Nonlinear Evidence Fusion and Propagation for Hyponymy Relation Mining

Fan Zhang[2*] **Shuming Shi**[1] **Jing Liu**[2] **Shuqi Sun**[3*] **Chin-Yew Lin**[1]
[1]Microsoft Research Asia
[2]Nankai University, China
[3]Harbin Institute of Technology, China
{shumings, cyl}@microsoft.com

Abstract

This paper focuses on mining the hyponymy (or is-a) relation from large-scale, open-domain web documents. A nonlinear probabilistic model is exploited to model the correlation between sentences in the aggregation of pattern matching results. Based on the model, we design a set of evidence combination and propagation algorithms. These significantly improve the result quality of existing approaches. Experimental results conducted on 500 million web pages and hypernym labels for 300 terms show over 20% performance improvement in terms of P@5, MAP and R-Precision.

1 Introduction

An important task in text mining is the automatic extraction of entities and their lexical relations; this has wide applications in natural language processing and web search. This paper focuses on mining the hyponymy (or is-a) relation from large-scale, open-domain web documents. From the viewpoint of entity classification, the problem is to automatically assign *fine-grained* class labels to terms.

There have been a number of approaches (Hearst 1992; Pantel & Ravichandran 2004; Snow et al., 2005; Durme & Pasca, 2008; Talukdar et al., 2008) to address the problem. These methods typically exploited manually-designed or automatically-learned patterns (e.g., "*NP such as NP*", "*NP like NP*", "*NP is a NP*"). Although some degree of success has been achieved with these efforts, the results are still far from perfect, in terms of both recall and precision. As will be demonstrated in this paper, even by processing a large corpus of 500 million web pages with the most popular patterns, we are not able to extract correct labels for many (especially rare) entities. Even for popular terms, incorrect results often appear in their label lists.

The basic philosophy in existing hyponymy extraction approaches (and also many other text-mining methods) is *counting*: count the number of supporting sentences. Here a *supporting sentence* of a term-label pair is a sentence from which the pair can be extracted via an extraction pattern. We demonstrate that the specific way of counting has a great impact on result quality, and that the state-of-the-art counting methods are not optimal. Specifically, we examine the problem from the viewpoint of probabilistic evidence combination and find that the probabilistic assumption behind simple counting is the statistical *independence* between the observations of supporting sentences. By assuming a *positive correlation* between supporting sentence observations and adopting properly designed nonlinear combination functions, the results precision can be improved.

It is hard to extract correct labels for rare terms from a web corpus due to the data sparseness problem. To address this issue, we propose an evidence propagation algorithm motivated by the observation that similar terms tend to share common hypernyms. For example, if we already know that 1) Helsinki and Tampere are cities, and 2) Porvoo is similar to Helsinki and Tampere, then Porvoo is

[*] This work was performed when Fan Zhang and Shuqi Sun were interns at Microsoft Research Asia

very likely also a city. This intuition, however, does not mean that the labels of a term can always be transferred to its similar terms. For example, Mount Vesuvius and Kilimanjaro are volcanoes and Lhotse is similar to them, but Lhotse is not a volcano. Therefore we should be very conservative and careful in hypernym propagation. In our propagation algorithm, we first construct some *pseudo supporting sentences* for a term from the supporting sentences of its similar terms. Then we calculate label scores for terms by performing *nonlinear* evidence combination based on the (pseudo and real) supporting sentences. Such a nonlinear propagation algorithm is demonstrated to perform better than linear propagation.

Experimental results on a publicly available collection of 500 million web pages with hypernym labels annotated for 300 terms show that our nonlinear evidence fusion and propagation significantly improve the precision and coverage of the extracted hyponymy data. This is one of the technologies adopted in our semantic search and mining system *NeedleSeek*[2].

In the next section, we discuss major related efforts and how they differ from our work. Section 3 is a brief description of the baseline approach. The probabilistic evidence combination model that we exploited is introduced in Section 4. Our main approach is illustrated in Section 5. Section 6 shows our experimental settings and results. Finally, Section 7 concludes this paper.

2 Related Work

Existing efforts for hyponymy relation extraction have been conducted upon various types of data sources, including plain-text corpora (Hearst 1992; Pantel & Ravichandran, 2004; Snow et al., 2005; Snow et al., 2006; Banko, et al., 2007; Durme & Pasca, 2008; Talukdar et al., 2008), semi-structured web pages (Cafarella et al., 2008; Shinzato & Torisawa, 2004), web search results (Geraci et al., 2006; Kozareva et al., 2008; Wang & Cohen, 2009), and query logs (Pasca 2010). Our target for optimization in this paper is the approaches that use lexico-syntactic patterns to extract hyponymy relations from plain-text corpora. Our future work will study the application of the proposed algorithms on other types of approaches.

