Relation Adaptation: Domain Adaptation of Relation Extraction Systems

Danushka Bollegala
danushka@iba.t.u-.tokyo.ac.jp

Yutaka Matsuo
matsuo@biz-model.t.u-tokyo.ac.jp

Mitsuru Ishizuka
ishizuka@i.u-tokyo.ac.jp

The University of Tokyo
7-3-1, Hongo, Tokyo, 113-8656, Japan

Abstract

Supervised relation extraction systems which are trained to extract a particular relation type (source relation) does not accurately extract a new type of a relation (target relation) for which it has not been trained. 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.

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 information retrieval (IR). For example, to improve coverage in information retrieval, a query about a particular person can return documents describing the various semantic relations that the person under consideration has with other related entities.

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 (Bunescu and Mooney, 2005; Culotta and Sorensen, 2004; 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. We study *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 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 Method

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). Following the previous work on relation extraction between entities, we use lexical and syntactic patterns extracted from the contexts in which two entities co-occur to represent the semantic relation that exists between those entities. First, we download Web snippets for the AND query of the two entities A and B. Next, we replace A and B respectively by two variables X and Y. Finally, we generate subsequence patterns that contain both X and Y. We extract both lexical and syntactic subsequence patterns using an improved version of the subsequence pattern mining algorithm proposed by Bollegala et al. (2010).

Once we express the relations that exist between entities using lexical and syntactic patterns, we compute the correspondence between patterns that express different semantic relations. First, we must identify which patterns are specific to a particular relation type. 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.

Figure 1 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 1, 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.

We construct a bipartite graph, $G = (V_{RS} \cup V_{RI}, E)$ between relation-specific (V_{RS}) and relation-independent (V_{RI}) patterns to represent the intrinsic relationship between those patterns. Each

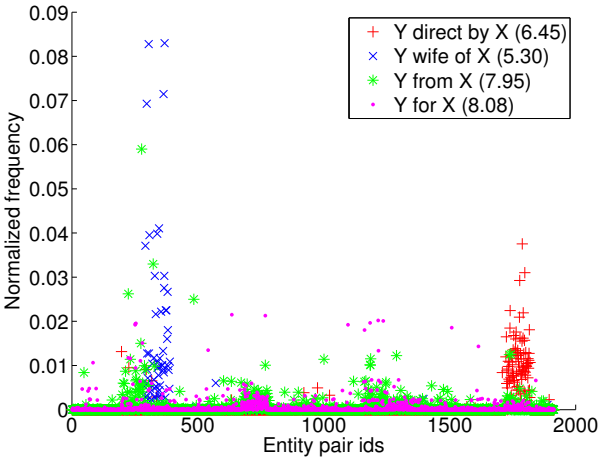

Figure 1: Distributions of four patterns over entity pairs. Entropies are shown within brackets.

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) is connected to a vertex in V_{RI} (corresponding to a relation-independent pattern) 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 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. For simplicity, we use the number of different entity pairs from which two patterns are extracted as the edge-weighting measure.

Given as input an edge-weight matrix $\mathbf{M}$ for the bipartite graph G and dimensionality $k(< n)$ of the latent space, Algorithm 1 returns a projection matrix $\mathbf{U}$ from the original n dimensional pattern space to a k dimensional latent space. The (i, j) element of the edge-weight matrix $\mathbf{M}$ represents the weight of the edge that connects a relation-specific pattern ρ_i to a relation-independent pattern ρ_j.

The low-dimensional projection reduces the mismatch between patterns in source and target relation types, thereby enabling us to train a classifier for the target relation type using labeled entity pairs for both source and target relation types. However, we must

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** $\mathbf{U}$

overcome two challenges before we can use the projected vectors to train a classifier for a target relation type: *loss of information because of imperfect projections*, and *imbalance between source and target relation training datasets*. Next, we discuss each challenge in detail and propose solutions to overcome them.

