Reducing Wrong Labels in Distant Supervision for Relation Extraction

Shingo Takamatsu
System Technologies Laboratories
Sony Corporation
5-1-12 Kitashinagawa, Shinagawa-ku, Tokyo
Shingo.Takamatsu@jp.sony.com

Issei Sato and **Hiroshi Nakagawa**
Information Technology Center
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo
{sato@r., n3@}dl.itc.u-tokyo.ac.jp

Abstract

In relation extraction, distant supervision seeks to extract relations between entities from text by using a knowledge base, such as Freebase, as a source of supervision. When a sentence and a knowledge base refer to the same entity pair, this approach heuristically labels the sentence with the corresponding relation in the knowledge base. However, this heuristic can fail with the result that some sentences are labeled wrongly. This noisy labeled data causes poor extraction performance. In this paper, we propose a method to reduce the number of wrong labels. We present a novel generative model that directly models the heuristic labeling process of distant supervision. The model predicts whether assigned labels are correct or wrong via its hidden variables. Our experimental results show that this model detected wrong labels with higher performance than baseline methods. In the experiment, we also found that our wrong label reduction boosted the performance of relation extraction.

1 Introduction

Machine learning approaches have been developed to address relation extraction, which is the task of extracting semantic relations between entities expressed in text. Supervised approaches are limited in scalability because labeled data is expensive to produce. A particularly attractive approach, called distant supervision (DS), creates labeled data by heuristically aligning entities in text with those in a knowledge base, such as Freebase (Mintz et al., 2009).

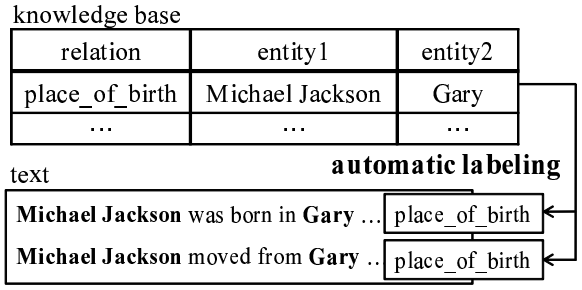

Figure 1: Automatic labeling by distant supervision. Upper sentence: correct labeling; lower sentence: incorrect labeling.

With DS it is assumed that if a sentence contains an entity pair in a knowledge base, such a sentence actually expresses the corresponding relation in the knowledge base.

However, the DS assumption can fail, which results in noisy labeled data and this causes poor extraction performance. An entity pair in a target text generally expresses more than one relation while a knowledge base stores a subset of the relations. The assumption ignores this possibility. For instance, consider the *place_of_birth* relation between *Michael Jackson* and *Gary* in Figure 1. The upper sentence indeed expresses the *place_of_birth* relation between the two entities. In DS *place_of_birth* is assigned to the sentence, and it becomes a useful training example. On the other hand, the lower sentence does not express this relation between the two entities, but the DS heuristic wrongly labels the sentence as expressing it.

Riedel et al. (2010) relax the DS assumption as at least one sentence containing an entity pair ex-

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 721–729,
Jeju, Republic of Korea, 8-14 July 2012.

pressing the corresponding relation in the knowledge base. They cast the relaxed assumption as multi-instance learning. However, even the relaxed assumption can fail. The relaxation is equivalent to the DS assumption when a labeled pair of entities is mentioned once in a target corpus (Riedel et al., 2010). In fact, 91.7% of entity pairs appear only once in Wikipedia articles (see Section 7).

In this paper, we propose a method to reduce the number of wrong labels generated by DS without using either of these assumptions. Given the labeled corpus created with the DS assumption, we first predict whether each *pattern*, which frequently appears in text to express a relation (see Section 4), expresses a target relation. Patterns that are predicted not to express the relation are used to form a negative pattern list for removing wrong labels of the relation.

