Robust Domain Adaptation for Relation Extraction via Clustering Consistency

Minh Luan Nguyen[†]*, **Ivor W. Tsang**[‡], **Kian Ming A. Chai**[§], **and Hai Leong Chieu**[§]

[†]Institute for Infocomm Research, Singapore
[‡]Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Australia
[§]DSO National Laboratories, Singapore

mlnguyen@i2r.a-star.edu.sg, ivor.tsang@gmail.com
{ckianmin,chaileon}@dso.org.sg

Abstract

We propose a two-phase framework to adapt existing relation extraction classifiers to extract relations for new target domains. We address two challenges: negative transfer when knowledge in source domains is used without considering the differences in relation distributions; and lack of adequate labeled samples for rarer relations in the new domain, due to a small labeled data set and imbalance relation distributions. Our framework leverages on both labeled and unlabeled data in the target domain. First, we determine the relevance of each source domain to the target domain for each relation type, using the consistency between the clustering given by the target domain labels and the clustering given by the predictors trained for the source domain. To overcome the lack of labeled samples for rarer relations, these clusterings operate on both the labeled and unlabeled data in the target domain. Second, we trade-off between using relevance-weighted source-domain predictors and the labeled target data. Again, to overcome the imbalance distribution, the source-domain predictors operate on the unlabeled target data. Our method outperforms numerous baselines and a weakly-supervised relation extraction method on ACE 2004 and YAGO.

1 Introduction

The World Wide Web contains information on real-world entities, such as persons, locations and organizations, which are interconnected by various semantic relations. Detecting these relations between two entities is important for many tasks on the Web, such as information retrieval (Salton and McGill, 1986) and information extraction for question answering (Etzioni et al., 2008). Recent work on relation extraction has demonstrated that supervised machine learning coupled with intelligent feature engineering can provide state-of-the-art performance (Jiang and Zhai, 2007b). However, most supervised learning algorithms require adequate labeled data for every relation type to be extracted. Due to the large number of relations among entities, it may be costly to annotate a large enough set of training data to cover each relation type adequately in every new domain of interest. Instead, it can be more cost-effective to adapt an existing relation extraction system to the new domain using a small set of labeled data. This paper considers relation adaptation, where a relation extraction system trained on many source domains is adapted to a new target domain.

There are at least three challenges when adapting a relation extraction system to a new domain. First, the same semantic relation between two entities can be expressed using different lexical or syntactic patterns. For example, the acquisition of company A by company B can be expressed with "B bought over by A", "A buys B" and "A purchases B". To extract a relation, we need to capture the different ways in which it can be expressed across different open domains on the Web.

Second, the emphasis or interest on the different relation types varies from domain to domain. For example, in the organization domain, we may be more interested in extracting relations such as *locatedIn* (between a company and a location) and *founderOf* (between a company and a person),

[*]The work is done while Nguyen was a research staff in Nanyang Technological University, Singapore.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 807–817,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

whereas in the person domain we may be more interested in extracting relations such as *liveIn* (between a person and a location) and *workAt* (between a person and a company). Therefore, although the two domains may have the same set of relations, they probably have different marginal distributions on the relations. This can produce a negative transfer phenomenon (Rosenstein et al., 2005), where using knowledge from other domains degrades the performance on the target domain. Hence, when transferring knowledge from multiple domains, it is overly optimistic to believe that all source domains will contribute positively. We call a source domain *irrelevant* when it has no or negative contribution to the performance of the target domain. One example is named entities extraction adaptation, where naïve transfer of information from a mixed-case domain with capitalization information (e.g., news-wire) to a single-case domain (e.g., conversational speech transcripts) will miss most names in the single-case domain due to the absence of case information, which is typically important in the mixed-case domain.

Third, the annotated instances for the target domain are typically much fewer than those for the source domains. This is primarily due to the lack of resources such as raw target domain documents, time, and people with the expertise. Together with imbalanced relation distributions inherent in the domain, this can cause some rarer relations to constitute only a very small proportion of the labeled data set. This makes learning a relation classifier for the target domain challenging.