First, the criterion for selecting relation-independent and relation-specific patterns might not be perfect, thereby introducing some noise to the created bipartite graph. For that reason, the computed projection matrix might not be perfect. To compensate for the loss of information because of imperfect feature projection, we augment all the patterns in the original vector $\boldsymbol{x}_{AB} \in \mathbb{R}^{n\times 1}$ to the projection $\mathbf{U}\boldsymbol{x}_{AB} \in \mathbb{R}^{k\times 1}$ to construct a new representation $\tilde{\boldsymbol{x}}_{AB} \times \mathbb{R}^{(n+k)\times 1}$ for an entity pair (A, B) as

$$\tilde{\boldsymbol{x}}_{AB} = [\boldsymbol{x}_{AB}, \lambda\mathbf{U}\boldsymbol{x}_{AB}]. \quad (1)$$

The single scalar parameter λ is useful to balance the tradeoff between original and projected features in the new representation. Using a set of heldout data, we set λ such that the average L_1 norm on the source relation projection vectors $\mathbf{U}\boldsymbol{x}$ is equal to that of the original vectors $\boldsymbol{x}$. This new representation retains all the features (pattern frequencies) in the original vector in addition to the projected features, thereby overcoming any disfluencies attributable to potential imperfect projections.

Second, 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 use one-sided under-sampling which first selects a subset of the 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 (Kubat and Matwin, 1997; Provost, 2000).

3 Experiments and Results

To evaluate the proposed method, we select 20 relation types that have been used frequently for evaluating relation extraction systems (Agichtein and Gravano, 2000; Banko et al., 2007; Bollegala et al., 2010) from the YAGO ontology[1]. 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. The YAGO ontology has a high level of manually confirmed accuracy. It is suitable as a gold standard for evaluating relations between entity pairs on the Web (Suchanek et al., 2007).

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.

[1] http://www.mpi-inf.mpg.de/yago-naga/yago/

Table 1: Macro-average results for various methods.

Method	F-measure
Random	7.24
RS patterns	41.41
RI patterns	51.40
All patterns	47.94
Projected	44.86
Combined (all patterns + projected)	56.99
RS patterns + Sampling	49.78
RI patterns + Sampling	54.83
All patterns + Sampling	57.62
Projected + Sampling	47.61
Jiang (Jiang, 2009)	55.62
Combined + Sampling (**PROPOSED**)	**62.77**

From Table 1, we see that the proposed method has the best macro-average 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 **RI patterns** baseline that uses only relation-specific patterns. Using all the patterns (i.e. **All patterns** baseline) 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. There can be errors in identifying relation-independent patterns using strategies such as mutual information, which engender some noise in the constructed bipartite graph. Consequently, using only the **Projected** features is not satisfactory. However, by augmenting the original features to the projected features (i.e. **Combined** baseline), this problem can be overcome. Next, we evaluate the effect of the one-sided undersampling on top of the numerous baselines discussed above. From Table 1, it is apparent that, by sampling, we consistently improve all the baselines: **RS patterns**, **RI patterns**, **All patterns**, and **Projected**. In fact, the proposed method, which uses augmented feature vectors with sampling, shows a 6 percent improvement over not using sampling (i.e. **Combined** baseline).

4 Conclusion

We proposed and investigated a method to learn a relational classifier for a target relation using multiple source relations. Our experimental results show that the proposed method significantly outperforms 10 baselines and a previously proposed weakly-supervised relation extraction method on a dataset that contains 2000 entity pairs for 20 different relation types. Both feature projection and sampling positively contribute to the proposed method. Moreover, the proposed method performs consistently under different parameter settings. In future studies, we intend to apply the proposed method to other classification tasks.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *ICDL'00*.

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

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

R. Bunescu and R. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proc. of EMNLP'05*, pages 724 – 731.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. of ACL'04*, pages 423–429.

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

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

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

Miroslav Kubat and Stan Matwin. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML'97*, pages 179 – 186.

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

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *WWW'07*.