Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Relation Adaptation: Learning to Extract Novel Relations with Minimum Supervision

Danushka Bollegala Yutaka Matsuo Mitsuru Ishizuka
The University of Tokyo
Hongo, 7-3-1, Tokyo,
113-8656, Japan.

Abstract

Extracting the relations that exist between two entities is an important step in numerous Web-related tasks such as information extraction. A supervised relation extraction system that is trained to extract a particular relation type might not accurately extract a new type of a relation for which it has not been trained. However, it is costly to create training data manually for every new relation type that one might want to extract. We propose a method to *adapt* an existing relation extraction system to extract new relation types with minimum supervision. Our proposed method comprises two stages: *learning a lower-dimensional projection* between different relations, and *learning a relational classifier* for the target relation type with instance sampling. We evaluate the proposed method using a dataset that contains 2000 instances for 20 different relation types. Our experimental results show that the proposed method achieves a statistically significant macro-average F-score of 62.77. Moreover, the proposed method outperforms numerous baselines and a previously proposed weakly-supervised relation extraction method.

1 Introduction

The World Wide Web contains information related to numerous real-world entities (e.g. persons, locations, organizations, etc.) interconnected by various semantic relations. Accurately detecting the semantic relations that exist between two entities is of paramount importance for numerous tasks on the Web such as information extraction (IE) [Banko *et al.*, 2007]. Recent work on relation extraction has demonstrated that supervised machine learning algorithms coupled with intelligent feature engineering provide state-of-the-art solutions to this problem [GuoDong *et al.*, 2005]. However, supervised learning algorithms depend heavily on the availability of adequate labeled data for the target relation types that must be extracted. Considering the potentially numerous semantic relations that exist among entities on the Web, it is costly to create labeled data manually for each new relation type that we want to extract. Instead of annotating a large set of training data manually for each new relation type, it would be cost effective if we could somehow *adapt* an existing relation extraction system to those new relation types using a small set of training instances. As described in this paper, we examine *relation adaptation* – how to adapt an existing relation extraction system that is trained to extract some specific relation types, to extract new relation types in a weakly-supervised setting. We designate the existing relation types on which a relation extraction system has been trained as *source* relations, whereas the novel relation type to which we must adapt is called the *target* relation.

We must overcome three fundamental challenges when adapting a relation extraction system to new relation types. First, a semantic relation that exists between two entities can be expressed using more than one lexical or syntactic pattern. For example, the acquiredBy relation that exist between two companies **X** and **Y** can be expressed using lexical patterns such as **X** *acquires* **Y**, **X** *buys* **Y**, and **X** *purchases* **Y**. To classify a relation accurately, we must recognize the different ways in which it can be expressed on the Web. Second, the types of relations are strongly dependent on the application domain. For example, in the *financial* domain, we might be interested in extracting relations such as acquiredBy (between two companies) and ceoOf (between a company and a person), whereas, in the *movie* domain we might be interested in extracting relations such as actedIn (between an actor and a movie) and directed (between a director and a movie). Therefore, a classifier trained on the financial domain might not be applied directly to classify relations in the movie domain because the two domains have different sets of relations. Third, the labeled instances for the target relation are markedly fewer than those for the source relations. It is challenging to learn a classifier for the target relation type using such an unbalanced dataset.

We propose a two-stage approach to adapt an existing relation extraction system to new relation types. First, to represent a semantic relation $\mathcal{R}$ that exists between two entities A and B, we extract lexical and syntactic patterns from contexts in which those two entities co-occur. Our method is inspired by the observation that different semantic relations share some lexical and syntactic patterns. For example, the lexical pattern **X** *directs* **Y** holds between a company **Y** and its CEO **X**, as well as a movie **Y** and its director **X**. We designate patterns that appear in different relation types as *relation-independent* patterns, whereas patterns that appear only in a particular relation type are called *relation-specific* patterns. If we can find the correspondence between relation-specific and relation independent patterns, then we can use that to *transfer* the knowledge learnt for a particular relation to recognize a different relation. To identify relation-specific and relation-independent patterns, we propose the use of the entropy of a pattern over the distribution of entity pairs. We then create a bipartite graph between relation-specific and relation-independent patterns and perform spectral clustering on this graph to compute a lower-dimensional mapping between relation-specific and relation-independent patterns. This mapping is used to *project* feature vectors to train a relational classifier.