The main contributions of this paper are as follows:

- To make the pattern prediction, we propose a generative model that directly models the process of automatic labeling in DS. Without any strong assumptions like Riedel et al. (2010)'s, the model predicts whether each pattern expresses each relation via hidden variables (see Section 5).
- Our variational inference for our generative model lets us automatically calibrate parameters for each relation, which are sensitive to the performance (see Section 6).
- We applied our method to Wikipedia articles using Freebase as a knowledge base and found that (i) our model identified patterns expressing a given relation more accurately than baseline methods and (ii) our method led to better extraction performance than the original DS (Mintz et al., 2009) and MultiR (Hoffmann et al., 2011), which is a state-of-the-art multi-instance learning system for relation extraction (see Section 7).

2 Related Work

The increasingly popular approach, called distant supervision (DS), or weak supervision, utilizes a knowledge base to heuristically label a corpus (Wu and Weld, 2007; Bellare and McCallum, 2007; Pal et al., 2007). Our work was inspired by Mintz et al. (2009) who used Freebase as a knowledge base by making the DS assumption and trained relation extractors on Wikipedia. Previous works (Hoffmann et al., 2010; Yao et al., 2010) have pointed out that the DS assumption generates noisy labeled data, but did not directly address the problem. Wang et al. (2011) applied a rule-based method to the problem by using popular entity types and keywords for each relation. In (Bellare and McCallum, 2007; Riedel et al., 2010; Hoffmann et al., 2011), they used multi-instance learning, which deals with uncertainty of labels, to relax the DS assumption. However, the relaxed assumption can fail when a labeled entity pair is mentioned only once in a corpus (Riedel et al., 2010). Our approach relies on neither of these assumptions.

Bootstrapping for relation extraction (Riloff and Jones, 1999; Pantel and Pennacchiotti, 2006; Carlson et al., 2010) is related to our method. In bootstrapping, seed entity pairs of the target relation are given in order to select reliable patterns, which are used to extract new entity pairs. To avoid the selection of unreliable patterns, bootstrapping introduces scoring functions for each pattern candidate. This can be applied to our approach, which seeks to reduce the number of unreliable patterns by using a set of given entity pairs. However, the bootstrapping-like approach suffers from sensitive parameters that are critical to its performance. Ideally, the parameters such as a threshold for scoring function should be determined for each relation, but there are no principled methods (Komachi et al., 2008). In our approach, parameters are calibrated for each relation by maximizing the likelihood of our generative model.

3 Knowledge-based Distant Supervision

In this section, we describe DS for relation extraction. We use the term *relation* as the relation between two entities. A *relation instance* is a tuple consisting of two entities and relation r. For example, *place_of_birth*(*Michael Jackson*, *Gary*) in Figure 1 is a relation instance.

Relation extraction seeks to extract relation instances from text. An entity is mentioned as a named entity in text. We extract a relation instance from a

single sentence. For example, from the upper sentence in Figure 1 we extract *place_of_birth*(*Michael Jackson*, *Gary*). Since two entities mentioned in a sentence do not always have a relation, we select entity pairs from a corpus when: (i) the path of the dependency parse tree between the corresponding two named entities in the sentence is no longer than 4 and (ii) the path does not contain a sentence-like boundary, such as a relative clause[1] (Banko et al., 2007; Banko and Etzioni, 2008). Banko and Etzioni (2008) found that a set of eight lexico-syntactic forms covers nearly 95% of relation phrases in their corpus. (Fader et al. (2011) found that this set covers 69% of their corpus). Our rule is designed to cover at least the eight lexico-syntactic forms. We use the entity pairs extracted by this rule.

DS uses a knowledge base to create labeled data for relation extraction by heuristically matching entity pairs. A *knowledge base* is a set of relation instances about predefined relations. For each sentence in the corpus, we extract all of its entity pairs. Then, for each entity pair, we try to retrieve the relation instances about the entity pair from the knowledge base. If we found such a relation instance, then the set of its relation, the entity pair, and the sentence is stored as a positive example. If not, then the set of the entity pair and the sentence is stored as a negative example. Features of an entity pair are extracted from the sentence containing the entity pair.

As mentioned in Section 1, the assumption of DS can fail, resulting in wrong assignments of a relation to sentences that do not express the relation. We call such assignments *wrong labels*. An example of a wrong label is *place_of_birth* assigned to the lower sentence in Figure 1.