[2] http://research.microsoft.com/en-us/projects/needleseek/ or http://needleseek.msra.cn/

The probabilistic evidence combination model that we exploit here was first proposed in (Shi et al., 2009), for combining the page in-link evidence in building a nonlinear static-rank computation algorithm. We applied it to the hyponymy extraction problem because the model takes the dependency between supporting sentences into consideration and the resultant evidence fusion formulas are quite simple. In (Snow et al., 2006), a probabilistic model was adopted to combine evidence from heterogeneous relationships to jointly optimize the relationships. The independence of evidence was assumed in their model. In comparison, we show that better results will be obtained if the evidence correlation is modeled appropriately.

Our evidence propagation is basically about using term similarity information to help instance labeling. There have been several approaches which improve hyponymy extraction with instance clusters built by distributional similarity. In (Pantel & Ravichandran, 2004), labels were assigned to the committee (i.e., representative members) of a semantic class and used as the hypernyms of the whole class. Labels generated by their approach tend to be rather coarse-grained, excluding the possibility of a term having its private labels (considering the case that one meaning of a term is not covered by the input semantic classes). In contrast to their method, our label scoring and ranking approach is applied to every single term rather than a semantic class. In addition, we also compute label scores in a nonlinear way, which improves results quality. In Snow et al. (2005), a supervised approach was proposed to improve hypernym classification using coordinate terms. In comparison, our approach is unsupervised. Durme & Pasca (2008) cleaned the set of instance-label pairs with a TF*IDF like method, by exploiting clusters of semantically related phrases. The core idea is to keep a term-label pair (T, L) only if the number of terms having the label L in the term T's cluster is above a threshold and if L is not the label of too many clusters (otherwise the pair will be discarded). In contrast, we are able to add new (high-quality) labels for a term with our evidence propagation method. On the other hand, low quality labels get smaller score gains via propagation and are ranked lower.

Label propagation is performed in (Talukdar et al., 2008; Talukdar & Pereira, 2010) based on multiple instance-label graphs. Term similarity information was not used in their approach.

Most existing work tends to utilize small-scale or private corpora, whereas the corpus that we used is publicly available and much larger than most of the existing work. We published our term sets (refer to Section 6.1) and their corresponding user judgments so researchers working on similar topics can reproduce our results.

Type	Pattern
Hearst-I	NP_L {,} (such as) {NP,}* {and|or} NP
Hearst-II	NP_L {,} (include(s) | including) {NP,}* {and|or} NP
Hearst-III	NP_L {,} (e.g.|e.g) {NP,}* {and|or} NP
IsA-I	NP (is|are|was|were|being) (a|an) NP_L
IsA-II	NP (is|are|was|were|being) {the, those} NP_L
IsA-III	NP (is|are|was|were|being) {another, any} NP_L

Table 1. Patterns adopted in this paper (NP: named phrase representing an entity; NP_L: label)

3 Preliminaries

The problem addressed in this paper is corpus-based *is-a* relation mining: extracting hypernyms (as labels) for entities from a large-scale, open-domain document corpus. The desired output is a mapping from terms to their corresponding hypernyms, which can naturally be represented as a weighted bipartite graph (term-label graph). Typically we are only interested in top labels of a term in the graph.

Following existing efforts, we adopt pattern-matching as a basic way of extracting hypernymy/hyponymy relations. Two types of patterns (refer to Table 1) are employed, including the popular "Hearst patterns" (Hearst, 1992) and the IsA patterns which are exploited less frequently in existing hyponym mining efforts. One or more term-label pairs can be extracted if a pattern matches a sentence. In the baseline approach, the weight of an edge $T \rightarrow L$ (from term T to hypernym label L) in the term-label graph is computed as,

$$w(T \rightarrow L) = m \cdot \mathrm{IDF}(L) = m \cdot \log \frac{1+N}{1+\mathrm{DF}(L)} \qquad (3.1)$$

where m is the number of times the pair (T, L) is extracted from the corpus, $\mathrm{DF}(L)$ is the number of in-links of L in the graph, N is total number of terms in the graph, and IDF means the "inverse document frequency".

A term can only keep its top-k neighbors (according to the edge weight) in the graph as its final labels.

Our pattern matching algorithm implemented in this paper uses part-of-speech (POS) tagging information, without adopting a parser or a chunker. The noun phrase boundaries (for terms and labels) are determined by a manually designed POS tag list.

4 Probabilistic Label-Scoring Model

Here we model the hyponymy extraction problem from the probability theory point of view, aiming at estimating the score of a term-label pair (i.e., the score of a label w.r.t. a term) with probabilistic evidence combination. The model was studied in (Shi et al., 2009) to combine the page in-link evidence in building a nonlinear static-rank computation algorithm.

We represent the score of a term-label pair by the probability of the label being a correct hypernym of the term, and define the following events,

$A_{T,L}$: Label L is a hypernym of term T (the abbreviated form A is used in this paper unless it is ambiguous).

E_i: The observation that (T, L) is extracted from a sentence S_i via pattern matching (i.e., S_i is a supporting sentence of the pair).