In the second stage, we train a classifier for the target relation type using training instances for both source and target relation types. A fundamental problem in training a relational classifier for a target relation type for which only a few labeled instances are available is that, because of the numerous source relation instances, the finally trained classifier becomes biased towards the source relation types. To solve this problem, we propose a method that first samples a subset of source relation instances. Then we use that subset to train a classifier for the target relation type. This method reduces the imbalance between source and target relation datasets, thereby improving the classification accuracy for the target relation type.

2 Relation Adaptation

2.1 Problem Definition

Given two entities A and B, we define relation extraction as the task of selecting the relation $\mathcal{R}$, that exists between A and B, from a given set of relation types. Moreover, entity pair (A, B) is regarded as an *instance* of the relation $\mathcal{R}$. According to our definition, relation extraction can be modeled as a multi-class classification problem.

Definition: We define **Relation Adaptation** as the problem of learning a classifier for a target relation type $\mathcal{T}$, for which we have a few entity pairs as training instances, given numerous entity pairs for some N source relation types, $\mathcal{S}_1, \ldots, \mathcal{S}_N$. We use the notation $\Omega = \{\mathcal{S}_1, \ldots, \mathcal{S}_N, \mathcal{T}\}$ to denote the set of all relations. A particular relation type from this set is denoted by $\mathcal{R}$ (i.e $\mathcal{R} \in \Omega$). An entity pair that consists of two entities A and B is denoted as (A, B). Moreover, we use the notation $(A, B) \in \mathcal{R}$ to indicate that the relation $\mathcal{R}$ exists between two entities A and B.

2.2 A Motivating Example

To illustrate our method consider the example shown in Table 1. The leaderOf relation exists between a country and its current leader, whereas the ceoOf relation exist between a company and the chief executive officer of that company. Assuming that we are given contexts in which instances of the relation leaderOf occurs, we intend to train a relational classifier for the ceoOf relation. The two relations under consideration have very different distributions. Consequently, a relational classifier trained on one relation is not guaranteed to work well for the other relation. In Table 1, two contexts are provided for the leaderOf relation instance entity pair (*George Bush*, *U.S.*) and for the ceoOf relation instance entity pair (*Steve Jobs*, *Apple*).

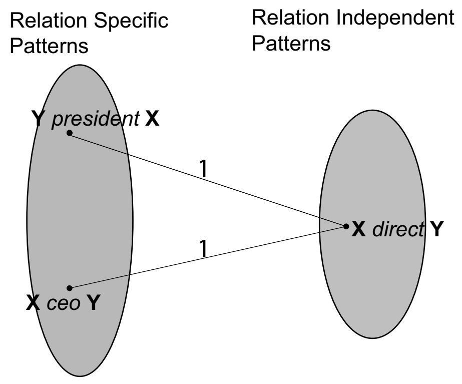

Figure 1: A bipartite graph between relation-specific patterns and relation-independent patterns shown in Table 1

To represent a semantic relation, we extract lexical and syntactic patterns as described later in Section 2.3. To illustrate our example, we assume that we extract the lexical patterns shown within brackets alongside with contexts in Table 1. The pattern, **X** *direct* **Y** appears in both relation types. We designate such patterns as *relation-independent* (RI) patterns. However, patterns such as **Y** *president* **X** and **X** *ceo* **Y** appear in only one of the two relation types. We designate such patterns as *relation-specific* (RS) patterns. For relation adaptation, we assume that we have sufficiently numerous source relation entity pairs, but only a few entity pairs for the target relation. Therefore, it is particularly challenging to learn proper weights for the target relation-specific patterns such as **X** *ceo* **Y**. However, relation-specific patterns in the target relation are extremely useful when determining whether a particular entity pair belongs to the target relation.