4 Wrong Label Reduction

We define a *pattern* as the entity types of an entity pair[2] as well as the sequence of words on the path of the dependency parse tree from the first entity to the second one. For example, from "Michael Jackson was born in Gary" in Figure 1, the pattern "[Person] born in [Location]" is extracted. We use entity types to distinguish the sentences that express different relations with the same dependency path, such as "ABBA was formed in Stockholm." and "ABBA was formed in 1970."

Algorithm 1 Wrong Label Reduction

labeled data generated by DS: LD
negative patterns for relation r: $NegPat(r)$
for each entry (r, $Pair$, $Sentence$) in LD **do**
  pattern $Pat \leftarrow$ the pattern from ($Pair$, $Sentence$)
  if $Pat \in NegPat(r)$ **then**
    remove (r, $Pair$, $Sentence$) from LD
  end if
end for
return LD

Our aim is to remove wrong labels assigned to frequent patterns, which cause poor precision. Indeed, in our Wikipedia corpus, more than 6% of the sentences containing the pattern "[Person] moved to [Location]", which does not express *place_of_death*, are labeled as *place_of_death*, and the labels assigned to these sentences hurt extraction performance (see Section 7.3.3). We would like to remove *place_of_death* from the sentences that contain this pattern.

In our method, we reduce the number of wrong labels as follows: (i) given a labeled corpus with the DS assumption, we first predict whether a pattern expresses a relation and then (ii) remove wrong labels using the negative pattern list, which is defined as patterns that are predicted not to express the relation. In the first step, we introduce the novel generative model that directly models DS's labeling process and make the prediction (see Section 5). The second step is formally described in Algorithm 1. For relation extraction, we train a classifier for entity pairs using the resultant labeled data.

5 Generative Model

We now describe our generative model, which predicts whether a pattern expresses relation r or not via hidden variables. In this section, we consider relation r since parameters are conditionally independent if relation r and the hyperparameter are given.

An observation of our model is whether entity pair i appearing with pattern s in the corpus is labeled with relation r or not. Our binary observations are written as $\mathbf{X}_r = \{(x_{rsi}) | s = 1, \ldots, S, i =$

[1] We reject sentence-like dependencies such as *ccomp*, *complm* and *mark*

[2] If we use a standard named entity tagger, the entity types are Person, Location, and Organization.

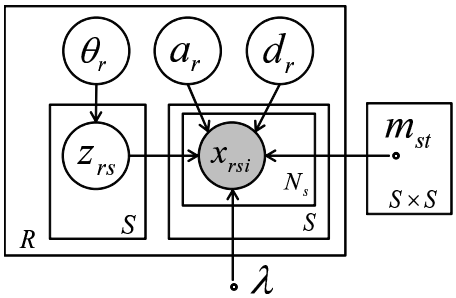

Figure 2: Graphical model representation of our model. R indicates the number of relations. S is the number of patterns. N_s is the number of entity pairs that appear with pattern s in the corpus. x_{rsi} is the observed variables. The circled variables except x_{rsi} are parameters or hidden variables. λ is the hyperparameter and m_{st} is constant. The boxes are "plates" representing replicates.

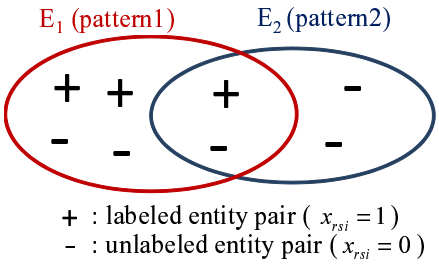

Figure 3: Venn diagram-like description. E_1 and E_2 are sets of entity pairs. E_1/E_2 has 6/4 entity pairs because the 6/4 entity pairs appear with pattern 1/2 in the target corpus. Pattern 1 expresses relation r and pattern 2 does not. Elements in E_1 are labeled with probability $a_r = 3/6 = 0.5$. Those in E_2 are labeled with probability $b_{r2} = a_r(|E_1 \cap E_2|/|E_2|) = 0.5(2/4) = 0.25$.

