., finding all the super concepts of an instance).

Example 1 *Is* Einstein *a* scientist*?*
hyponym(*einstein, physicist*)
hyponym(*physicist, scientist*)
$\Rightarrow$ *hyponym*(*einstein, scientist*)

Unfortunately, transitivity does not always hold in data-driven lexical taxonomies. Let us consider the following two examples:

Example 2 *Is* Einstein *a* profession*?*
hyponym(*einstein, scientist*)
hyponym(*scientist, profession*)
$\nRightarrow$ *hyponym*(*einstein, profession*)

Example 3 *Is a* car seat *a piece of* furniture*?*
hyponym(*car seat, chair*)
hyponym(*chair, furniture*)
$\nRightarrow$ *hyponym*(*car seat, furniture*)

It is obvious that Einstein is not a profession. However, in a data-driven lexical taxonomy such as Probase, we have strong evidence that *hyponym*(einstein, scientist) and *hyponym*(scientist, profession). If transitivity holds, we will draw a conclusion that conflicts with common sense. As for car seat and furniture, we are trapped in a similar situation. Thus, it is clear that transitivity does not always hold in data-driven lexical taxonomies.

One way out of this dilemma is to enforce word sense disambiguation, just as WordNet does. For example, we may manually distinguish the two senses of scientist in in Example 2. One of them is a person, which we denote as $\texttt{scientist}_{\textbf{person}}$, and the other is a profession, which we denote as $\texttt{scientist}_{\textbf{profession}}$. The evidence provided by the taxonomy is that $\texttt{scientist}_{\textbf{person}}$ is a hypernym of einstein, and profession is a hypernym of $\texttt{scientist}_{\textbf{profession}}$, which does not lead to the wrong conclusion *hyponym*(einstein, profession).

We argue that the above solution is flawed both in practice and in theory. In practice, it is too difficult to perform word sense disambiguation in a lexical taxonomy containing millions of concepts and millions of hypernym-hyponym relationships. In theory, it is not always possible to divide the meaning of a word into finite and discrete senses. Take the word chair as an example. There are many different types of chairs, including office chair, bench, stool, car seat, etc. We may have to decide whether chair in each of these senses is considered as furniture or not. To make it worse, manufacturers may make new types of chairs such as those can be used both in the car and in the room. Thus, we may end up having a concept for every single object, and such a conceptual system is useless because it does not have any generalization ability.

In summary, given *hyponym*(A, B) and *hyponym*(B, C), it is not a trivial task to tell whether transitivity holds, that is, whether we may draw the conclusion *hyponym*(A, C) or not. The goal of this paper is to devise a mechanism in this situation, to infer whether *hyponym*(A, C) holds or not for any given pair *hyponym*(A, B) and *hyponym*(B, C).

Naive Inference Mechanisms

Our goal is to decide if transitivity holds for a pair of hypernym-hyponym relation in the data-driven lexical taxonomy. That is, for any *hyponym*(a, b) and *hyponym*(b, c), we need to determine whether *hyponym*(a, c) holds or not. The direct inference mechanism is that if *hyponym*(a, c) can be observed frequently in corpora, it is quite likely that *hyponym*(a, c) holds. Since some taxonomies such as Probase contain *weights* or *probabilities* for each hypernym-hyponym relation, we wonder whether we can infer transitivity directly from such weights and probabilities.

Next, we show that this naive inference mechanism does not always work. The problem is, for any ⟨a, b⟩ and ⟨b, c⟩, where *hyponym*(a, b) and *hyponym*(b, c) hold, it is very likely that *hyponym*(a, c) has a zero or extremely small frequency to be observed. We cannot conclude whether *hyponym*(a, c) holds in this case, because the low frequency could be the result of either *data sparsity* or *noise*. In Figure 1, we give the distribution of $freq_{ac}$ (the frequency of c being a hypernym of a in Probase) over ⟨a, c⟩ pairs for which there exist a term b such that *hyponym*(a, b) and *hyponym*(b, c) hold. It shows that over 4.5×10^9 pairs of *hyponym*(a, c) have a zero frequency and the majority of the rest has a small frequency in Probase. This is the reason we need to infer transitivity from other evidence.