As a solution to this *mismatch* between source and target relation-specific patterns, we propose a method to find a mapping between source and target relations using relation-independent patterns as pivots. First, Figure 1 shows that we create a bipartite graph between relation-specific patterns and relation-independent patterns. Each pattern is represented as a vertex in the bipartite graph. Two vertices are connected by a weighted undirected edge if the corresponding patterns are extracted from the same entity pair. For instance, in Table 1, the two patterns **X** *direct* **Y** and **Y** *president* **X** are extracted from contexts for the entity pair (*George Bush*, *U.S.*). Therefore, those patterns are connected by an edge in the bipartite graph portrayed in Figure 1. Similarly, the relation specific pattern **X** *ceo* **Y** and the relation independent pattern **X** *direct* **Y** are connected by an edge because those two patterns are extracted from the contexts for the same entity pair (**Steve Jobs**, **Apple**). Next, we perform spectral clustering on this bipartite graph to compute a latent mapping between relation-specific and relation-independent patterns. This mapping is then used to project feature vectors to train a relation classifier.

2.3 Relation Representation

The contexts in which two entities A and B co-occur on the Web provide useful clues to the relations existing between

Table 1: Example contexts for two relation types: leaderOf and ceoOf. Entities between which the specified relation exist are marked in **boldface**. Words that contribute to important lexical patterns are shown in *italic*. Some lexical patterns extracted by the proposed method are shown within squared brackets.

leaderOf (source relation)	ceoOf (target relation)
President **George Bush** directed **U.S.** to an unnecessary war against Iraq. [**X** *direct* **Y**]	**Steve Jobs** personally directs **Apple** and makes final decisions on various UI designs. [**X** *direct* **Y**]
U.S. president *George Bush* attended the G8 summit last month. [**Y** *president* **X**]	**Steve Jobs** is the CEO of **Apple**, which he co-founded in 1976. [**X** *ceo* **Y**]

those entities. We use the term *context* to refer a window of text in which two entities co-occur. A context might not necessarily be a complete sentence. Retrieving contexts in which two entities co-occur has been studied in previous work on relation extraction [Banko and Etzioni, 2008; Bollegala *et al.*, 2009]. In our work, we assume that we are provided with contexts in which entities co-occur and only specifically examine the relation adaptation problem.

Given a pair of entities (A, B), the first step is to express the relation between A and B using some feature representation. Lexical or syntactic patterns have been successfully used in numerous natural language processing tasks involving relation extraction such as extracting hypernyms [Hearst, 1992]. We use the subsequence pattern extraction algorithm proposed by Bollegala et al. [2010] to extract lexical and syntactic patterns from contexts. Because of the limited availability of space we omit the details of this algorithm. Subsequence patterns are particularly useful to represent semantic relations between entities because they consider all the words in a context, and is *not* limited to extracting patterns only from the mid-fix (i.e., the portion of text in a context that appears between a pair of entities). Moreover, the consideration of gaps enables us to capture relations between entities located at some distance in a context. All lexical and syntactic patterns extracted from all contexts in which two entities A and B co-occur are arranged in a pattern frequency vector $\boldsymbol{x}_{AB}$ to represent the entity pair (A, B). The elements of $\boldsymbol{x}_{AB}$ correspond to, $f(\rho, A, B)$, the total number of times a pattern ρ is extracted from contexts in which A and B co-occur. It is analogous to the term frequency vector used in information retrieval.