To tackle these challenges, we propose a two-phase Robust Domain Adaptation (RDA) framework. In the first phase, Supervised Voting is used to determine the relevance of each source domain to each region in the target domain, using both labeled and unlabeled data in the target domain. By using also unlabeled data, we alleviate the lack of labeled samples for rarer relations due to imbalanced distributions in relation types.

The second phase uses the relevances determined the first phase to produce a reference predictor by weighing the source-domain predictors for each target domain sample separately. The intention is to alleviate the effect of mismatched distributions. The final predictor in the target domain is trained on the labeled target domain data while taking reference from the reference predictions on the unlabeled target domain data. This ensures reasonable predictive performance even when all the source domains are irrelevant and augments the rarer classes with examples in the unlabeled data. We compare the proposed two-phase framework with state-of-the-art domain adaptation baselines for the relation extraction task, and we find that our method outperforms the baselines.

2 Related Work

Relation extraction is usually considered a classification problem: determine if two given entities in a sentence have a given relation. Kernel-based supervised methods such as dependency tree kernels (Culotta and Sorensen, 2004), subsequence kernels (Bunescu and Mooney, 2006) and convolution tree kernels (Qian et al., 2008) have been rather successful in learning this task. However, purely supervised relation extraction methods assume the availability of sufficient labeled data, which may be costly to obtain for new domains. We address this by augmenting a small labeled data set with other information in the domain adaptation setting.

Bootstrapping methods (Zhu et al., 2009; Agichtein and Gravano, 2000; Xu et al., 2010; Pasca et al., 2006; Riloff and Jones, 1999) to relation extraction are attractive because they require fewer training instances than supervised approaches. Bootstrapping methods are either initialized with a few instances (often designated as seeds) of the target relation (Zhu et al., 2009; Agichtein and Gravano, 2000) or a few extraction patterns (Xu et al., 2010). In subsequent iterations, new extraction patterns are discovered, and these are used to extract new instances. The quality of the extracted relations depends heavily on the seeds (Kozareva and Hovy, 2010). Different from bootstrapping, we not only use labeled target domain data as seeds, but also leverage on existing source-domain predictors to obtain a robust relation extractor for the target domain.

Open Information Extraction (Open IE) (Etzioni et al., 2008; Banko et al., 2008; Mesquita et al., 2013) is a domain-independent information extraction paradigm to extract relation tuples from collected corpus (Shinyama and Sekine, 2006) and Web (Etzioni et al., 2008; Banko et al., 2008). Open IE systems are initialized with a few domain-independent extraction patterns. To create labeled data, the texts are dependency-parsed, and the domain-independent patterns on the parses form the basis for extractions. Recently, to reduce labeling effort for relation extraction, distant supervision (Mintz et al., 2009; Takamatsu et al., 2012; Min et al., 2013; Xu et al., 2013) has been proposed. This is an unsupervised approach that exploits textual features in large unlabeled corpora. In contrast to Open IE, we tune the relation patterns for a domain of interest, using labeled relation instances in source and target domains and unlabeled instances in the target domain.

Our work is also different from the multi-schema matching in database integration (Doan et al., 2003). Multi-schema matching finds relations between columns of schemas, which have the same semantic. In addition, current weighted schema matching methods do not address negative transfer and imbalance class distribution.

Domain adaptation methods can be classified broadly into weakly-supervised adaptation (Daume and Marcu, 2007; Blitzer et al., 2006; Jiang and Zhai, 2007a; Jiang, 2009), and unsupervised adaptation (Pan et al., 2010; Blitzer et al., 2006; Plank and Moschitti, 2013). In the weakly-supervised approach, we have plenty of labeled data for the source domain and a few labeled instances in the target domain; in the unsupervised approach, the data for the target domain are not labeled. Among these studies, Plank and Moschitti's is the closest to ours because it adapts relation extraction systems to new domains. Most other works focused on adapting from old to new relation types. Typical relation adaptation methods first identify a set of common features in source and target domains and then use those features as pivots to map source domain features to the target domain. These methods usually assume that each source domain is relevant to the task on the target domain. In addition, these methods do not handle the imbalanced distribution of relation data explicitly. In this work, we study how to learn the target prediction using only a few seed instances, while dealing with negative transfer and imbalanced relation distribution explicitly. These issues are seldom explored in relation adaptation.