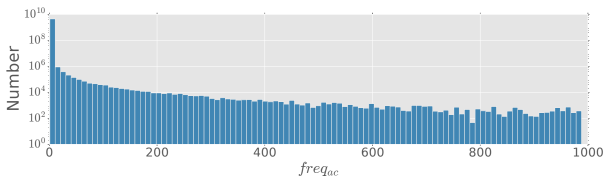


Figure 1: The frequency of *hyponym*(a, c). For most *hyponym*(a, b) and *hyponym*(b, c), *hyponym*(a, c) has a zero or extremely small frequency.

Inferring Transitivity

In this section, we present our method of checking whether transitivity holds for a given pair. We present our solutions mostly with respect to Probase, but the problem and the solution presented here apply to other large-scale data-driven lexical taxonomies as well.

A Simple Classifier

Before we dive into the details of inferring transitivity, we introduce a classifier, which uses several simple features, and discuss how the classifier can be improved for our purpose. It is clear that the problem of detecting transitivity can be modeled as a classification problem. The classifier accepts *hyponym*(a, b) and *hyponym*(b, c) as inputs, and produces *TRUE* or *FALSE* which indicates whether *hyponym*(a, c) holds or not.

Let $n(\langle \mathtt{a}, \mathtt{c}\rangle)$ be the frequency of *hyponym*(a, c) in Probase. We use it as the first feature in the classifier. Because when $n(\langle \mathtt{a}, \mathtt{c}\rangle)$ is high, transitivity mostly holds, since we have strong evidence of *hyponym*(a, c) in the data. However, when $n(\langle \mathtt{a}, \mathtt{c}\rangle)$ is low, we are not sure. As discussed, the low frequency could be a result of incorrect extraction (noise) or because either a or c is a rare term (data sparsity).

Hence, the classifier might need to use more features, including $n(\langle \mathtt{a}, \mathtt{c}\rangle)$, $n(\langle \mathtt{a}, \mathtt{b}\rangle)$,$n(\langle \mathtt{b}, \mathtt{c}\rangle)$, $n(\mathtt{a})$, $n(\mathtt{b})$, $n(\mathtt{c})$, where the last three denote the popularity of the three terms. This simple classifier, however, is not very powerful. In the rest of the section, we introduce more advanced signals, which can be used by the classifier for transitivity detection.

Siblings

Assume that we know *hyponym*(newton, physicist) and *hyponym*(physicist, scientist), and we need to determine whether *hyponym*(newton, scientist) holds or not. As shown in Figure 2(a), almost all instances of physicist (einstein, hawking, faraday, etc.) are scientist. This is a strong evidence that *hyponym*(newton, scientist).

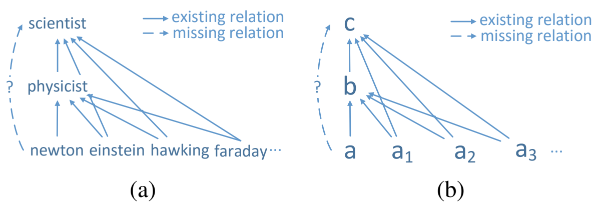


Figure 2: Using siblings to infer transitivity

We denote einstein, newton, hawking, ... as *siblings* (with regard to their shared hypernym physicist). Generally, as shown in Figure 2(b), if most of a's siblings are hyponyms of c, then it is reasonable to believe that a should be as well, and vice versa.

To simplify the notation, we also use a triple $t = \langle \mathtt{a}, \mathtt{b}, \mathtt{c}\rangle$ (we refer to the triple as a *hyponym triple*) to represent hypernym-hyponym relations *hyponym*(a, b) and *hyponym*(b, c). Thus, we introduce a new feature $sib_r(t)$ for our classifier:

$$sib_r(t) = \frac{|hypo(\mathtt{b}) \cap hypo(\mathtt{c})|}{|hypo(\mathtt{b})|}, t = \langle \mathtt{a}, \mathtt{b}, \mathtt{c}\rangle \qquad (1)$$

where $hypo(\mathtt{b})$ denotes the set of hyponyms of b. Intuitively, $sib_r(t)$ is the percentage of a's siblings that have c as their hypernym. We also use the number of common hyponyms of b and c (denoted as sib) as a feature.