We propose a strategy for selecting relation independent patterns using the entropy of a pattern over the distribution of entity pairs. The proposed strategy is inspired by the fact that if a pattern is relation-independent, then its distribution over the entity pairs tends to become more uniform. However, if a pattern is relation-specific, then its distribution is concentrated over a small set of entity pairs that belong to a specific relation type. The entropy of a pattern increases as its distribution becomes more uniform. The entropy, $H(\rho)$, of a pattern ρ is computed as

$$H(\rho) = \sum_{\mathcal{R} \in \Omega} \sum_{(A,B) \in \mathcal{R}} p(\rho, A, B) \log_2 p(\rho, A, B). \qquad (1)$$

Here, the joint probability between a pattern ρ and an entity pair (A, B) is given as

$$p(\rho, A, B) = \frac{f(\rho, A, B)}{\sum_{\rho \in \Phi} \sum_{\mathcal{R} \in \Omega} \sum_{(A,B) \in \mathcal{R}} f(\rho, A, B)}. \qquad (2)$$

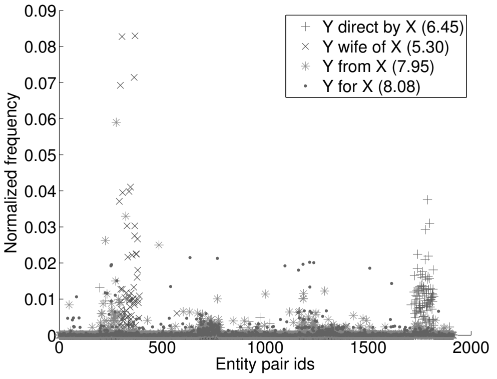

Figure 2: Distributions of four patterns over entity pairs. Entropies are shown within brackets.

Figure 2 presents an example in which we plot the distributions over entity pairs (numeric ids are assigned to entity pairs and grouped by their relation types for illustrative purposes) for four lexical patterns. From Figure 2, it is apparent that relation-specific patterns such as **Y** *directed by* **X** (directed relation), and **Y** *wife* **X** (isMarriedTo relation) are concentrated over a small set of entity pairs, whereas relation-independent patterns such as **Y** *from* **X**, and **Y** *for* **X** are distributed over a large set of entity pairs. Consequently, relation-independent patterns have higher entropy values than relation-specific patterns do. Note that the number of entity pairs must be balanced across different relation types prior to computing pattern entropy using Equation 1.

For the set of all patterns Φ (total no. of patterns, $|\Phi| = n$) extracted for all entity pairs in source and target relations (i.e. Ω), we use pattern entropy to identify a set $\Phi_{RS} \subseteq \Phi$ of relation-specific patterns, and a set $\Phi_{RI} \subseteq \Phi$ of relation-independent patterns. Here, $\Phi_{RS} \cup \Phi_{RI} = \Phi$ and $\Phi_{RS} \cap \Phi_{RI} = \phi$. We construct a bipartite graph, $G = (V_{RS} \cup V_{RI}, E)$ between relation-specific and relation-independent patterns to represent the intrinsic relationship between those patterns. Each vertex in V_{RS} corresponds to a relation-specific pattern, and each vertex in V_{RI} corresponds to a relation-independent pattern. A vertex in V_{RS} (corresponding to a relation-specific pattern $\rho_i \in \Phi_{RS}$) is connected to a vertex in V_{RI} (corresponding to a relation-independent pattern $\rho_j \in \Phi_{RI}$) by an undirected edge $e_{ij} \in E$. Note that there are no intra-set edges connecting vertices in V_{RS} and V_{RI}. Moreover, each edge $e_{ij} \in E$ is associated with a non-negative weight m_{ij}, that measures the strength of association between the corresponding patterns ρ_i and ρ_j. We set m_{ij} to the number of different entity pairs from which