$1, \ldots, N_s\}$,[3] where we define S to be the number of patterns and N_s to be the number of entity pairs appearing with pattern s. Note that we count an entity pair for given pattern s once even if the entity pair is mentioned with pattern s more than once in the corpus, because DS assigns the same relation to all mentions of the entity pair.

Given relation r, our model assumes the following generative process:

1. For each pattern s
   Choose whether s expresses relation r or not
   $$z_{rs} \sim Be(\theta_r)$$
2. For each entity pair i appearing with pattern s
   Choose whether i is labeled or not
   $$x_{rsi} \sim P(x_{rsi}|\mathbf{Z}_r, a_r, d_r, \lambda, \mathbf{M}),$$

where $Be(\theta_r)$ is a Bernoulli distribution with parameter θ_r, z_{rs} is a binary hidden variable that is 1 if pattern s expresses relation r and 0 otherwise, and $\mathbf{Z}_r = \{(z_{rs})|s = 1, \ldots, S\}$. Given a value of z_{rs}, we model two kinds of probabilities: one for patterns that actually express relation r, i.e., $P(x_{rsi} = 1|z_{rs} = 1)$, and one for patterns that do not express r, i.e., $P(x_{rsi} = 1|z_{rs} = 0)$. The former is simply parameterized as $0 \leq a_r \leq 1$. We express the latter as $b_{rs} = P(x_{rsi} = 1|\mathbf{Z}_r, a_r, d_r, \lambda, \mathbf{M})$, which is a function of $\mathbf{Z}_r$, a_r, d_r, λ and $\mathbf{M}$; we explain its modeling in the following two subsections.

[3]Since a set of entity pairs appearing with pattern s is different, i should be written as i_s. For simplicity, however, we use i for each pattern.

The graphical model of our model is shown in Figure 2.

5.1 Example of Wrong Labeling

Using a simple example, we describe how we model b_{rs}, the probability with which DS assigns relation r to pattern s via entity pairs when pattern s does not express relation r.

Consider two patterns: pattern 1 that expresses relation r and pattern 2 that does not (i.e., $z_{r1} = 1$ and $z_{r2} = 0$). We also assume that there are entity pairs that appear with pattern 1 as well as with pattern 2 in different places in the corpus (for example, *Michael Jackson* and *Gary* in Figure 1). When such entity pairs are labeled, relation r is assigned to pattern 1 and at the same time to wrong pattern 2. Such entity pairs are observed as elements in the intersection of the two sets of entity pairs, E_1 and E_2. Here, E_s is the set of entity pairs that appear with pattern s in the corpus. This situation is described in Figure 3.

We model probability b_{r2} as follows. In E_1, an entity pair is labeled with probability a_r. We assume that entity pairs in the intersection, $E_1 \cap E_2$, are also labeled with a_r. From the viewpoint of E_2, entity pairs in its subset, $E_1 \cap E_2$, are labeled with a_r. Therefore, b_{r2} is modeled as

$$b_{r2} = a_r \frac{|E_1 \cap E_2|}{|E_2|},$$

where $|E|$ denotes the number of elements in set E. An example of this calculation is shown in Figure 3.

We generalize the example in the next subsection.

5.2 Modeling of Probability b_{rs}

We model b_{rs} so that it is proportional to the number of entity pairs that are shared with correct patterns whose $z_{rs} = 1$, i.e.,

$$b_{rs} = a_r \frac{\left| \left(\bigcap_{\{t|z_{rt}=1, t\neq s\}} E_t \right) \cap E_s \right|}{|E_s|}, \quad (1)$$

where $\bigcap$ indicates set intersections. However, the enumeration in Eq.1 requires $O(SN_s^2)$ computational cost and a huge amount of memory to store all of the entity pairs. We approximate the right-hand side of Eq.1 as

$$b_{rs} \approx a_r \left(1 - \prod_{t=1, t\neq s}^{S} \left(1 - \frac{|E_t \cap E_s|}{|E_s|} \right)^{z_{rt}} \right).$$

This approximation is made, given the sizes of all E_ss and those of all intersections of two E_ss. This has a lower computational cost of $O(S)$ and let us use less memory. We define $S \times S$ matrix $\mathbf{M}$ whose elements are $m_{st} = |E_t \cap E_s|/|E_s|$.