We conduct a simple test to show the effectiveness of the above feature. We randomly sample 5k triples $\langle \mathtt{a}, \mathtt{b}, \mathtt{c}\rangle$ from our negative and positive labeled dataset (more details are in Section "Construction of the Labeled Dataset"), respectively. The average sib_r of negative samples is 0.015, and the average sib_r of positive samples is 0.132. The Wilcoxon rank-sum test shows that the difference between positive and negative samples is significant with p-value smaller than 2.2×10^{-16}.

Concept Similarity

Assume that we are to determine whether *hyponym*(ak-47, military weapon) holds given *hyponym*(ak-47, gun) and *hyponym*(gun, military weapon). From Figure 3(a), we observe that transitivity holds in many triples $\langle$ak-47, gun, $c_i\rangle$ where c_i is weapon, modern weapon, etc., and these c_is have a high semantic similarity to military weapon. Hence, it is reasonable to infer that military weapon is also a hypernym of ak-47.

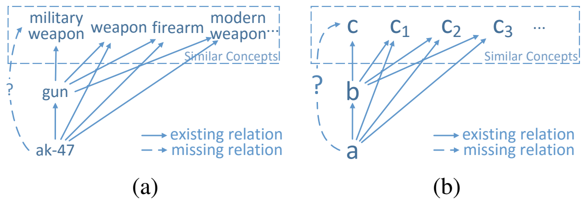


Figure 3: Using concept similarity to infer transitivity

Based on the above observation, we introduce a new feature for our classifier:

$$sim(t) = \frac{\sum_{c_i \in hype(\mathtt{a},\mathtt{b})} sim_c(\mathtt{c}, c_i)}{|hype(\mathtt{a}, \mathtt{b})|}, t = \langle \mathtt{a}, \mathtt{b}, \mathtt{c}\rangle \qquad (2)$$

where $hype(\mathtt{a}, \mathtt{b})$ is the set of common hypernyms of a and b, and $sim_c(c_1, c_2)$ measures similarity between two concepts c_1 and c_2. Intuitively, the feature measures how similar c is to other concepts which are hypernyms of both a and b. Specifically, the similarity function is defined as:

$$sim_c(c_1, c_2) = 1 - (1 - s_e(c_1, c_2)) \times (1 - s_o(c_1, c_2)) \qquad (3)$$

where $s_e(c_1, c_2)$ is the cosine similarity between c_1 and c_2's hypernym vectors, and $s_o(c_1, c_2)$ is the cosine similarity between c_1 and c_2's hyponym vectors. Intuitively, this is a noisy-or model that considers both hypernym and hyponym similarity. The rationale of using the noisy-or model is to amplify the similarity signal suppressed by the data sparsity problem in a data-driven knowledge base.

From Table 1, we can see that positive examples in general have a significantly larger sim than negative examples. Again, we use the 5k positive examples and 5k negative examples for some statistics tests. For the negative samples, the average value is 0.204, which is smaller than the average value 0.307 in positive samples. The Wilcoxon rank-sum test supports this conclusion strongly with p-value smaller than 2.2×10^{-16}.

Pos/Neg	Triple	sim
Positive	⟨ physicist, scientist, profession ⟩	0.545
Positive	⟨ ak-47, gun, dangerous weapon ⟩	0.406
Negative	⟨ newton, scientist, profession ⟩	0.140
Negative	⟨ ak-47, gun, combat skill ⟩	0.035

Table 1: Examples of similarity feature

Senses

For a triple $t = \langle \texttt{a, b, c} \rangle$, we consider the number of different senses b has. In other words, we want to know to what degree b is ambiguous. For example, the term search engine is not ambiguous. For any a that satisfies *hyponym*(a, search engine), e.g. a = bing, and for any c that satisfies *hyponym*(search engine, c), e.g. c = service, we probably have *hyponym*(a, c), because no semantic drift is possible when we follow the link from a through b to c. On the other hand, the term dish has many senses. For example, in *hyponym*(pasta, dish) and *hyponym*(dish, utensil), the term dish has two different meanings. Thus, it is very likely *hyponym*(pasta, utensil) does not hold.