3 Problem Statement

This section defines the domain adaptation problem and describes our feature extraction scheme.

3.1 Relation Extraction Domain Adaptation

Given two entities A and B in a sentence S, relation extraction is the task of selecting the relation y between A and B from a fixed set of c relation types, which includes the *not-a-relation* type. We introduce a feature extraction χ that maps the triple (A,B,S) to its feature vector x. Learning relation extraction can then be abstracted to finding a function p such that $p(\chi(A,B,S)) = p(x) = y$.

For adaptation, we have k source domains and a target domain. We shall assume that all domains have the same set of relation types. The target domain has a few labeled data $D_l = \{(x_i,y_i)\}_{i=1}^{n_l}$ and plenty of unlabeled data $D_u = \{(x_i)\}_{n_l+1}^{n_l+n_u}$, where n_l and n_u are the number of labeled and unlabeled samples respectively, x_i is the feature vector, y_i is the corresponding label (if available). Let $n = n_l + n_u$. For the sth source domain, we have an adequate labeled data set D^s. We define *domain adaptation* as the problem of learning a classifier p for relation extraction in the target domain using the data sets D_l, D_u and D^s, $s = 1,\ldots,k$.

3.2 Relational Feature Representation

We consider relation extraction as a classification problem, where each pair of entities A and B within a sentence S is a candidate relation instance. The contexts in which entities A and B co-occur provide useful features to the relations between them. 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 S. Retrieving contexts in which two entities co-occur has been studied in previous works investigating the relations between entities.

Given a pair of entities (A,B) in S, the first step is to express the relation between A and B with some feature representation using a feature extraction scheme χ. Lexical or syntactic patterns have been successfully used in numerous natural language processing tasks, including relation extraction. Jiang and Zhai (2007b) have shown that selected lexical and syntactic patterns can give good performance for relation extraction. Following their work[1], we also use lexical and syntactic patterns extracted from the contexts to represent the relations between entities. We extract features from a sequence representation and a parse tree representation of each relation instance. The details are as follows.

Entity Features Entity types and entity mention types are very useful for relation extraction. We

[1] The source code for extracting entity features is provided by the authors (Jiang and Zhai, 2007b).

use a subgraph in the relation instance graph (Jiang and Zhai, 2007b) that contains only the node presenting the head word of the entity A, labeled with the entity type or entity mention types, to describe a single entity attribute.

Sequence Features The sequence representation preserves the order of the tokens as they occur in the original sentence. Each node in the graph is a token augmented with its relevant attributes.

Syntactic Features The syntactic parse tree of the relation instance sentence can be augmented to represent the relation instance. Each node is augmented with relevant part-of-speech (POS) using the Python Natural Language Processing Tool Kit.

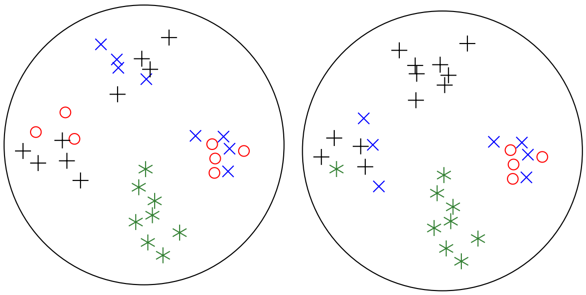
Each node in the sequence or the parse tree is augmented by an argument tag that indicates whether the node corresponds to entity A, B, both, or neither. The nodes that represent the argument are also labeled with the entity type, subtype and mention type. We trim the parse tree of a relation instance so that it contains only the most essential tree components based on constituent dependencies (Qian et al., 2008). We also use unigram features and bigram features from a relation instance graph.