In reality, factors other than the process described in the previous subsection can cause wrong labeling (for example, errors in the knowledge base). We introduce a parameter $0 \leq d_r \leq 1$ that covers such factors. Finally, we define b_{rs} as

$$b_{rs} \equiv a_r \left(\lambda \left(1 - \prod_{t=1, t\neq s}^{S} (1 - m_{st})^{z_{rt}} \right) + (1 - \lambda)\, d_r \right), \quad (2)$$

where $0 \leq \lambda \leq 1$ is the hyperparameter that controls how strongly b_{rs} is affected by the main labeling process explained in the previous subsection.

5.3 Likelihood

Given observation $\mathbf{X}_r$, the likelihood of our model is

$$P(\mathbf{X}_r|\theta_r, a_r, d_r, \lambda, \mathbf{M}) = \sum_{\mathbf{Z}_r} P(\mathbf{Z}_r|\theta_r)\, P(\mathbf{X}_r|\mathbf{Z}_r, a_r, d_r, \lambda, \mathbf{M}),$$

where

$$P(\mathbf{Z}_r|\theta_r) = \prod_{s=1}^{S} \theta_r^{z_{rs}} (1 - \theta_r)^{1 - z_{rs}}.$$

For each pattern s, we define n_{rs} as the number of entity pairs to which relation r is assigned (i.e., $n_{rs} = \sum_i x_{rsi}$).

$$p(\mathbf{X}_r|\mathbf{Z}_r, a_r, d_r, \lambda, \mathbf{M}) = \prod_{s=1}^{S} \left\{ a_r^{n_{rs}} (1 - a_r)^{N_s - n_{rs}} \right\}^{z_{rs}} \left\{ b_{rs}^{n_{rs}} (1 - b_{rs})^{N_s - n_{rs}} \right\}^{1 - z_{rs}}, \quad (3)$$

where b_{rs} is in Eq.2.

6 Learning

We learn parameters a_r, θ_r, and d_r and infer hidden variables $\mathbf{Z}_r$ by maximizing the log likelihood given $\mathbf{X}_r$. Estimated $\mathbf{Z}_r$ is used to predict which patterns express relation r.

To infer z_{rs}, we would like to calculate the posterior probability of z_{rs}. However, this calculation is intractable because each z_{rs} depends on the others, $\{(z_{rt})|t \neq s\}$, as shown in Eqs.2 and 3. This prevents us from using the EM algorithm. Instead, we apply variational approximation to the posterior distribution by using the following trial distribution:

$$Q(\mathbf{Z}_r|\Phi_r) = \prod_{s=1}^{S} \phi_{rs}^{z_{rs}} (1 - \phi_{rs})^{1 - z_{rs}},$$

where $0 \leq \phi_{rs} \leq 1$ is a parameter for the trial distribution.

The following function F_r is a lower bound of the log likelihood, and maximizing this function with respect to Φ_r is equivalent to minimizing the KL divergence between the trial distribution and the posterior distribution of $\mathbf{Z}_r$.

$$F_r = E_Q[\log P(\mathbf{Z}_r, \mathbf{X}_r|\theta_r, a_r, d_r, \lambda, \mathbf{M})] - E_Q[\log Q(\mathbf{Z}_r|\Phi_r)]. \quad (4)$$

$E_Q[\bullet]$ represents the expectation over trial distribution Q. We maximize function F_r with respect to the parameters instead of the log likelihood.

However, we need further approximation for two terms on expanding Eq.4. Both of the terms are expressed as $E_Q[\log(f(\mathbf{Z}_r))]$, where $f(\mathbf{Z}_r)$ is a function of $\mathbf{Z}_r$. We apply the following approximation (Asuncion et al., 2009).

$$E_Q[\log(f(\mathbf{Z}_r))] \approx \log(E_Q[f(\mathbf{Z}_r)]).$$

This is based on the Taylor series of log at $E_Q[f(\mathbf{Z}_r)]$. In our problem, since the second derivative is sufficiently small, we use the zeroth-order approximation.[4]

Our learning algorithm is derived by calculating the stationary condition of the resultant evaluation function with respect to each parameter. We have the exact solution for θ_r. For each ϕ_{rs} and d_r, we derive a fixed point iteration. We update a_r by using the steepest ascent. We update each parameter in turn while keeping the other parameters fixed. Parameter updating proceeds until a termination condition is met.