In summary, for a triple $\langle \texttt{a, b, c} \rangle$, the more senses b has, the less likely the transitivity holds. To find out how many senses b has, we consult WordNet for the number of synsets of b. There are two cases we need to address: b exists or not in WordNet. In the first case, we use the number of synsets of b as the count of b's senses, denoted by $synsets(b)$. In the second case, b is a rare word that is less likely to be ambiguous and has a unique sense. Thus, we define a new feature as follows:

$$s_b(t) = \begin{cases} synsets(\texttt{b}) & \texttt{b} \in \text{WordNet;} \\ 1 & \text{otherwise.} \end{cases}, t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle \quad (4)$$

Furthermore, some of b's senses correspond to entities, and entities cannot be someone's hypernyms. For example, one of the senses of china is People's Republic of China, which is a specific entity and cannot be a concept. We don't count these synsets as b's senses number. A sense is an entity in WordNet if it has no hyponyms. Therefore, we revise Eq 4 accordingly:

$$sc_b(t) = \begin{cases} synsets(\texttt{b}) - \theta(\texttt{b}) & \texttt{b} \in \text{WordNet;} \\ 1 & \text{otherwise.} \end{cases}, t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle \quad (5)$$

where $\theta(\texttt{b})$ denotes number of senses of b that correspond to entities.

Other Inference Mechanisms

Besides the above features, we also consider another two categories of features, which can be derived from Probase. The first category includes all features, which can be computed based on the structure of Probase. For a relation *hyponym*(a, b) in Probase, we can regard it as a directed edge from a to b. Thus, Probase can be considered as a directed graph. Two basic measures in a directed graph are in-degree (number of hyponyms) and out-degree (number of hypernyms) of each node. Specifically, for a hyponym triple $t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle$, we consider all the following measures:

1. $u_x(t)$: the number of hypernyms of x, where $x \in \{\texttt{a}, \texttt{b}, \texttt{c}\}$.
2. $v_x(t)$: the number of hyponyms of x, where $x \in \{\texttt{a}, \texttt{b}, \texttt{c}\}$.

The second category of features is based on the frequency information of Probase. For a relation *hyponym*(a, b), Probase contains the frequency that the relation is observed from corpus. Based on the frequency, we define the following features:

1. $freq_x(t)$: $n(x)$, where $x \in \{\texttt{a}, \texttt{b}, \texttt{c}\}$.
2. $freq_y(t)$: $n(y)$, where $y \in \{\langle \texttt{a}, \texttt{b} \rangle, \langle \texttt{b}, \texttt{c} \rangle\}$.
3. $PMI_y(t)$: the PMI of y, where $y \in \{\langle \texttt{a}, \texttt{b} \rangle, \langle \texttt{b}, \texttt{c} \rangle\}$.

where PMI is point-wise mutual information. It is a measure of association in information theory and statistics. For a relation *hyponym*(a, b), $PMI_{ab}(t)$ is defined as:

$$PMI_{ab}(t) = log \frac{Pr(X = \texttt{a}, Y = \texttt{b})}{Pr(X = \texttt{a}) Pr(Y = \texttt{b})}, t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle \quad (6)$$

where X and Y are two random variables representing the two ends of a hypernym-hyponym relation.

For all the above measures except PMI, we feed them into the classifier with the logarithm of them. Specifically, we use $ln(1+x)$ to compute the logarithm of these features to avoid overflow caused by zero value of x.

Generate Missing Relations

One direct application of transitivity inference is the completion of lexical taxonomies. Next, we elaborate how to generate new hypernym-hyponym relations for Probase. For a given term pair, say $\langle \texttt{a}, \texttt{c} \rangle$, we need to determine whether *hyponym*(a, c) holds or not.

A direct idea is training a binary classifier with the features for a triple $t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle$, and deriving a classifier of term pairs from the classifier of triples. However, such derivation is not trivial. For two different triples $t_1 = \langle \texttt{a}, b_1, \texttt{c} \rangle$ and $t_2 = \langle \texttt{a}, b_2, \texttt{c} \rangle$, it is often the case that the classifier defined on triples produces conflicting results for *hyponym*(a, c).

In general, we need an appropriate aggregation strategy to predict the relation of a given term pair. We propose three strategies. In the first strategy, we build a classifier of term pairs and aggregate the information of different triples as features of the term pair. In the second and third strategies, we directly aggregate the results produced by the triple classifier.