4 Robust Domain Adaptation

In this section, we describe our two-phase approach, which comprises of a Supervised Voting scheme and a combined classifier learning phase.

4.1 Phase 1: Clustering Consistency via Supervised Voting

In this section, we use the concept of clustering consistency to determine the relevance of a source domain to particular regions in the target domain. Figure 1 illustrates this. There, both enclosing circles in the left and right figures denote the same input space of the target domain. There are four disjoint regions within the input space, located at the left, right, top and bottom of the space. There are four classes of labels: plus ($+$), cross ($\times$), circle ($\circ$) and asterisk ($*$). The labels in the left figure are given by a *preliminary predictor* in the target domain data, while the labels in the right figure are given by a predictor trained on the source domain data. Comparing the figures, we see the preliminary predictor and source domain predictor are consistent for the bottom and right regions, but are inconsistent for the top and left regions. This suggests that the source domain is very relevant for the bottom and right regions of the target input space, but less so for the top and left regions.



Target domain input space with transductive learning using labeled and unlabeled target domain data.

Target domain input space with labels from the predictor trained on the source domain data set.

Figure 1: Clustering consistency is used to determine the relevance of a source domain to a region in the target domain data. The bottom and right regions are more relevant than the top and left regions. See text for explanation.

To apply this idea to relation classification, we have to (i) partition the target domain input space into regions and (ii) assign preliminary labels for all the examples. We approximate the target domain input space with all the samples from D_l and D_u. With data from both the labeled and unlabeled data sets, we apply transductive inference or semi-supervised learning (Zhou et al., 2003) to achieve both (i) and (ii). By augmenting with unlabeled data D_u, we aim to alleviate the effect of imbalanced relation distribution, which causes a lack of labeled samples for rarer classes in a small set of labeled data. Briefly, the known labels in D_l are propagated to the entire target input space by encouraging label smoothness in neighborhoods. The next three paragraphs give more details.

At present, we assume a similarity matrix W, where W_{ij} is the similarity between the ith and the jth input samples in $D_l \cup D_u$. Matrix W then determines the neighborhoods. Let Λ be a diagonal matrix where the (i,i)th entry is the sum of the ith row of W. Let us also encode the the labeled data D_l in an n-by-c matrix H, such that $H_{ij} = 1$ if sample i is labeled with relation class j in D_l, and $H_{ij} = 0$ otherwise. Our objective is the c-dimensional relation-class indicator vector F_i for the ith sample, for every sample. This is achieved via a regularization framework (Zhou et al., 2003):

$$\min_{\{F_i\}_{i=1}^n} \left(\sum_{i,j=1}^n W_{ij} \left\| \frac{F_i}{\sqrt{\Lambda_{ii}}} - \frac{F_j}{\sqrt{\Lambda_{jj}}} \right\|^2 + \mu \sum_{i=1}^n \|F_i - H_i\|^2 \right).$$

This trades off two criteria: the first term encourages nearby samples (under distance metric W) to have the same labels, while the second encourages samples to take their labels from the labeled data. The closed-form solution is

$$F^* = (I - (1+\mu)^{-1} L)^{-1} H, \qquad (1)$$

where $L = \Lambda^{-1/2} W \Lambda^{-1/2}$; and the n-by-c matrix F^* is the concatenation of the F_is.

Using vector F_i^*, we now assign preliminary labels to the samples. For a sample i, we transform F_i^* into probabilities $p_i^1, p_i^2, \ldots, p_i^c$ using softmax. Our propagated label ℓ_i for sample i is then

$$\ell_i = \begin{cases} \textit{not-a-relation} & \text{if } (\max_j p_i^j) < \theta, \\ \arg\max_j p_i^j & \text{otherwise.} \end{cases} \qquad (2)$$

The second clause is self-evident, but the first needs further explanation. Because *not-a-relation* is a background or default relation type in the relation classification task, and because it has rather high variation when manifested in natural language, we have found it difficult to obtain a distance metric W that allows the *not-a-relation* samples to form clusters naturally using transductive inference. Therefore, we introduce the first clause to assign the *not-a-relation* label to a sample when there is no strong evidence for any of the positive relation types. The amount of evidence needed is quantified by the parameter $\theta > 1/c$. In addition, the second clause will also assign *not-a-relation* to a sample if that probability is the highest.