Algorithm 1 Mapping patterns extracted from source and target relations to a lower-dimensional space.

```
Input: An edge-weight matrix, $\mathbf{M} \in \mathbb{R}^{(n-l)\times l}$ of a bipartite graph
    $G(V_{RS} \cup V_{RI}, E)$, and the number of clusters (latent dimensions) $k$.
Output: A projection matrix, $\mathbf{U} \in \mathbb{R}^{n\times k}$.
1: Compute the affinity matrix, $\mathbf{A} \in \mathbb{R}^{n\times n}$, of the bipartite graph
   $G$ as $\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{M}^\top & \mathbf{0} \end{bmatrix}$.
2: Compute the Laplacian, $\mathbf{L}$, of the bipartite graph $G$ as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$,
   where the diagonal matrix $\mathbf{D}$ has elements $D_{ii} = \sum_j A_{ij}$, and $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the unit matrix.
3: Find the eigenvectors corresponding to the $k$ smallest eigenvalues of $\mathbf{L}$, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$,
   and arrange them in columns to form the projection matrix $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k] \in \mathbb{R}^{n\times k}$.
4: return U
```

both ρ_i and ρ_j are extracted. Edge weights m_{ij} are represented collectively by an edge-weight matrix $\mathbf{M}$ of the bipartite graph G. Figure 1 portrays a bipartite graph constructed from the example shown in Table 1.

2.4 Relational Mapping

In relation adaptation, we assume that: (1) if two relation-specific patterns are connected to many common relation-independent patterns, then those relation-specific patterns must be mutually similar, (2) if two relation-independent patterns are connected to many common relation-specific patterns, then those relation-independent patterns must be mutually similar, and (3) there exist a lower-dimensional latent space in which similar patterns in the original space are located close together in this lower-dimensional space. Under those assumptions, we can use spectral graph theory to find a latent mapping between patterns extracted for source and target relation types, as shown in Algorithm 1.

In previous work on spectral clustering [Ding and He, 2004], it has been shown that the k smallest eigenvectors of the Laplacian matrix can be used to cluster a set of data points by mapping them into a k dimensional space spanned by those eigenvectors. Moreover, the k smallest eigenvectors act as the continuous solution of the cluster membership indicators. Consequently, we arrange the k smallest eigenvectors, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ in columns to construct a projection matrix $\mathbf{U}$. Once the projection matrix $\mathbf{U}$ is computed as described in Algorithm 1, we use it to project entity pairs that belong to different relation types into a common latent subspace. Specifically, for an entity pair (A, B) represented by a pattern-frequency vector $\boldsymbol{x}_{AB}$, its projection into the k-dimensional latent space is given as $\mathbf{U}\boldsymbol{x}_{AB}$.

In relation adaptation, the number of target relation training instances (entity pairs) is significantly smaller than that of the source relations. Given such an unbalanced training dataset, most supervised classification algorithms treat the minority class (target relation) instances as noise or outliers. Therefore, learning a classifier for a target relation type which has only a few instances is difficult in practice. To overcome this problem, we present a one-sided under-sampling algorithm (Algorithm 2), which first selects a subset of the

Algorithm 2 One-sided under-sampling algorithm to select a subset of source relation instances.

```
Input: Set $\Lambda$ that contains entity pairs for all source relations $\mathcal{S}_1, \ldots, \mathcal{S}_N$ and the target relation $\mathcal{T}$.
Output: A set $\Gamma \subseteq \Lambda$.
1: Initialize $\Gamma$ to the set containing all entity pairs of $\mathcal{T}$.
2: Randomly select an entity pair from each source relation $\mathcal{S}_i$ and add to $\Gamma$.
3: Classify $\Lambda$ with the 1-NN rule using the instances in $\Gamma$, and compare the assigned relation labels with the original ones.
4: Move all misclassified instances from $\Lambda$ into $\Gamma$.
5: return $\Gamma$
```

source relation training data and then uses that subset to train a multi-class classifier. One-sided under-sampling methods have been used to select a subset of the majority class in previous work investigating the problem of machine learning with unbalanced datasets [Provost, 2000].