- **Strategy 1: Classifier of term pairs.** Given a pair of terms $\langle \texttt{a}, \texttt{c} \rangle$, we compute all the metrics defined in the previous texts for each $t_i = \langle \texttt{a}, b_i, \texttt{c} \rangle$. Then we use mean pooling to aggregate the feature vectors from different triples. These features allow us to build a classifier of term pairs.
- **Strategy 2: Majority voting.** For all triples $t_i = \langle \texttt{a}, b_i, \texttt{c} \rangle$, *hyponym*(a, c) holds if and only if most t_is are predicted to be positive by the classifier of triples.
- **Strategy 3: Weighted voting.** From the labeled dataset, we can build a regressor instead of a classifier, e.g. a random forest regressor. The regressor allows us to predict a triple t_i with a value θ_i between $+1$ and -1 so that a positive/negative θ_i indicates a positive/negative label. $|\theta_i|$ expresses the confidence of the prediction. We thus use $|\theta_i|$ as the weight to implement a weighted voting strategy. Specifically, *hyponym*(a, c) holds if and only if $\sum_i \theta_i$ is positive.

Construction of the Labeled Dataset

We need a labeled dataset to develop an effective classifier. The task of labeling is to label whether *hyponym*$(\texttt{a}, \texttt{c})$ holds or not for the given hyponym triple $t = \langle \texttt{a}, \texttt{b}, \texttt{c} \rangle$. We may resort to human labeling. However, human labeling is costly and can only produce a limited number of labeled samples. Hence, an automatic method for labeling data is necessary. We use WordNet to construct more labeled samples.

Labeling by WordNet We use the hypernym-hyponym relationships among nouns in WordNet to construct a labeled dataset. As for our study, we used WordNet 3.0, which contains 82,115 synsets and 84,428 hypernym-hyponym relations among synsets. Each synset represents a sense of a word in WordNet. The hypernyms and hyponyms are assigned to each sense and they are certain senses of corresponding words.

To generate the labeled data, we first select a word $\texttt{b}$ in WordNet that has multiple senses. Let b_i and b_j be two *different* senses of $\texttt{b}$. Let a be the hyponym of b_i and c be the hypernym of b_j. We use any $t = \langle \bar{a}, \bar{b_i}, \bar{c} \rangle$ as a *candidate negative triple*, where $\bar{a}$ is the word of sense $\texttt{a}$. We use any triple $t = \langle \bar{x}, \bar{b_i}, \bar{y} \rangle$ such that x and y are the hyponym and hypernym of b_i, respectively, as *candidate positive triple*. In addition, we only use the triples whose words exist in Probase and WordNet simultaneously. Finally, we generate 9.9k positive triples and 9.4k negative triples. We find no conflicts in the labeled data set (i.e. no triples will be marked with both positive and negative labels). We illustrate the construction procedure in Example 4.

Example 4 *The word* `tank` *has two synsets in WordNet. The first is* $\texttt{tank}_1$*=storage tank, and the second* $\texttt{tank}_2$*=army tank. In WordNet, we find the following relations:*
hyponym$(\texttt{water tank}, \texttt{tank}_1)$, *hyponym*$(\texttt{tank}_1, \texttt{vessel})$,
hyponym$(\texttt{tank}_2, \texttt{military vehicle})$

Then we construct two samples:
$\langle \texttt{water tank, tank, vessel} \rangle$*: positive,*
$\langle \texttt{water tank, tank, military vehicle} \rangle$*: negative.*

That is, hyponym$(\texttt{water tank, vessel})$ *holds because the two relations use the same sense of* `tank`*. And hyponym*$(\texttt{water tank, military vehicle})$ *is wrong, because the two relations use different senses of* `tank`.

Evaluation

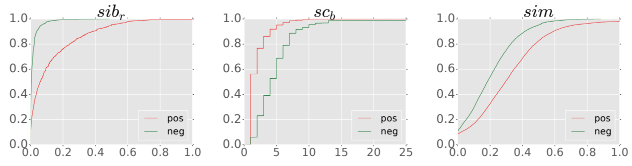
In this section, we evaluate the effectiveness of our approach. First, we evaluate the effectiveness of the features. Then we evaluate the quality of the hypernym-hyponym pairs predicted by our transitivity inference mechanisms.