After learning, we have ϕ_{rs} for each pair of relation r and pattern s. The greater the value of ϕ_{rs} is, the more likely it is that pattern s expresses relation r. We set a threshold and determine $z_{rs} = 0$ when ϕ_{rs} is less than the threshold.

7 Experiments

We performed two sets of experiments.

Experiment 1 aimed to evaluate the performance of our generative model itself, which predicts whether a pattern expresses a relation, given a labeled corpus created with the DS assumption.

Experiment 2 aimed to evaluate how much our wrong label reduction in Section 4 improved the performance of relation extraction. In our method, we trained a classifier with a labeled corpus cleaned by Algorithm 1 using the negative pattern list predicted by the generative model.

7.1 Dataset

Following Mintz et al. (2009), we carried out our experiments using Wikipedia as the target corpus and Freebase (September, 2009, (Google, 2009)) as the knowledge base. We used more than 1,300,000 Wikipedia articles in the wex dump data (September, 2009, (Metaweb Technologies, 2009)). The properties of our data are shown in Table 1.

In Wikipedia articles, named entities were identified by anchor text linking to another article and starting with a capital letter (Yan et al., 2009). We applied Open NLP POS tagger[5] and MaltParser (Nivre et al., 2007) to sentences containing more than one named entity. We then extracted sentences containing related entity pairs with the method explained in Section 3. To match entity pairs, we used ID mapping between the dump data and Freebase. We used the most frequent 24 relations.

Table 1: Properties of Wikipedia dataset

documents	1,303,000
entity pairs	2,017,000
(matched to Freebase)	129,000
(with entity types)	913,000
frequent patterns	3,084
relations	24

7.2 Experiment 1: Pattern Prediction

We compared our model with baseline methods in terms of ability to predict patterns that express a given relation.

The input of this task was $\mathbf{X}_r$s, which expresses whether or not each entity pair appearing with each pattern is labeled with relation r, as explained in Section 5. In Experiment 1, since we needed entity types for patterns, we restricted ourselves to entities matched with Freebase, which also provides entity types for entities. We used patterns that appear more than 20 times in the corpus.

7.2.1 Evaluation

We split the data into training data and test data. The training data was $\mathbf{X}_r$s for 12 relations and the test data was that for the remaining 12 relations. The training data was used to calibrate parameters (see the following subsection for details). The test data was used for evaluation. We randomly split the data five times and took the average of the following evaluation values.

We evaluated the performance by precision, recall, and F value. They were calculated using gold standard data, which was constructed by hand. We manually selected patterns that actually express a target relation as positive patterns for the relation. [6] We averaged the evaluation values in terms of macro average over relations before averaging over the data splits.

[4] The first-order information becomes zero in this case.

[5] http://opennlp.sourceforge.net/

[6] Patterns that ambiguously express the relation, for instance "[Person] in [Location]" for *place_of_birth*, were not selected as positive patterns.

Table 2: Averages of precision, recall, and F value in Experiment 1. The averages of threshold of RS(rank) and RS(value) were 6.2 $\pm$ 3.2 and 0.10 $\pm$ 0.06, respectively. The averages of hyperparameters of PROP were 0.84 $\pm$ 0.05 for λ and 0.85 $\pm$ 0.10 for the threshold.

	Precision	Recall	F value
Baseline	0.339	1.000	0.458
RS(rank)	0.749	0.549	0.467
RS(value)	0.601	0.647	0.545
PROP	0.782	0.688	0.667

Table 3: Example of estimated ϕ_{rs} for $r = place\_of\_birth$. Entity types are omitted in patterns. n_{rs}/N_s is the ratio of the number of labeled entity pairs to the number of entity pairs appearing with pattern s.

pattern s	n_{rs}/N_s	ϕ_{rs}	expresses r?
born in	0.512	0.999	true
actor from	0.480	0.999	true
elected Mayor of	0.384	0.855	false
family moved from	0.344	0.055	false
native of	0.327	0.999	true
grew in	0.162	0.000	false

7.2.2 Methods

We compared the following methods:

Baseline: This method assigns relation r to a pattern when the pattern is mentioned with at least one entity pair corresponding to relation r in Freebase. This method is based on the DS assumption.