Algorithm 2 takes a set, $\Lambda = \{(A, B)|(A, B) \in \mathcal{R}, \forall\mathcal{R} \in \Omega\}$, of all the entity pairs for source and target relations and creates a set $\Gamma \subseteq \Lambda$ that contains all target relation entity pairs and a subset of the source relation entity pairs. In Algorithm 2, we first select all target relation entity pairs as the set Γ (Line 1). In relation adaptation problem setting, the number of target relation entity pairs is small and we do not sample from the target relation (hence the name *one-sided* sampling). Next, we randomly select an entity pair from each source relation type and include those entity pairs in Γ (Line 2). We then use the single nearest neighbor (1-NN) rule to classify the entity pairs in Λ using the entity pairs in Γ as the labeled instances (Line 3). We use the Euclidean distance between the projected feature vectors to identify the nearest neighbors. All misclassified entity pairs in Λ are then moved to Γ (Line 4). After this operation, we obtain a set Γ that is consistent with the set Λ, although it contains fewer source domain entity pairs, thereby decreasing the imbalance between source and target relation entity pairs. In general, any classification algorithm is useful to train from the feature vectors. For simplicity, we use multi-class logistic regression as our classifier[1].

3 Experiments

To evaluate the proposed method, we select 20 relation types from the YAGO ontology[2]. For each selected relation, we randomly selected 100 entity pairs listed for that relation in the YAGO ontology. Overall, the dataset contains 2000 (20 relations $\times$ 100 instances) entity pairs. We use the Yahoo BOSS search API[3] to download contexts for the entity pairs in the dataset. Specifically, we construct numerous contextual queries that include the two entities in an entity pair and download snippets that contain those entities using the method proposed in [Bollegala *et al.*, 2009]. On average, we have ca. 7000 snippets for a pair of entities in the dataset. The dataset and the source code for the proposed method is

[1] `http://www.chokkan.org/software/classias/`
[2] `http://www.mpi-inf.mpg.de/yago-naga/yago/`
[3] `http://developer.yahoo.com/search/boss/`

Table 2: Macro-average results for various methods.

Method	Precision	Recall	F
Random	7.25	7.33	7.24
RS patterns	77.08	29.99	41.41
RI patterns	81.38	40.22	51.40
All patterns	79.34	37.11	47.94
Projected	78.48	33.56	44.86
Comb. (all patterns + projected)	84.21	45.11	56.99
RS patterns + Samp.	83.58	38.44	49.78
RI patterns + Samp.	75.60	45.11	54.83
All patterns + Samp.	80.94	47.11	57.62
Projected + Samp.	72.07	37.33	47.61
Jiang [Jiang, 2009]	81.06	44.89	55.62
Comb. + Samp. (**PROPOSED**)	**86.47**	**51.78**	**62.77**

publicly available[4].

We extract $68,822$ patterns for the entity pairs in the dataset and use those for the remainder of the experiments described in the paper (i.e. $n = 68822$). We then use the entropy-based relation-independent pattern selection criterion and select the top 1000 ranked patterns as relation-independent patterns (i.e. $l = 1000$). The remaining $67,822$ patterns are selected as relation-specific patterns. Next, we apply Algorithm 1 on the created bipartite graph to compute feature vector projections. We set the number of clusters to $k = 1000$ in our experiments [5].

For each relation type $\mathcal{R}$, we randomly allocated its 100 instances (entity pairs) into three groups: 60 instances as training instances when $\mathcal{R}$ is a source relation, 10 instances as training instances when $\mathcal{R}$ is a target relation, and 30 instances as test instances for $\mathcal{R}$. For each target relation type, therefore we have 1140 (19×60) source relation training instances and 10 target relation training instances, which well simulates the problem setting in relation adaptation. We repeat the above-described data splitting and report the average results of 5 random times. To evaluate the performance of a relation adaptation method, we select one relation type in the dataset as a target relation and train a multi-class classifier as described in Section 2. We use macro-averaged precision, recall, and F-measure over the 20 relation types as the evaluation metrics.

