# Semi-supervised Relation Extraction with Label Propagation

**Jinxiu Chen**[1] **Donghong Ji**[1] **Chew Lim Tan**[2] **Zhengyu Niu**[1]

[1]Institute for Infocomm Research
21 Heng Mui Keng Terrace
119613 Singapore
{jinxiu,dhji,zniu}@i2r.a-star.edu.sg

[2]Department of Computer Science
National University of Singapore
117543 Singapore
tancl@comp.nus.edu.sg

## Abstract

To overcome the problem of not having enough manually labeled relation instances for supervised relation extraction methods, in this paper we propose a label propagation (LP) based semi-supervised learning algorithm for relation extraction task to learn from both labeled and unlabeled data. Evaluation on the ACE corpus showed when only a few labeled examples are available, our LP based relation extraction can achieve better performance than SVM and another bootstrapping method.

## 1 Introduction

Relation extraction is the task of finding relationships between two entities from text. For the task, many machine learning methods have been proposed, including supervised methods (Miller et al., 2000; Zelenko et al., 2002; Culotta and Soresen, 2004; Kambhatla, 2004; Zhou et al., 2005), semi-supervised methods (Brin, 1998; Agichtein and Gravano, 2000; Zhang, 2004), and unsupervised method (Hasegawa et al., 2004).

Supervised relation extraction achieves good performance, but it requires a large amount of manually labeled relation instances. Unsupervised methods do not need the definition of relation types and manually labeled data, but it is difficult to evaluate the clustering result since there is no relation type label for each instance in clusters. Therefore, semi-supervised learning has received attention, which can minimize corpus annotation requirement.

Current works on semi-supervised resolution for relation extraction task mostly use the bootstrapping algorithm, which is based on a **local consistency assumption**: examples close to labeled examples within the same class will have the same labels. Such methods ignore considering the similarity between unlabeled examples and do not perform classification from a global consistency viewpoint, which may fail to exploit appropriate manifold structure in data when training data is limited.

The objective of this paper is to present a label propagation based semi-supervised learning algorithm (LP algorithm) (Zhu and Ghahramani, 2002) for Relation Extraction task. This algorithm works by representing labeled and unlabeled examples as vertices in a connected graph, then propagating the label information from any vertex to nearby vertices through weighted edges iteratively, finally inferring the labels of unlabeled examples after the propagation process converges. Through the label propagation process, our method can make the best of the information of labeled and unlabeled examples to realize a **global consistency assumption**: similar examples should have similar labels. In other words, the labels of unlabeled examples are determined by considering not only the similarity between labeled and unlabeled examples, but also the similarity between unlabeled examples.

## 2 The Proposed Method

### 2.1 Problem Definition

Let $X = \{x_i\}_{i=1}^n$ be a set of contexts of occurrences of all entity pairs, where $x_i$ represents the contexts of the $i$-th occurrence, and $n$ is the total number of occurrences of all entity pairs. The first $l$ examples are labeled as $y_g$ ( $y_g \in \{r_j\}_{j=1}^R$, $r_j$ denotes relation type and $R$ is the total number of relation types). And the remaining $u(u = n - l)$ examples are unlabeled.

Intuitively, if two occurrences of entity pairs have

*Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 25–28,
New York, June 2006. ©2006 Association for Computational Linguistics

the similar contexts, they tend to hold the same relation type. Based on this assumption, we create a graph where the vertices are all the occurrences of entity pairs, both labeled and unlabeled. The edge between vertices represents their similarity. Then the task of relation extraction can be formulated as a form of propagation on a graph, where a vertex's label propagates to neighboring vertices according to their proximity. Here, the graph is connected with the weights: $W_{ij} = exp(-\frac{s_{ij}^2}{\alpha^2})$, where $s_{ij}$ is the similarity between $x_i$ and $x_j$ calculated by some similarity measures. In this paper,two similarity measures are investigated, i.e. Cosine similarity measure and Jensen-Shannon (JS) divergence (Lin, 1991). And we set $\alpha$ as the average similarity between labeled examples from different classes.

### 2.2 Label Propagation Algorithm

Given such a graph with labeled and unlabeled vertices, we investigate the label propagation algorithm (Zhu and Ghahramani, 2002) to help us propagate the label information of any vertex in the graph to nearby vertices through weighted edges until a global stable stage is achieved.