Ratio-based Selection (RS): Given relation r and pattern s, this method calculates n_{rs}/N_s, which is the ratio of the number of labeled entity pairs appearing with pattern s to the number of entity pairs including unlabeled ones. RS then selects the top n patterns (RS(rank)). We also tested a version using a real-valued threshold (RS(value)). In training, we selected the threshold that maximized the F value. Some bootstrapping approaches (Carlson et al., 2010) use a rank-based threshold like RS(rank).

Proposed Model (PROP): Using the training data, we determined the two hyperparameters, λ and the threshold to round ϕ_{rs} to 1 or 0, so that they maximized the F value. When ϕ_{rs} is greater than the threshold, we select pattern s as one expressing relation r.

7.2.3 Result and Discussion

The results of Experiment 1 are shown in Table 2. Our model achieved the best precision, recall, and F value. RS(value) had the second best F value, but it completely removed more than one infrequent relation on average in test sets. This is problematic for real situations. RS(rank) achieved the second highest precision. However, its recall, which is also important in our task, was the lowest and its F value was almost the same as naive Baseline.

The thresholds of RS, which directly affect their performance, should be calibrated for each relation, but it is hard to do this in advance. On the other hand, our model learns parameters such as a_r for each relation and thus the hyperparameter of our model does not directly affect its performance. This results in a high prediction performance.

Examples of estimated ϕ_{rs}, the probability with which pattern s expresses relation r, are shown in Table 3. The pattern, "[Person] family moved from [Location]", which does not express *place_of_birth*, had low ϕ_{rs} in spite of having higher n_{rs}/N_s than the valid pattern "[Person] native of [Location]". The former pattern had higher b_{rs}, the probability with which relation r is wrongly assigned to pattern s via entity pairs, because there were more entity pairs that appeared not only with this pattern but also with patterns that was predicted to express *place_of_birth*.

7.3 Experiment 2: Relation Extraction

We investigated the performance of relation extraction using our wrong label reduction, which uses the results of the pattern prediction.

Following Mintz et al. (2009), we performed an automatic held-out evaluation and a manual evaluation. In both cases, we used 400,000 articles for testing and the remaining 903,000 for training.

7.3.1 Configuration of Classifiers

Following Mintz et al. (2009), we used a multiclass logistic classifier optimized using L-BFGS with Gaussian regularization to classify entity pairs to the predefined 24 relations and NONE. In order to train the NONE class, we randomly picked 100,000 examples that did not match to Freebase as pairs. (Several entities in the examples matched and had entity types of Freebase.) In this experiment, we

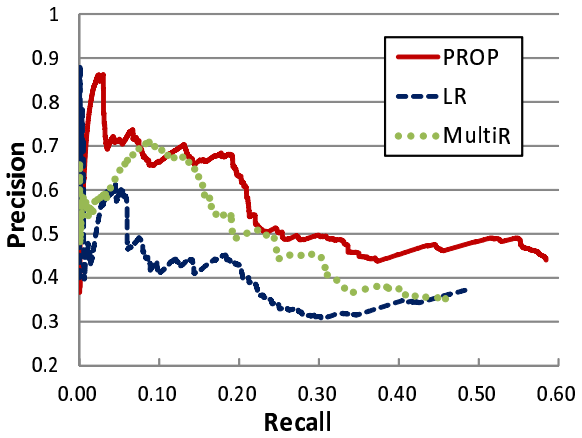

Figure 4: Precision-recall curves in held-out evaluation. Precision is reported at recall levels from 5 to 50,000.

used not only entity pairs matched to Freebase but also ones not matched to Freebase (i.e., entity pairs that do not have entity types). We used syntactic features (i.e., features obtained from the dependency parse tree of a sentence) and lexical features, and entity types, which essentially correspond to the ones developed by Mintz et al. (2009).