For comparison, Table 2 presents results obtained using the proposed method with 10 baselines and a previously proposed weakly-supervised relation extraction system on our dataset. The **Random** baseline randomly infers a relation for an entity pair out of the 20 relation types in the dataset. The **RS patterns**, **RI patterns**, and **All patterns** baselines respectively simulates the cases where we only use relation-specific, relation-independent, and all patterns as features to represent an entity pair to train a multi-class classifier. The **Projected** baseline use Algorithm 1 to compute a projection matrix $\mathbf{U}$ and project a feature vector $\boldsymbol{x}_{AB}$ for an entity pair (A, B) to a lower-dimensional vector $\mathbf{U}\boldsymbol{x}_{AB}$. This baseline is expected to show the level of performance we would have obtained if we had used only the lower-dimensional representation. The **Comb** (combined) baseline uses both the original feature vectors as well as their projection into the k-dimensional latent space and train a multi-class classifier using those augmented feature vectors. All of the above-mentioned baselines are implemented with and without one-sided under-sampling (indicated by **+Samp**).

Jiang is the current state-of-the-art cross-domain relation classification method [Jiang, 2009]. In this method, first, an entity-pair is represented as a set of lexical and syntactic features. Second, a multi-class logistic regression model is trained using those feature vectors. Some features are shared across different relations and the weight parameters for those features are learned in a joint fashion. We ran the original implementation that is publicly available[6] on our dataset.

From Table 2, we see that the proposed method (**PROPOSED**) has the best macro-average precision, recall, and F-measure among all the different methods. In particular, improvement against the previously proposed state-of-the-art weakly-supervised relation extraction method [Jiang, 2009] is statistically significant (paired t-test with $p < 0.05$ inferred as significant). The **Random** baseline on this balanced dataset only yields a very low F-score of 7.25. The **RI patterns** baseline that uses only relation-independent patterns outperforms the **RS patterns** baseline that uses only relation-specific patterns. This result is particularly interesting considering that only 1000 relation-independent patterns exist, whereas $67,822$ relation-specific patterns exist. Even with a few relation-independent patterns, we can learn a better relational adaptation model than using many relation-specific patterns. Relation-specific patterns occur in only few relation types. Therefore, a model trained using those patterns do not generalize well to a novel target relation type. Moreover, among the 1000 relation-independent patterns, we have 454 lexical patterns and 546 syntactic patterns, whereas among the $67,822$ relation-specific patterns we have $65,771$ lexical patterns and 2051 syntactic patterns. Considering the fact that part-of-speech tags abstract individual words, it is not surprising that a major proportion of the relation-independent patterns are indeed syntactic patterns.

Using all the patterns (**All patterns**) performs slightly worse than when using only relation-independent patterns. One reason for this is that the overall performance of the **All patterns** baseline is dominated by the numerous relation-specific patterns, which adapt poorly to target relations. Pattern entropy-based relation-independent pattern selection method might not detect all relation-independent patterns thereby introducing some noise in the created bipartite graph. This might in turn lead to incorrect projections. Consequently, using only the **Projected** features is not satisfactory. However, by augmenting the original features to the projected features (**Comb**), this problem can be overcome. Moreover, by sampling, we consistently improve all the baselines. In fact, the proposed method, which uses augmented feature vectors with sampling, shows a 6 percent improvement over not using sampling (**Comb**). This underscores the importance

[4]`www.iba.t.u-tokyo.ac.jp/~danushka/RA/`

[5]The performance of the proposed method is insensitive across a wide range of parameter values