Assuming that we already know m supporting sentences ($S_1 \sim S_m$), our problem is to compute $P(A|E_1,E_2,..,E_m)$, the *posterior* probability that L is a hypernym of term T, given evidence $E_1 \sim E_m$. Formally, we need to find a function f to satisfy,

$$P(A|E_1,\ldots,E_m) = f(P(A), P(A|E_1)\ldots, P(A|E_m)) \qquad (4.1)$$

For simplicity, we first consider the case of m=2. The case of m>2 is quite similar.

We start from the simple case of independent supporting sentences. That is,

$$P(E_1, E_2) = P(E_1) \cdot P(E_2) \qquad (4.2)$$

$$P(E_1, E_2|A) = P(E_1|A) \cdot P(E_2|A) \qquad (4.3)$$

By applying Bayes rule, we get,

$$\begin{aligned} P(A|E_1, E_2) &= \frac{P(E_1, E_2|A) \cdot P(A)}{P(E_1, E_2)} \\ &= \frac{P(E_1|A) \cdot P(A)}{P(E_1)} \cdot \frac{P(E_2|A) \cdot P(A)}{P(E_2)} \cdot \frac{1}{P(A)} \\ &= \frac{P(A|E_1) \cdot P(A|E_2)}{P(A)} \end{aligned} \qquad (4.4)$$

Then define

$$G(A|E) = \log \frac{P(A|E)}{P(A)} = \log(P(A|E)) - log(\mathrm{P}(A))$$

Here $G(A|E)$ represents the *log-probability-gain* of A given E, with the meaning of the *gain* in the log-probability value of A after the evidence E is observed (or known). It is a measure of the impact of evidence E to the probability of event A. With the definition of $G(A|E)$, Formula 4.4 can be transformed to,

$$G(A|E_1, E_2) = G(A|E_1) + G(A|E_2) \quad (4.5)$$

Therefore, if E_1 and E_2 are independent, the log-probability-gain of A given both pieces of evidence will exactly be the sum of the gains of A given every single piece of evidence respectively. It is easy to prove (by following a similar procedure) that the above Formula holds for the case of m>2, as long as the pieces of evidence are mutually independent.

Therefore for a term-label pair with m mutually independent supporting sentences, if we set every gain $G(A|E_i)$ to be a constant value g, the posterior gain score of the pair will be $\sum_{i=1}^{m} g = mg$. If the value g is the IDF of label L, the posterior gain will be,

$$G(A_{T,L}|E_1 \ldots, E_m) = \sum_{i=1}^{m} \mathrm{IDF}(L) = m \cdot \mathrm{IDF}(L) \quad (4.6)$$

This is exactly the Formula 3.1. By this way, we provide a probabilistic explanation of scoring the candidate labels for a term via simple counting.

	Hearst-I	**IsA-I**	**E_1: Hearst-I E_2: IsA-I**
R_A: $\frac{P(E_1,E_2\|A)}{P(E_1\|A)P(E_2\|A)}$	66.87	17.30	24.38
R: $\frac{P(E_1,E_2)}{P(E_1)P(E_2)}$	5997	1711	802.7
R_A/R	0.011	0.010	0.030

Table 2. Evidence dependency estimation for intra-pattern and inter-pattern supporting sentences

In the above analysis, we assume the statistical independence of the supporting sentence observations, which may not hold in reality. Intuitively, if we already know one supporting sentence S_1 for a term-label pair (T, L), then we have more chance to find another supporting sentence than if we do not know S_1. The reason is that, before we find S_1, we have to estimate the probability with the chance of discovering a supporting sentence for a *random* term-label pair. The probability is quite low because most term-label pairs do not have hyponymy relations. Once we have observed S_1, however, the chance of (T, L) having a hyponymy relation increases. Therefore the chance of observing another supporting sentence becomes larger than before.

Table 2 shows the rough estimation of $\frac{P(E_1,E_2|A)}{P(E_1|A)P(E_2|A)}$ (denoted as R_A), $\frac{P(E_1,E_2)}{P(E_1)P(E_2)}$ (denoted as R), and their ratios. The statistics are obtained by performing maximal likelihood estimation (MLE) upon our corpus and a random selection of term-label pairs from our term sets (see Section 6.1) together with their top labels[3]. The data verifies our analysis about the correlation between E_1 and E_2 (note that R=1 means independent). In addition, it can be seen that the conditional independence assumption of Formula 4.3 does not hold (because R_A>1). It is hence necessary to consider the correlation between supporting sentences in the model. The estimation of Table 2 also indicates that,

$$\frac{P(E_1, E_2)}{P(E_1)P(E_2)} > \frac{P(E_1, E_2|A)}{P(E_1|A)P(E_2|A)} \quad (4.7)$$

By following a similar procedure as above, with Formulas 4.2 and 4.3 replaced by 4.7, we have,

$$G(A|E_1, E_2) < G(A|E_1) + G(A|E_2) \quad (4.8)$$

This formula indicates that when the supporting sentences are positively correlated, the posterior score of label L w.r.t. term T (given both the sentences) is smaller than the sum of the gains caused by one sentence only. In the extreme case that sentence S_2 fully depends on E_1 (i.e. $P(E_2|E_1)$=1), it is easy to prove that

$$G(A|E_1, E_2) = G(A|E_1)$$

It is reasonable, since event E_2 does not bring in more information than E_1.