Define a $n \times n$ probabilistic transition matrix $T$ $T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}}$, where $T_{ij}$ is the probability to jump from vertex $x_j$ to vertex $x_i$. Also define a $n \times R$ label matrix $Y$, where $Y_{ij}$ representing the probabilities of vertex $y_i$ to have the label $r_j$.

Then the label propagation algorithm consists the following main steps:

**Step1: Initialization** Firstly, set the iteration index $t = 0$. Then let $Y^0$ be the initial soft labels attached to each vertex and $Y_L^0$ be the top $l$ rows of $Y^0$, which is consistent with the labeling in labeled data ($Y_{ij}^0 = 1$ if $y_i$ is label $r_j$ and 0 otherwise ). Let $Y_U^0$ be the remaining $u$ rows corresponding to unlabeled data points and its initialization can be arbitrary.

**Step 2: Propagate the label by $Y^{t+1} = \overline{T}Y^t$,** where $\overline{T}$ is the row-normalized matrix of $T$, i.e. $\overline{T_{ij}} = T_{ij}/\sum_k T_{ik}$, which can maintain the class probability interpretation.

**Step 3: Clamp the labeled data**, i.e., replace the top $l$ row of $Y^{t+1}$ with $Y_L^0$. In this step, the labeled data is clamped to replenish the label sources from these labeled data. Thus the labeled data act like sources to push out labels through unlabeled data.

Table 1: Frequency of Relation SubTypes in the ACE training and devtest corpus.

| Type | SubType | Training | Devtest |
|---|---|---|---|
| ROLE | General-Staff | 550 | 149 |
| | Management | 677 | 122 |
| | Citizen-Of | 127 | 24 |
| | Founder | 11 | 5 |
| | Owner | 146 | 15 |
| | Affiliate-Partner | 111 | 15 |
| | Member | 460 | 145 |
| | Client | 67 | 13 |
| | Other | 15 | 7 |
| PART | Part-Of | 490 | 103 |
| | Subsidiary | 85 | 19 |
| | Other | 2 | 1 |
| AT | Located | 975 | 192 |
| | Based-In | 187 | 64 |
| | Residence | 154 | 54 |
| SOC | Other-Professional | 195 | 25 |
| | Other-Personal | 60 | 10 |
| | Parent | 68 | 24 |
| | Spouse | 21 | 4 |
| | Associate | 49 | 7 |
| | Other-Relative | 23 | 10 |
| | Sibling | 7 | 4 |
| | GrandParent | 6 | 1 |
| NEAR | Relative-Location | 88 | 32 |

**Step 4: Repeat from step 2 until $Y$ converges.**

**Step 5: Assign $x_h (l + 1 \leq h \leq n)$ with a label:** $y_h = argmax_j Y_{hj}$.

## 3 Experiments and Results

### 3.1 Data

Our proposed graph-based method is evaluated on the ACE corpus [1], which contains 519 files from sources including broadcast, newswire, and newspaper. A break-down of the tagged data by different relation subtypes is given in Table 1.

### 3.2 Features

We extract the following lexical and syntactic features from two entity mentions, and the contexts before, between and after the entity pairs. Especially, we set the mid-context window as everything between the two entities and the pre- and post- context as up to two words before and after the corresponding entity. Most of these features are computed from the parse trees derived from Charniak Parser (Charniak, 1999) and the Chunklink script [2] written by Sabine Buchholz from Tilburg University.

[1] http://www.ldc.upenn.edu/Projects/ACE/

[2] Software available at http://ilk.uvt.nl/∼sabine/chunklink/

Table 2: Performance of Relation Detection: SVM and LP algorithm with different size of labeled data. The LP algorithm is performed with two similarity measures: Cosine similarity and JS divergence.

| | SVM | | | $\text{LP}_{Cosine}$ | | | $\text{LP}_{JS}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Percentage | P | R | F | P | R | F | P | R | F |
| 1% | 35.9 | 32.6 | 34.4 | 58.3 | 56.1 | 57.1 | 58.5 | 58.7 | 58.5 |
| 10% | 51.3 | 41.5 | 45.9 | 64.5 | 57.5 | 60.7 | 64.6 | 62.0 | 63.2 |
| 25% | 67.1 | 52.9 | 59.1 | 68.7 | 59.0 | 63.4 | 68.9 | 63.7 | 66.1 |
| 50% | 74.0 | 57.8 | 64.9 | 69.9 | 61.8 | 65.6 | 70.1 | 64.1 | 66.9 |
| 75% | 77.6 | 59.4 | 67.2 | 71.8 | 63.4 | 67.3 | 72.4 | 64.8 | 68.3 |
| 100% | 79.8 | 62.9 | 70.3 | 73.9 | 66.9 | 70.2 | 74.2 | 68.2 | 71.1 |