Effectiveness of Features We use χ^2 and IG (information gain) to evaluate the effectiveness of the features used in the classifier. The features ranked by χ^2 are shown in Table 2. The results show that $sib_r(t)$ and $sc_b(t)$ are the top two features ranked by both two measures. We also give CDF (cumulative distribution function) for the top three features ranked by χ^2 in Figure 4. The results show that each feature can clearly separate the positive samples from negative samples. Similar CDF plots are observed for other features.

#	Feature	χ^2	IG%	#	Feature	χ^2	IG%
1	sib_r	844.50	20.44	11	PMI_{ab}	39.09	4.64
2	sc_b	461.64	23.39	12	v_a	37.07	0.62
3	sim	235.99	4.90	13	$freq_{ab}$	25.61	0.53
4	v_b	158.78	9.40	14	s_a	16.94	0.38
5	$freq_b$	82.09	6.73	15	v_c	4.90	1.32
6	sib	72.02	1.43	16	u_a	0.74	0.07
7	PMI_{bc}	70.20	5.12	17	$freq_c$	0.44	0.48
8	u_b	58.41	8.70	18	$freq_a$	0.08	0.05
9	$freq_{bc}$	53.74	1.32	19	u_c	0.08	1.42
10	sc_c	45.34	1.78				

Table 2: Effectiveness of features (sorted by χ^2)

Next, we evaluate the classification performance when top K features (sorted by χ^2) are used. We report *accuracy, precision, recall* as well as *F1-score* of the classier against K (the number of top features used in the model). All performance results (including those in the following experiments) are derived using 10-fold cross validation. We only report the result for the random forest model. Similar results are obtained for other models. The result is shown in Figure 5. We can see that in general the top 11 features dominate the performance. All the measures converge to an upper limit after the top 11 features are used. The results suggest that the top few features are effective enough for the classification.



(a) sib_r (b) sc_b (c) sim

Figure 4: CDF of features

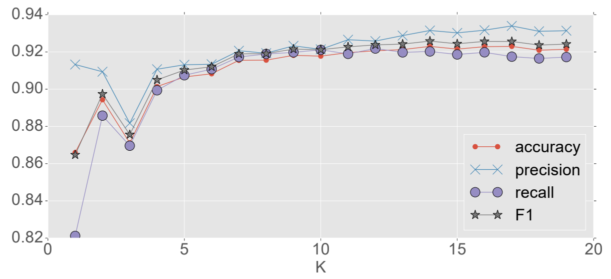


Figure 5: Performance of top-K features

Comparison of Classification Models We evaluate the effectiveness of our transitivity inference mechanisms under different classification models. We compare SVM (with rbf kernel and linear kernel), k-NN (both weighted and unweighted versions) and random forest. The result is shown in Table 3. We can see that the random forest model achieves the best result. The largest F1-score is about 92.2%, which suggests the practical usage of our approach.

Model	Accuracy	Precision	Recall	F1
rbf SVM	0.8009	0.8101	0.8114	0.8107
linear SVM	0.8054	0.8125	0.8186	0.8155
5-NN	0.8634	0.8788	0.8584	0.8685
15-NN	0.8510	0.8748	0.8361	0.8550
50-NN	0.8335	0.8735	0.7990	0.8346
weighted 5-NN	0.8764	0.8926	0.8695	0.8809
weighted 15-NN	0.8683	0.8912	0.8537	0.8720
weighted 50-NN	0.8534	0.8879	0.8253	0.8554
random forest	**0.9187**	**0.9276**	**0.9168**	**0.9221**

Table 3: Comparison of different classifiers

Effectiveness of Missing Relations Generation We compare the effectiveness of the three strategies to generate missing relations. In majority voting and weighted voting, we make decision for a term pair $\langle \mathtt{a}, \mathtt{c} \rangle$ by aggregating the results of the classifier for each triple $t = \langle \mathtt{a}, b_i, \mathtt{c} \rangle$. We directly train the classifiers of triples using the labeled triples in the previous experiments. Each labeled triple $t = \langle \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$ implies a labeled term pair $\langle \mathtt{a}, \mathtt{c} \rangle$ that has the same label. We use these labeled term pairs to train the binary classifier of term pairs. These labeled term pairs also serve as the test data for all the three strategies. The results are shown in Table 4, which shows that the weighted voting method is the best. Its F1-score is 93%, which guarantees the accuracy of the hyponym-hypernym relations found by our approaches.