We compared the following methods: logistic regression with the labeled data cleaned by the proposed method (PROP), logistic regression with the standard DS labeled data (LR), and MultiR proposed in (Hoffmann et al., 2011) as a state-of-the-art multi-instance learning system.[7] For logistic regression, when more than one relation is assigned to a sentence, we simply copied the feature vector and created a training example for each relation. In PROP, we used training articles for pattern prediction.[8]

7.3.2 Held-out Evaluation

In the held-out evaluation, relation instances discovered from testing articles were automatically compared with those in Freebase. This let us calculate the precision of each method for the best n relation instances. The precisions are underestimated because this evaluation suffers from false negatives due to the incompleteness of Freebase. We changed n from 5 to 50,000 and measured precision and recall. Precision-recall curves for the held-out data are shown in Figure 4.

PROP achieved comparable or higher precision at most recall levels compared with LR and MultiR. Its performance at $n = 50{,}000$ is much higher than that of the others. While our generative model does not use unlabeled examples as negative ones in detecting wrong labels, classifier-based approaches including MultiR do, suffering from false negatives.

Table 4: Averages of precisions at 50 for the most frequent 15 relations as well as example relations.

	PROP	MultiR	LR
place_of_birth	1.0	1.0	0.56
place_of_death	1.0	0.7	0.84
average	0.89 ± 0.14	0.83 ± 0.21	0.82 ± 0.23

7.3.3 Manual Evaluation

For manual evaluation, we picked the top ranked 50 relation instances for the most frequent 15 relations. The manually evaluated precisions averaged over the 15 relations are shown in table 4.

PROP achieved the best average precision. For *place_of_birth*, LR wrongly extracted entity pairs with "[Person] played with club [Location]", which does not express the relation. PROP and MultiR avoided this mistake. For *place_of_death*, LR and MultiR wrongly extracted entity pairs with "[Person] moved to [Location]". Multi-instance learning does not work for wrong labels assigned to entity pairs that appear only once in a corpus. In fact, 72% of entity pairs that appeared with this pattern and were wrongly labeled as *place_of_death* appeared only once in the corpus. Only PROP avoided mistakes of this kind because our method works in such situations.

8 Conclusion

We proposed a method that reduces the number of wrong labels created with the DS assumption, which is widely applied. Our generative model directly models the labeling process of DS and predicts patterns that are wrongly labeled with a relation. The predicted patterns are used for wrong label reduction. The experimental results show that this method successfully reduced the number of wrong labels and boosted the performance of relation extraction.

[7] For MultiR, we used the authors' implementation from http://www.cs.washington.edu/homes/raphaelh/mr/

[8] In Experiment 2 we set $\lambda = 0.85$ and the threshold at 0.95.

References

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee W. Teh. 2009. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*, pages 27–34.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '08)*, pages 28–36.

Michele Banko, Michael J Cafarella, Stephen Soderl, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI '07)*, pages 2670–2676.

Kedar Bellare and Andrew McCallum. 2007. Learning Extractors from Unlabeled Text using Relevant Databases. In *Sixth International Workshop on Information Integration on the Web (IIWeb '07)*.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*, pages 101–110.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1535–1545.

Google. 2009. Freebase data dumps. `http://download.freebase.com/datadumps/`.

Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 286–295.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '11)*, pages 541–550.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 1011–1020.

Metaweb Technologies. 2009. Freebase wikipedia extraction (wex). `http://download.freebase.com/wex/`.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP '09)*, pages 1003–1011.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 37:95–135.

Chris Pal, Gideon Mann, and Richard Minerich. 2007. Putting semantic information extraction on the map: Noisy label models for fact extraction. In *Sixth International Workshop on Information Integration on the Web (IIWeb '07)*.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '06)*, pages 113–120.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD '10)*, pages 148–163.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. 2011. Relation extraction with relation topics. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1426–1436.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management (CIKM '07)*, pages 41–50.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP '09)*, pages 1021–1029.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*, pages 1013–1023.