Formula 4.8 cannot be used directly for computing the posterior gain. What we really need is a function h satisfying

$$G(A|E_1, \ldots, E_m) = h\big(G(A|E_1), \ldots, G(A|E_m)\big) \quad (4.9)$$

and

$$h(x_1 \ldots, x_m) < \sum_{i=1}^{m} x_i \quad (4.10)$$

Shi et al. (2009) discussed other constraints to h and suggested the following nonlinear functions,

$$h_1(x_1 \ldots, x_m) = \ln(1 + \sum_{i=1}^{m} (\mathrm{e}^{x_i} - 1)) \quad (4.11)$$

[3] R_A is estimated from the labels judged as "Good"; whereas the estimation of R is from all judged labels.

$$h_2(x_1 \dots, x_m) = \sqrt[p]{\sum_{i=1}^{m} x_i^p} \qquad (p>1) \qquad (4.12)$$

In the next section, we use the above two *h* functions as basic building blocks to compute label scores for terms.

5 Our Approach

Multiple types of patterns (Table 1) can be adopted to extract term-label pairs. For two supporting sentences the correlation between them may depend on whether they correspond to the same pattern. In Section 5.1, our nonlinear evidence fusion formulas are constructed by making specific assumptions about the correlation between intra-pattern supporting sentences and inter-pattern ones.

Then in Section 5.2, we introduce our evidence propagation technique in which the evidence of a (T, L) pair is propagated to the terms similar to *T*.

5.1 Nonlinear evidence fusion

For a term-label pair (*T*, *L*), assuming *K* patterns are used for hyponymy extraction and the supporting sentences discovered with pattern *i* are,

$$S_{i,1}, S_{i,2} \dots, S_{i,m_i} \qquad (5.1)$$

where m_i is the number of supporting sentences corresponding to pattern *i*. Also assume the gain score of $S_{i,j}$ is $x_{i,j}$, i.e., $x_{i,j}=G(A|S_{i,j})$.

Generally speaking, supporting sentences corresponding to the same pattern typically have a higher correlation than the sentences corresponding to different patterns. This can be verified by the data in Table-2. By ignoring the inter-pattern correlations, we make the following simplified assumption:

Assumption: Supporting sentences corresponding to the same pattern are correlated, while those of different patterns are independent.

According to this assumption, our label-scoring function is,

$$score(T, L) = \sum_{i=1}^{K} h(x_{i,1}, x_{i,2} \dots, x_{i,m_i}) \qquad (5.2)$$

In the simple case that $x_{i,j} = \mathrm{IDF}(L)$, if the *h* function of Formula 4.12 is adopted, then,

$$Score(T, L) = \left(\sum_{i=1}^{K} \sqrt[p]{m_i} \right) \cdot \mathrm{IDF}(L) \qquad (5.3)$$

We use an example to illustrate the above formula.

Example: For term *T* and label L_1, assume the numbers of the supporting sentences corresponding to the six pattern types in Table 1 are (4, 4, 4, 4, 4, 4), which means the number of supporting sentences discovered by each pattern type is 4. Also assume the supporting-sentence-count vector of label L_2 is (25, 0, 0, 0, 0, 0). If we use Formula 5.3 to compute the scores of L_1 and L_2, we can have the following (ignoring IDF for simplicity),

$Score(L_1) = 6 \cdot \sqrt{4} = 12$; $Score(L_2) = \sqrt{25} = 5$

One the other hand, if we simply count the total number of supporting sentences, the score of L_2 will be larger.

The rationale implied in the formula is: For a given term *T*, the labels supported by multiple types of patterns tend to be more reliable than those supported by a single pattern type, if they have the same number of supporting sentences.

5.2 Evidence propagation

According to the evidence fusion algorithm described above, in order to extract term labels reliably, it is desirable to have many supporting sentences of different types. This is a big challenge for rare terms, due to their low frequency in sentences (and even lower frequency in supporting sentences because not all occurrences can be covered by patterns). With evidence propagation, we aim at discovering more supporting sentences for terms (especially rare terms). Evidence propagation is motivated by the following two observations:

(I) Similar entities or coordinate terms tend to share some common hypernyms.

(II) Large term similarity graphs are able to be built efficiently with state-of-the-art techniques (Agirre et al., 2009; Pantel et al., 2009; Shi et al., 2010). With the graphs, we can obtain the similarity between two terms without their hypernyms being available.

The first observation motivates us to "*borrow*" the supporting sentences from other terms as auxiliary evidence of the term. The second observation means that *new* information is brought with the state-of-the-art term similarity graphs (in addition to the term-label information discovered with the patterns of Table 1).