Method	Accuracy	Precision	Recall	F1
Binary classifier	88.7%	90.2%	88.2%	89.2%
Majority voting	91.2%	**92.6%**	90.4%	91.5%
Weighted voting	**92.4%**	90.1%	**96.0%**	**93.0%**

Table 4: Three strategies to generate missing relations

Adding Missing Relations into Probase Next, we use the weighted voting to find missing relations for Probase. We notice that there are overall $O(N^2)$ term pairs to test, which is computationally prohibitive on Probase that contains tens of millions of concepts and entities. Hence, we prune the following $\langle \mathtt{a}, \mathtt{c} \rangle$ pairs:

- First, either `a` or `c` is a stop concept such as `concept, person, item`. These concepts are too vague; finding relations for them is not interesting.
- Second, there are only one or two intermediate concepts `b` to construct triples with $\langle \mathtt{a}, \mathtt{c} \rangle$. Because the transitivity inference is based on the hyponym triples. If we cannot find enough triples, the inference is unreliable.

After the filtering, we only need to consider 12.30M pairs. We finally find 3.86M true hyponym relations. We randomly sample 200 newly generated relations and manually judge their correctness. The precision is 92.5%, which is consistent with the precision of the weighted voting approach. We also give some samples of the new hyponym relations in Table 5. These examples sufficiently show that our inference approach can find many meaningful missing relations.

a	b	c
albertsons	supermarket, ...	large store
ipod touch	mp3 player, ipods, ...	consumer electronics
television monitor	display device, ...	device
shampoo	cosmetic, cleaning agent ...	daily good
linkedin	social network, website, ...	web service

Table 5: Some new hypernym-hyponym relations

Related Work

Taxonomy Construction Recently, taxonomy construction has attracted wide research interests. Hand-crafted taxonomies such as WordNet (Miller 1995) and Cyc (Lenat and Guha 1989) usually have high quality but low coverage over instances and concepts. Manual construction also suffers from high human cost, which motivates the research about the automatic construction. The automatic approaches can be classified into two categories according to the data source. The first extracts taxonomy from free web text. Hearst (Hearst 1992) patterns are widely used to extract hypernym-hyponym lexical relations from the web text. The extraction framework based on lexical patterns was further improved from different aspects. Some (Pantel, Ravichandran, and Hovy 2004) focus on finding more lexical patterns from hypernym examples, while some others (Snow, Jurafsky, and Ng 2004) exploit dependency patterns instead of lexical patterns for the extraction. The extraction models with Hearst patterns were also bootstrapped to improve the performance (Kozareva, Riloff, and Hovy 2008; Hovy, Kozareva, and Riloff 2009). To further improve the coverage of instances and concepts, Probase (Wu et al. 2012) was built from billions of web pages. Probase has frequency information and covers tens of millions of instances. The second category extracts taxonomy from relatively structured Wikipedia like web sites, such as WikiTaxonomy (Ponzetto and Strube 2008) and Yago (Suchanek, Kasneci, and Weikum 2007). Both of them are large-scale knowledge resources generated from categories in Wikipedia. In general, they have a better quality but a lower coverage than the taxonomies constructed from web texts. All the above literatures focused on building a taxonomy from scratch. Instead, our work builds a more comprehensive taxonomy through the inference on an existing taxonomy.

Taxonomy Induction Many induction approaches are also proposed to complete a taxonomy or find more hypernym relations. There are two lines of these work. The first line focuses on the inference mechanisms. For example, Snow et al. (Snow, Jurafsky, and Ng 2006) proposed a probabilistic inference framework, which seeks for the taxonomy that maximizes the likelihood of the *hypernym relations* and *coordinate relations* observed from corpora. Yang et al. (Yang and Callan 2009) proposed a series of information based metrics and proposed a minimum evolution principle for the taxonomy induction. The aim is to minimize the information change when the model tries a new taxonomy. Compared to these works, we focus on the inference of new hypernyms in terms of the transitivity of hypernym relations and most detailed signals are summarized from the structures of a taxonomy. The second line focuses on the induction from specific evidences. For example, Bansal et al. (Bansal et al. 2014) used Web n-grams and Wikipedia abstracts as data source for induction via a belief propagation method. As a most recent attempt, Zhang et al. (Zhang et al. 2016) jointly leveraged text and images to do taxonomy induction via a probabilistic Bayesian model. Compared to these work, the most remarkable difference of our work is that we infer a new taxonomy just from the existing taxonomy without external data source.