[6]`www.mysmu.edu/faculty/jingjiang/software/DALR.html`

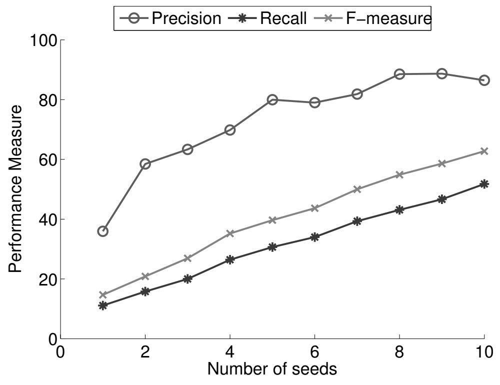

Figure 3: Effect of varying the number of target relation training instances.

of selecting a subset of source relation instances when training a classifier for a target relation. In relation adaptation, the number of source relation labeled instances significantly outperforms that for a target relation. Without proper sampling, any information related to the target relation, encoded in the small number of target relation instances, gets "washed-out" during training.

Figure 3 depicts the performance of the proposed method as a function of the number of training instances for the target relation. The figure shows that the performance increases steadily with the number of training instances we have for the target relation. This result emphasizes the importance of target relation instances for relation adaptation. It also justifies our decision to retain all target relation instances during sampling.

4 Related Work

Bootstrapping methods [Pasca *et al.*, 2006] to relation extraction are attractive because they require markedly fewer training instances than supervised approaches do. Bootstrapping methods are initialized with a few instances of the target relation. During subsequent iterations of the bootstrapping process, new extraction patterns are discovered and are used to extract new instances. However, the quality of the extracted relations depends heavily upon the initial seeds provided to the bootstrapping system [Kozareva and Hovy, 2010]. Different from bootstrapping, we not only use target relation seeds, but also use the existing training instances for numerous source relations to train a robust relation extractor for a target relation.

Open Information Extraction (Open IE) [Banko *et al.*, 2007] is a domain-independent information extraction paradigm. Open IE systems are initialized with a few manually provided domain independent extraction patterns. In contrast, we learn the domain-independent relation patterns using source and target relation instances. Moreover, open IE systems attempt to extract all relations that exist in a corpus; users cannot specify in advance which relation types (targets) they want to extract. Therefore, it is not guaranteed that we will be able to extract instances for the target relation type in which we are interested.

Jiang [2009] proposed a multi-task transfer learning method to train a relation extraction system. She models commonality among different relation types by a shared weight vector. Next, a multi-class logistic regression model is trained using both source and target relations. To determine which features to share between relation types, they propose an alternating optimization procedure as well as several heuristics. Unlike our proposed method, they do not compute a projection of features among relation types. Consequently, as shown in our experiments, our proposed method outperforms their multi-task transfer learning algorithm.

5 Conclusion

We proposed and investigated a method to learn a relational classifier for a target relation using multiple source relations. Both feature projection and sampling positively contribute to the proposed method. In future, we intend to apply the proposed method to other classification tasks.

References

[Banko and Etzioni, 2008] M. Banko and O. Etzioni. The tradeoffs between traditional and open relation extraction. In *ACL'08*, pages 28–36, 2008.

[Banko *et al.*, 2007] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI'07*, pages 2670–2676, 2007.

[Bollegala *et al.*, 2009] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *WWW'09*, pages 651 – 660, 2009.

[Bollegala *et al.*, 2010] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *WWW'10*, pages 151 – 160, 2010.

[Ding and He, 2004] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML'04*, pages 225 – 232, 2004.

[GuoDong *et al.*, 2005] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *ACL'05*, pages 427 – 434, 2005.

[Hearst, 1992] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING'92*, pages 539–545, 1992.

[Jiang, 2009] Jing Jiang. Multi-task transfer learning for weakly-supervised relation extraction. In *ACL'09*, pages 1012–1020, 2009.

[Kozareva and Hovy, 2010] Zornista Kozareva and Eduard Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *NAACL'10*, 2010.

[Pasca *et al.*, 2006] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *Proc. of AAAI'06*, pages 1400–1405, 2006.

[Provost, 2000] Foster Provost. Machine learning from imbalanced data sets. In *AAAI'00 Workshop on Imbalanced Data Sets*, 2000.