Our evidence propagation algorithm contains two phases. In phase I, some *pseudo supporting sentences* are constructed for a term from the supporting sentences of its neighbors in the similarity graph. Then we calculate the label scores for terms based on their (pseudo and real) supporting sentences.

Phase I: For every supporting sentence S and every similar term T_1 of the term T, add a pseudo supporting sentence S_1 for T_1, with the gain score,

$$G(A_{T_1,L_1}|S_1) = \mu \cdot \mathrm{Sim}(T,T_1) \cdot G(A_{T,L}|S) \qquad (5.5)$$

where $\mu \in [0,1]$ is the propagation factor, and $\mathrm{Sim}(\cdot,\cdot)$ is the term similarity function taking values in [0, 1]. The formula reasonably assumes that the gain score of the pseudo supporting sentence depends on the gain score of the original real supporting sentence, the similarity between the two terms, and the propagation factor.

Phase II: The nonlinear evidence combination formulas in the previous subsection are adopted to combine the evidence of pseudo supporting sentences.

Term similarity graphs can be obtained by distributional similarity or patterns (Agirre et al., 2009; Pantel et al., 2009; Shi et al., 2010). We call the first type of graph *DS* and the second type *PB*. DS approaches are based on the distributional hypothesis (Harris, 1985), which says that terms appearing in analogous contexts tend to be similar. In a DS approach, a term is represented by a feature vector, with each feature corresponding to a context in which the term appears. The similarity between two terms is computed as the similarity between their corresponding feature vectors. In PB approaches, a list of carefully-designed (or automatically learned) patterns is exploited and applied to a text collection, with the hypothesis that the terms extracted by applying each of the patterns to a specific piece of text tend to be similar. Two categories of patterns have been studied in the literature (Heast 1992; Pasca 2004; Kozareva et al., 2008; Zhang et al., 2009): sentence lexical patterns, and HTML tag patterns. An example of sentence lexical patterns is "*T {, T}*{,} (and|or) T*". HTML tag patterns include HTML tables, drop-down lists, and other tag repeat patterns. In this paper, we generate the DS and PB graphs by adopting the best-performed methods studied in (Shi et al., 2010). We will compare, by experiments, the propagation performance of utilizing the two categories of graphs, and also investigate the performance of utilizing both graphs for evidence propagation.

6 Experiments

6.1 Experimental setup

Corpus We adopt a publicly available dataset in our experiments: ClueWeb09[4]. This is a very large dataset collected by Carnegie Mellon University in early 2009 and has been used by several tracks of the Text Retrieval Conference (TREC)[5]. The whole dataset consists of 1.04 billion web pages in ten languages while only those in English, about 500 million pages, are used in our experiments. The reason for selecting such a dataset is twofold: First, it is a corpus large enough for conducting web-scale experiments and getting meaningful results. Second, since it is publicly available, it is possible for other researchers to reproduce the experiments in this paper.

Term sets Approaches are evaluated by using two sets of selected terms: *Wiki200*, and *Ext100*. For every term in the term sets, each approach generates a list of hypernym labels, which are manually judged by human annotators. Wiki200 is constructed by first randomly selecting 400 Wikipedia[6] titles as our candidate terms, with the probability of a title T being selected to be $\log(1 + F(T))$, where $F(T)$ is the frequency of T in our data corpus. The reason of adopting such a probability formula is to balance popular terms and rare ones in our term set. Then 200 terms are manually selected from the 400 candidate terms, with the principle of maximizing the diversity of terms in terms of length (i.e., number of words) and type (person, location, organization, software, movie, song, animal, plant, etc.). Wiki200 is further divided into two subsets: *Wiki100H* and *Wiki100L*, containing respectively the 100 high-frequency and low-frequency terms. Ext100 is built by first selecting 200 *non*-Wikipedia-title terms at random from the term-label graph generated by the baseline approach (Formula 3.1), then manually selecting 100 terms.

Some sample terms in the term sets are listed in Table 3.

[4] http://boston.lti.cs.cmu.edu/Data/clueweb09/
[5] http://trec.nist.gov/
[6] http://www.wikipedia.org/

Term Set	Sample Terms
Wiki200	Canon EOS 400D, Disease management, El Salvador, Excellus Blue Cross Blue Shield, F33, Glasstron, Indium, Khandala, Kung Fu, Lake Greenwood, Le Gris, Liriope, Lionel Barrymore, Milk, Mount Alto, Northern Wei, Pink Lady, Shawshank, The Dog Island, White flight, World War II…
Ext100	A2B, Antique gold, GPTEngine, Jinjiang Inn, Moyea SWF to Apple TV Converter, Nanny service, Outdoor living, Plasmid DNA, Popon, Spam detection, Taylor Ho Bynum, Villa Michelle…

Table 3. Sample terms in our term sets

Annotation For each term in the term set, the top-5 results (i.e., hypernym labels) of various methods are mixed and judged by human annotators. Each annotator assigns each result item a judgment of "Good", "Fair" or "Bad". The annotators do not know the method by which a result item is generated. Six annotators participated in the labeling with a rough speed of 15 minutes per term. We also encourage the annotators to add new good results which are not discovered by any method.