Table 3: Performance of Relation Classification on Relation Subtype: SVM and LP algorithm with different size of labeled data. The LP algorithm is performed with two similarity measures: Cosine similarity and JS divergence.

| | SVM | | | $\text{LP}_{Cosine}$ | | | $\text{LP}_{JS}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Percentage | P | R | F | P | R | F | P | R | F |
| 1% | 31.6 | 26.1 | 28.6 | 39.6 | 37.5 | 38.5 | 40.1 | 38.0 | 39.0 |
| 10% | 39.1 | 32.7 | 35.6 | 45.9 | 39.6 | 42.5 | 46.2 | 41.6 | 43.7 |
| 25% | 49.8 | 35.0 | 41.1 | 51.0 | 44.5 | 47.3 | 52.3 | 46.0 | 48.9 |
| 50% | 52.5 | 41.3 | 46.2 | 54.1 | 48.6 | 51.2 | 54.9 | 50.8 | 52.7 |
| 75% | 58.7 | 46.7 | 52.0 | 56.0 | 52.0 | 53.9 | 56.1 | 52.6 | 54.3 |
| 100% | 60.8 | 48.9 | 54.2 | 56.2 | 52.3 | 54.1 | 56.3 | 52.9 | 54.6 |

**Words:** Surface tokens of the two entities and three context windows.

**Entity Type:** the entity type of both entity mentions, which can be PERSON, ORGANIZATION, FACILITY, LOCATION and GPE.

**POS:** Part-Of-Speech tags corresponding to all tokens in the two entities and three context windows.

**Chunking features:** Chunk tag information and Grammatical function of the two entities and three context windows. IOB-chains of the heads of the two entities are also considered. IOB-chain notes the syntactic categories of all the constituents on the path from the root node to this leaf node of tree.

We combine the above features with their position information in the context to form the context vector. Before that, we filter out low frequency features which appeared only once in the entire set.

## 3.3 Experimental Evaluation

### 3.3.1 Relation Detection

We collect all entity mention pairs which co-occur in the same sentence from the training and devtest corpus into two set $C1$ and $C2$ respectively. The set $C1$ includes annotated training data $AC1$ and unrelated data $UC1$. We randomly sample $l$ examples from $AC1$ as **labeled data** and add a "NONE" class into labeled data for the case where the two entity mentions are not related. The data of the "NONE" class is resulted by sampling $l$ examples from $UC1$. Moreover, we combine the rest examples of $C1$ and the whole set $C2$ as **unlabeled data**.

Table 4: Comparison of performance on individual relation type of Zhang (2004)'s method and our method. For Zhang (2004)'s method, feature sampling probability is set to 0.3 and agreement threshold is set to 9 out of 10.

| | Bootstrapping | | | $\text{LP}_{JS}$ | | |
|---|---|---|---|---|---|---|
| Rel-Type | P | R | F | P | R | F |
| ROLE | 78.5 | 69.7 | 73.8 | 81.0 | 74.7 | 77.7 |
| PART | 65.6 | 34.1 | 44.9 | 70.1 | 41.6 | 52.2 |
| AT | 61.0 | 84.8 | 70.9 | 74.2 | 79.1 | 76.6 |
| SOC | 47.0 | 57.4 | 51.7 | 45.0 | 59.1 | 51.0 |
| NEAR | $undef$ | 0 | $undef$ | 13.7 | 12.5 | 13.0 |

Given labeled and unlabeled data,we can perform LP algorithm to detect possible relations, which are those entity pairs that are not classified to the "NONE" class but to the other 24 subtype classes. In addition,we conduct experiments with different sampling set size $l$, including $1\% \times N_{train}$,$10\% \times N_{train}$,$25\% \times N_{train}$,$50\% \times N_{train}$,$75\% \times N_{train}$, $100\% \times N_{train}$ ($N_{train} = |AC1|$). If any major subtype was absent from the sampled labeled set,we redo the sampling. For each size,we perform 20 trials and calculate an average of 20 random trials.