Transitivity on Semantic Networks Transitivity plays an important role in reasoning/induction/inference in semantic networks (Winston, Chaffin, and Herrmann 1987). Snow et al. (Snow, Jurafsky, and Ng 2006) used transitivity as one of constrains and proposed a probabilistic framework for taxonomy induction. The framework incorporates evidence from multiple classifiers over heterogeneous relationships to optimize the entire structure of the taxonomy. Cohen et al. (Cohen and Loiselle 1988) presented a method to derive plausible inference rules in knowledge bases; they use transitivity in the semantic network to test the plausibility of the derived rules. Fu et al. (Fu et al. 2014) proposed to use word embedding to construct a hierarchal taxonomy. In their method, they also used transitivity of "is-a" relations to identify potential "is-a" relations. Fallucchi et al. (Fallucchi and Zanzotto 2010) exploited transitivity to reinforce the confidence value of an "is-a" relation. They proposed two probabilistic models to exploit transitivity. Although transitivity is investigated in these related works, they in general did not consider the transitivity of hypernym-hyponym relationships in a data-driven taxonomy.

Conclusions

In this paper, we investigate the transitivity of hyponym relations in a data-driven knowledge base. We reveal that transitivity does not always hold in data-driven lexical taxonomies. We introduce a supervised approach to detect if transitivity holds for any given pair of hyponym relations. We propose a set of effective features for the transitivity inference, achieving 93% F1-score. Based on the transitivity inference, we find millions of new hyponym relations with 92.5% precision for a data-driven lexical taxonomy. These results justify the effectiveness of our approach.

References

Bansal, M.; Burkett, D.; De Melo, G.; and Klein, D. 2014. Structured learning for taxonomy induction with belief propagation. In *ACL (1)*, 1041–1051.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 1247–1250. ACM.

Cohen, P. R., and Loiselle, C. L. 1988. Beyond isa: Structures for plausible inference in semantic networks. In *AAAI*, 415–420.

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.

Fallucchi, F., and Zanzotto, F. M. 2010. Transitivity in semantic relation learning. In *NLP-KE*, 1–8. IEEE.

Fu, R.; Guo, J.; Qin, B.; Che, W.; Wang, H.; and Liu, T. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52th Annual Meeting of the ACL: Long Papers*, volume 1.

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

Hovy, E.; Kozareva, Z.; and Riloff, E. 2009. Toward completeness in concept extraction and classification. In *EMNLP*, 948–957. ACL.

Kozareva, Z.; Riloff, E.; and Hovy, E. H. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL*, volume 8, 1048–1056.

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.

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*.

Pantel, P., and Pennacchiotti, M. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *COLING*, 113–120. ACL.

Pantel, P.; Ravichandran, D.; and Hovy, E. 2004. Towards terascale knowledge acquisition. In *COLING*, 771. ACL.

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

Sang, E. T. K. 2007. Extracting hypernym pairs from the web. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 165–168. ACL.

Snow, R.; Jurafsky, D.; and Ng, A. Y. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

Snow, R.; Jurafsky, D.; and Ng, A. Y. 2006. Semantic taxonomy induction from heterogenous evidence. In *COLING*, 801–808. ACL.

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

Winston, M. E.; Chaffin, R.; and Herrmann, D. 1987. A taxonomy of part-whole relations. *Cognitive science* 11(4):417–444.

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

Yang, H., and Callan, J. 2009. A metric-based framework for automatic taxonomy induction. In *ACL-IJCNLP*, 271–279. ACL.

Zhang, H.; Hu, Z.; Deng, Y.; Sachan, M.; Yan, Z.; and Xing, E. P. 2016. Learning concept taxonomies from multi-modal data. *arXiv preprint arXiv:1606.09239*.