The term sets and their corresponding user annotations are available for download at the following links (dataset ID=data.queryset.semcat01):

http://research.microsoft.com/en-us/projects/needleseek/
http://needleseek.msra.cn/datasets/

Evaluation We adopt the following metrics to evaluate the hypernym list of a term generated by each method. The evaluation score on a term set is the average over all the terms.

Precision@*k*: The percentage of relevant (good or fair) labels in the top-*k* results (labels judged as "Fair" are counted as 0.5)

Recall@k: The ratio of relevant labels in the top-*k* results to the total number of relevant labels

R-Precision: Precision@R where *R* is the total number of labels judged as "Good"

Mean average precision (*MAP*): The average of precision values at the positions of all good or fair results

Before annotation and evaluation, the hypernym list generated by each method for each term is preprocessed to remove *duplicate* items. Two hypernyms are called *duplicate* items if they share the same head word (e.g., "military conflict" and "conflict"). For duplicate hypernyms, only the first (i.e., the highest ranked one) in the list is kept. The goal with such a preprocessing step is to partially consider results diversity in evaluation and to make a more meaningful comparison among different methods. Consider two hypernym lists for "subway":

List-1: restaurant; chain restaurant; worldwide chain restaurant; franchise; restaurant franchise…

List-2: restaurant; franchise; transportation; company; fast food…

There are more detailed hypernyms in the first list about "subway" as a restaurant or a franchise; while the second list covers a broader range of meanings for the term. It is hard to say which is better (without considering the upper-layer applications). With this preprocessing step, we keep our focus on short hypernyms rather than detailed ones.

Term Set	Method	MAP	R-Prec	P@1	P@5
Wiki200	Linear	0.357	0.376	0.783	0.547
	Log	0.371 ↑3.92%	0.384 ↑2.13%	0.803 ↑2.55%	0.561 ↑2.56%
	PNorm	0.372 ↑4.20%	0.384 ↑2.13%	0.800 ↑2.17%	0.562 ↑2.74%
Wiki100H	Linear	0.363	0.382	0.805	0.627
	Log	0.393 ↑8.26%	0.402 ↑5.24%	0.845 ↑4.97%	0.660 ↑5.26%
	PNorm	0.395 ↑8.82%	0.403 ↑5.50%	0.840 ↑4.35%	0.662 ↑5.28%

Table 4. Performance comparison among various evidence fusion methods (Term sets: Wiki200 and Wiki100H; p=2 for PNorm)

6.2 Experimental results

We first compare the evaluation results of different evidence fusion methods mentioned in Section 4.1. In Table 4, *Linear* means that Formula 3.1 is used to calculate label scores, whereas *Log* and *PNorm* represent our nonlinear approach with Formulas 4.11 and 4.12 being utilized. The performance improvement numbers shown in the table are based on the linear version; and the upward pointing arrows indicate relative percentage improvement over the baseline. From the table, we can see that the nonlinear methods outperform the linear ones on the Wiki200 term set. It is interesting to note that the performance improvement is more significant on Wiki100H, the set of high frequency terms. By examining the labels and supporting sentences for the terms in each term set, we find that for many low-frequency terms (in Wiki100L), there are only a few supporting sentences (corresponding

to one or two patterns). So the scores computed by various fusion algorithms tend to be similar. In contrast, more supporting sentences can be discovered for high-frequency terms. Much information is contained in the sentences about the hypernyms of the high-frequency terms, but the linear function of Formula 3.1 fails to make effective use of it. The two nonlinear methods achieve better performance by appropriately modeling the dependency between supporting sentences and computing the log-probability gain in a better way.

The comparison of the linear and nonlinear methods on the Ext100 term set is shown in Table 5. Please note that the terms in Ext100 do not appear in Wikipedia titles. Thanks to the scale of the data corpus we are using, even the baseline approach achieves reasonably good performance. Please note that the terms (refer to Table 3) we are using are "harder" than those adopted for evaluation in many existing papers. Again, the results quality is improved with the nonlinear methods, although the performance improvement is not big due to the reason that most terms in Ext100 are rare. Please note that the recall (R@1, R@5) in this paper is pseudo-recall, i.e., we treat the number of *known* relevant (Good or Fair) results as the total number of relevant ones.

Method	MAP	R-Prec	P@1	P@5	R@1	R@5
Linear	0.384	0.429	0.665	0.472	0.116	0.385
Log	0.395 ↑2.86%	0.429 ↑0%	0.715 ↑7.52%	0.472 ↑0%	0.125 ↑7.76%	0.385 ↑0%
PNorm	0.390 ↑1.56%	0.429 ↑0%	0.700 ↑5.26%	0.472 ↑0%	0.120 ↑3.45%	0.385 ↑0%

Table 5. Performance comparison among various evidence fusion methods (Term set: Ext100; p=2 for PNorm)

The parameter p in the PNorm method is related to the degree of correlations among supporting sentences. The linear method of Formula 3.1 corresponds to the special case of p=1; while p=∞ represents the case that other supporting sentences are fully correlated to the supporting sentence with the maximal log-probability gain. Figure 1 shows that, for most of the term sets, the best performance is obtained for $p \in$[2.0, 4.0]. The reason may be that the sentence correlations are better estimated with p values in this range.