### 3.3.2 SVM vs. LP

Table 2 reports the performance of relation detection by using SVM and LP with different sizes of labled data. For SVM, we use LIBSVM tool with linear kernel function [3]. And the same sampled labeled data used in LP is used to train SVM models. From Table 2, we see that both LP$_{Cosine}$ and LP$_{JS}$ achieve higher *Recall* than SVM. Especially, with small labeled dataset (percentage of labeled data $\leq$ $25\%$), this merit is more distinct. When the percentage of labeled data increases from $50\%$ to $100\%$, LP$_{Cosine}$ is still comparable to SVM in *F-measure* while LP$_{JS}$ achieves better *F-measure* than SVM. On the other hand, LP$_{JS}$ consistently outperforms LP$_{Cosine}$.

Table 3 reports the performance of relation classification, where the performance describes the average values over major relation subtypes. From Table 3, we see that LP$_{Cosine}$ and LP$_{JS}$ outperform SVM by *F-measure* in almost all settings of labeled data, which is due to the increase of *Recall*. With smaller labeled dataset, the gap between LP and SVM is larger. On the other hand, LP$_{JS}$ divergence consistently outperforms LP$_{Cosine}$.

### 3.3.3 LP vs. Bootstrapping

In (Zhang, 2004), they perform relation classification on ACE corpus with bootstrapping on top of SVM. To compare with their proposed Bootstrapped SVM algorithm, we use the same feature stream setting and randomly selected 100 instances from the training data as the size of initial labeled data.

Table 4 lists the performance on individual relation type. We can find that LP algorithm achieves 6.8% performance improvement compared with the (Zhang, 2004)'s bootstrapped SVM algorithm average on all five relation types. Notice that performance reported on relation type "NEAR" is low, because it occurs rarely in both training and test data.

## 4 Conclusion and Future work

This paper approaches the task of semi-supervised relation extraction on Label Propagation algorithm. Our results demonstrate that, when only very few labeled examples are available, this manifold learning based algorithm can achieve better performance than supervised learning method (SVM) and bootstrapping based method, which can contribute to minimize corpus annotation requirement. In the future we would like to investigate how to select more useful feature stream and whether feature selection method can improve the performance of our graph-based semi-supervised relation extraction.

[3] $LIBSVM$: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

## References

Agichtein E. and Gravano L. 2000. *Snowball: Extracting Relations from large Plain-Text Collections, In Proceeding of the* $5^{th}$ *ACM International Conference on Digital Libraries*.

Brin Sergey. 1998. *Extracting patterns and relations from world wide web. In Proceeding of WebDB Workshop at 6th International Conference on Extending Database Technology*. pages 172-183.

Charniak E. 1999. *A Maximum-entropy-inspired parser. Technical Report CS-99-12*. Computer Science Department, Brown University.

Culotta A. and Soresen J. 2004. *Dependency tree kernels for relation extraction, In Proceedings of 42th ACL conference*.

Hasegawa T., Sekine S. and Grishman R. 2004. *Discovering Relations among Named Entities from Large Corpora, In Proceeding of Conference ACL2004*. Barcelona, Spain.

Kambhatla N. 2004. *Combining lexical, syntactic and semantic features with Maximum Entropy Models for extracting relations, In Proceedings of 42th ACL conference*. Spain.

Lin,J. 1991. *Divergence Measures Based on the Shannon Entropy. IEEE Transactions on Information Theory*. 37:1,145-150.

Miller S.,Fox H.,Ramshaw L. and Weischedel R. 2000. *A novel use of statistical parsing to extract information from text. In Proceedings of 6th Applied Natural Language Processing Conference* 29 April-4 may 2000, Seattle USA.

Yarowsky D. 1995. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. pp.189-196.

Zelenko D., Aone C. and Richardella A. 2002. *Kernel Methods for Relation Extraction, In Proceedings of the EMNLP Conference*. Philadelphia.

Zhang Zhu. 2004. *Weakly-supervised relation classification for Information Extraction, In proceedings of ACM 13th conference on Information and Knowledge Management*. 8-13 Nov 2004. Washington D.C.,USA.

Zhou GuoDong, Su Jian, Zhang Jie and Zhang min. 2005. *Combining lexical, syntactic and semantic features with Maximum Entropy Models for extracting relations, In proceedings of 43th ACL conference*. USA.

Zhu Xiaojin and Ghahramani Zoubin. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation. CMU CALD tech report CMU-CALD-02-107*.