Next, we partition the data in $D_l \cup D_u$ into c regions, $R_1, R_2, \ldots, R_c$, corresponding to the c relation types. The intuition is to use the true label in D_l when available, or otherwise resort to using the propagated label. That is,

$$x_i \in \begin{cases} R_{y_i} & \text{if } x_i \in D_l, \\ R_{\ell_i} & \text{if } x_i \in D_u. \end{cases}$$

We now have the necessary ingredients to quantify the *clustering consistency* between a source domain and a region in the target domain. Intuitively, this is the agreement between the source-domain predictor and the preliminary predictor within the target domain. We use supervised voting in the following manner. For every source domain, say domain s, we first train a relation-type predictor p_s based on its training data D^s. Then, for every region R_j, we compute the relevance score $w_{s,j} = \sum_{x_i \in R_j} [\![p_s(x_i) = \ell_i]\!] / |R_j|$, where $[\![\cdot]\!]$ is the Iverson bracket.

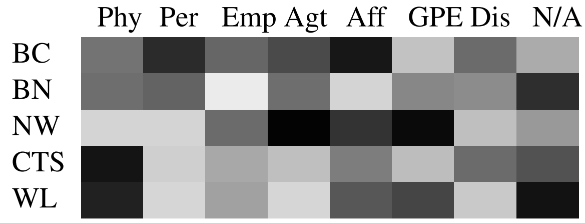


Figure 2: Heat map of the relevance scores $w_{s,j}$ between the target domain Usenet (UN) with the other domains on ACE 2004 data set. A lighter shade means a higher score, or more relevant. N/A refers to *not-a-relation*; for the other abbreviations, see the second paragraph in section 5.

Figure 2 shows the heat map of the relevance scores $w_{s,j}$ between the target domain Usenet (UN) with the other domains in the ACE 2004 corpus. We observe, for example, that the Broadcast News (BN) domain is more relevant in the Personal-Social region of the target domain than the Broadcast Conversation (BC) domain. These relevance scores will be used in the next phase of the framework to weigh the contributions of source-domain predictors to the eventual target-domain relation classifier.

4.2 Phase 2: Target Classifier Learning

The second phase uses both the weighted predictions from all sources and the target labeled data D_l to learn a relation classifier. This ensures that even when most of the source domains are irrelevant, the performance of our method is no worse than using the target-domain labeled data alone.

The previous phase has computed the relevance $w_{s,j}$ for source domain s in region R_j. We translate this to the relevance weight $u_{s,i}$ for an example x_i: if $x_i \in R_j$, then $u_{s,i} = w_{s,j}$. At our disposal from the previous phase are also k source-domain predictors p_s that have been trained on D^s. Combining and weighing the predictions from multiple sources, we obtain the *reference predic-*

tion $\hat{r}_{ji} = \sum_{s=1}^{k} u_{s,i}(2[\![p_s(x_i) = j]\!] - 1)$ for example x_i belonging to relation j, using the ± 1 encoding.

The relation classifier consists of c functions $f_1, \ldots, f_c$ using the one-versus-rest decoding for multi-class classification.[2] Inspired by the Domain Adaptive Machine (Duan et al., 2009), we combine the reference predictions and the labeled data of the target domain to learn these functions:

$$\min_{\{f_j\}_{j=1}^{c}} \sum_{j=1}^{c} \left\{ \frac{1}{n_l} \sum_{i=1}^{n_l} (f_j(x_i) - r_{ji})^2 + \gamma \|f_j\|_{\mathcal{H}}^2 + \frac{\beta}{2} \sum_{i=n_l+1}^{n} \|f_j(x_i) - \hat{r}_{ji}\|^2 \right\}, \quad (3)$$

where $r_{ji} = 2[\![y_i = j]\!] - 1$ is the ± 1 binary encoding for the i labeled sample belonging to relation j. Here, we have multiple objectives: the first term controls the training error; the second regularizes the complexity of the functions f_js in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$; and the third prefers the predicted labels of the unlabeled data D_l to be close to the reference predictions. The third term provides additional pseudo-training samples for the rarer relation classes, if these are available in D_u. Parameters β and γ govern the trade-offs between these objectives.