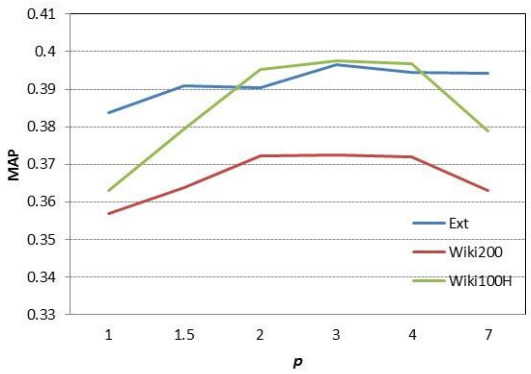

Figure 1. Performance curves of PNorm with different parameter values (Measure: MAP)

The experimental results of evidence propagation are shown in Table 6. The methods for comparison are,

Base: The linear function without propagation.

NL: Nonlinear evidence fusion (PNorm with p=2) without propagation.

LP: Linear propagation, i.e., the linear function is used to combine the evidence of pseudo supporting sentences.

NLP: Nonlinear propagation where PNorm (p=2) is used to combine the pseudo supporting sentences.

NL+NLP: The nonlinear method is used to combine both supporting sentences and pseudo supporting sentences.

Method	MAP	R-Prec	P@1	P@5	R@5
Base	0.357	0.376	0.783	0.547	0.317
NL	0.372 ↑4.20%	0.384 ↑2.13%	0.800 ↑2.17%	0.562 ↑2.74%	0.325 ↑2.52%
LP	0.357 ↑0%	0.376 ↑0%	0.783 ↑0%	0.547 ↑0%	0.317 ↑0%
NLP	0.396 ↑10.9%	0.418 ↑11.2%	0.785 ↑0.26%	0.605 ↑10.6%	0.357 ↑12.6%
NL+NLP	0.447 ↑25.2%	0.461 ↑22.6%	0.840 ↑7.28%	0.667 ↑21.9%	0.404 ↑27.4%

Table 6. Evidence propagation results (Term set: Wiki200; Similarity graph: PB; Nonlinear formula: PNorm)

In this paper, we generate the DS (distributional similarity) and PB (pattern-based) graphs by adopting the best-performed methods studied in (Shi et al., 2010). The performance improvement numbers (indicated by the upward pointing arrows) shown in tables 6~9 are relative percentage improvement

over the *base* approach (i.e., linear function without propagation). The values of parameter μ are set to maximize the MAP values.

Several observations can be made from Table 6. First, no performance improvement can be obtained with the linear propagation method (LP), while the nonlinear propagation algorithm (NLP) works quite well in improving both precision and recall. The results demonstrate the high correlation between pseudo supporting sentences and the great potential of using term similarity to improve hypernymy extraction. The second observation is that the NL+NLP approach achieves a much larger performance improvement than NL and NLP. Similar results (omitted due to space limitation) can be observed on the Ext100 term set.

Method	MAP	R-Prec	P@1	P@5	R@5
Base	0.357	0.376	0.783	0.547	0.317
NL+NLP (PB)	0.415 ↑16.2%	0.439 ↑16.8%	0.830 ↑6.00%	0.633 ↑15.7%	0.379 ↑19.6%
NL+NLP (DS)	0.456 ↑27.7%	0.469 ↑24.7%	0.843 ↑7.66%	0.673 ↑23.0%	0.406 ↑28.1%
NL+NLP (PB+DS)	0.473 ↑32.5%	0.487 ↑29.5%	0.860 ↑9.83%	0.700 ↑28.0%	0.434 ↑36.9%

Table 7. Combination of PB and DS graphs for evidence propagation (Term set: Wiki200; Nonlinear formula: Log)

Method	MAP	R-Prec	P@1	P@5	R@5
Base	0.351	0.370	0.760	0.467	0.317
NL+NLP (PB)	0.411 ↑17.1%	0.448 ↑21.1%	0.770 ↑1.32%	0.564 ↑20.8%	0.401 ↑26.5%
NL+NLP (DS)	0.469 ↑33.6%	0.490 ↑32.4%	0.815 ↑7.24%	0.622 ↑33.2%	0.438 ↑38.2%
NL+NLP (PB+DS)	0.491 ↑39.9%	0.513 ↑38.6%	0.860 ↑13.2%	0.654 ↑40.0%	0.479 ↑51.1%

Table 8. Combination of PB and DS graphs for evidence propagation (Term set: Wiki100L)

Now let us study whether it is possible to combine the PB and DS graphs to obtain better results. As shown in Tables 7, 8, and 9 (for term sets Wiki200, Wiki100L, and Ext100 respectively, using the *Log* formula for fusion and propagation), utilizing both graphs really yields additional performance gains. We explain this by the fact that the information in the two term similarity graphs tends to be complimentary. The performance improvement over Wiki100L is especially remarkable. This is reasonable because rare terms do not have adequate information in their supporting sentences due to data sparseness. As a result, they benefit the most from the pseudo supporting sentences propagated with the similarity graphs.

Method	MAP	R-Prec	P@1	P@5	R@5
Base	0.384	0.429	0.665	0.472	0.385
NL+NLP (PB)	0.454 ↑18.3%	0.479 ↑11.7%	0.745 ↑12.0%	0.550 ↑16.5%	0.456 ↑18.4%
NL+NLP (DS)	0.404 ↑5.18%	0.441 ↑2.66%	0.720 ↑8.27%	0.486 ↑2.97%	0.402 ↑4.37%
NL+NLP(PB+DS)	0.483 ↑26.0%	0.518 ↑20.6%	0.760 ↑14.3%	0.586 ↑24.2%	0.492 ↑27.6%

Table 9. Combination of PB and DS graphs for evidence propagation (Term set: Ext100)

7 Conclusion

We demonstrated that the way of aggregating supporting sentences has considerable impact on results quality of the hyponym extraction task using lexico-syntactic patterns, and the widely-used counting method is not optimal. We applied a series of nonlinear evidence fusion formulas to the problem and saw noticeable performance improvement. The data quality is improved further with the combination of nonlinear evidence fusion and evidence propagation. We also introduced a new evaluation corpus with annotated hypernym labels for 300 terms, which were shared with the research community.

Acknowledgments

We would like to thank Matt Callcut for reading through the paper. Thanks to the annotators for their efforts in judging the hypernym labels. Thanks to Yueguo Chen, Siyu Lei, and the anonymous reviewers for their helpful comments and suggestions. The first author is partially supported by the NSF of China (60903028,61070014), and Key Projects in the Tianjin Science and Technology Pillar Program.

References

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa. 2009. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In Proc. of NAACL-HLT'2009.

M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open Information Extraction from the Web. In Proc. of IJCAI'2007.

M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. 2008. WebTables: Exploring the Power of Tables on the Web. In Proceedings of the 34th Conference on Very Large Data Bases (VLDB'2008), pages 538–549, Auckland, New Zealand.

B. Van Durme and M. Pasca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. Twenty-Third AAAI Conference on Artificial Intelligence.

F. Geraci, M. Pellegrini, M. Maggini, and F. Sebastiani. 2006. Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. In Proceedings of the 13th Conference on String Processing and Information Retrieval (SPIRE'2006), pages 25–36, Glasgow, Scotland.

Z. S. Harris. 1985. Distributional Structure. The Philosophy of Linguistics. New York: Oxford University Press.

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In Fourteenth International Conference on Computational Linguistics, Nantes, France.

Z. Kozareva, E. Riloff, E.H. Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In Proc. of ACL'2008.

P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu and V. Vyas. 2009. Web-Scale Distributional Similarity and Entity Set Expansion. EMNLP'2009. Singapore.

P. Pantel and D. Ravichandran. 2004. Automatically Labeling Semantic Classes. In Proc. of the 2004 Human Language Technology Conference (HLT-NAACL'2004), 321–328.

M. Pasca. 2004. Acquisition of Categorized Named Entities for Web Search. In Proc. of CIKM'2004.

M. Pasca. 2010. The Role of Queries in Ranking Labeled Instances Extracted from Text. In Proc. of COLING'2010, Beijing, China.

S. Shi, B. Lu, Y. Ma, and J.-R. Wen. 2009. Nonlinear Static-Rank Computation. In Proc. of CIKM'2009, Kong Kong.

S. Shi, H. Zhang, X. Yuan, J.-R. Wen. 2010. Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches. In Proc. of COLING'2010, Beijing, China.

K. Shinzato and K. Torisawa. 2004. Acquiring Hyponymy Relations from Web Documents. In Proc. of the 2004 Human Language Technology Conference (HLT-NAACL'2004).

R. Snow, D. Jurafsky, and A. Y. Ng. 2005. Learning Syntactic Patterns for Automatic Hypernym Discovery. In Proceedings of the 19th Conference on Neural Information Processing Systems.

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06), 801–808.

P. P. Talukdar and F. Pereira. 2010. Experiments in Graph-based Semi-Supervised Learning Methods for Class-Instance Acquisition. In 48th Annual Meeting of the Association for Computational Linguistics (ACL'2010).

P. P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. 2008. Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'2008), pages 581–589.

R.C. Wang. W.W. Cohen. Automatic Set Instance Extraction using the Web. In Proc. of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP'2009), pages 441–449, Singapore.

H. Zhang, M. Zhu, S. Shi, and J.-R. Wen. 2009. Employing Topic Models for Pattern-based Semantic Class Discovery. In Proc. of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP'2009), pages 441–449, Singapore.