Let $K(\cdot, \cdot)$ be the reproducing kernel for $\mathcal{H}$. By the Representer Theorem (Smola and Scholkopf, 1998), the solution for Eq. 3 is linear in $K(x_i, \cdot)$: $f_j(x) = \sum_{i=1}^{n} \alpha_{ji} K(x_i, x)$. Putting this into Eq. 3, parameter vectors α_j are (Belkin et al., 2006):

$$\alpha_j^* = (JK + \gamma(n_l + \beta n_u)I)^{-1} J R_j. \quad (4)$$

Here, R_j is an $(n_l + n_u)$-vector, where $R_{ji} = r_{ji}$ if sample i belongs to the labeled set, and $R_{ji} = \hat{r}_{ij}$ if it belongs to the unlabeled set; and J is an $(n_l + n_u)$-by-$(n_l + n_u)$ diagonal matrix where the first n_l diagonal entries are ones and the rest are βs.

5 Experiments

We evaluate our algorithm on two corpora: Automatic Content Extraction (ACE) 2004 and YAGO[3]. Table 1 provides some statistics on them.

ACE 2004 consists of six domains: Broadcast Conversations (BC), Broadcast News (BN), Conversational Telephone Speech (CTS), Newswire (NW), Usenet (UN) and Weblog (WL). There are seven positive relation types: Physical (Phy), Personal/Social (Per), Employment/Membership/Subsidiary (Emp), Agent-Artifact (Agt), PER/ORG Affiliation (Aff), GPE Affiliation (GPE) and Discourse (Dis).

Table 1: Statistics on ACE 2004 and YAGO

Properties	ACE 2004	YAGO
# relation types	7	20
# candidate relations	48,625	68,822
# gold relations	4,296	2,000
# mentions per entity pair	6	11
% mentions with +ve relations	8.8%	21%

YAGO is an open information extraction data set. The relation types of YAGO are built from Wikipedia and WordNet, while the labeled text for YAGO is from Bollegala et al. (2011). It consists of twenty relation types such as *ceo_company*, *bornIn* and *isMarriedTo*, and each of them is considered as a domain in this work. YAGO is different from ACE 2004 in two aspects: there is less overlapping of topics, entity types and relation types between domains; and it has more relation mentions with 11 mentions per pair of entities on the average.

We used Collins parser (Collins, 1999) to parse the sentences. The constituent parse trees were then transformed into dependency parse trees, using the head of each constituent (Jiang and Zhai, 2007b). The candidate relation instances were generated by considering all pairs of entities that occur in the same sentence. For the similarity matrix W in section 4.1 and the kernel $K(\cdot, \cdot)$ in section 4.2, we used the composite kernel function (Zhang et al., 2006), which is based on structured features and entity-related features.

F_1 is used to measure the performance of the algorithms. This is the harmonic mean of precision $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and recall $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, where TP, FP and FN are the numbers of correct, missing and wrongly recognized relations.

5.1 Experimental Settings

For ACE 2004, we used each of the six domains as the target domain and the remaining domains as source domains. For YAGO, each relation type in YAGO was considered as a domain. For each domain in YAGO, we have a binary classification task: whether the instance has the relation corresponding to the domain. Five-fold cross-validation was used to evaluate the performance.

[2] For two-classes, though, only one function is needed.

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

For every target domain, we divided all data into 5 subsets, and we used each subset for testing and the other four subsets for training. In the training set, we randomly removed most of the positive instances of the target domain from the training set except for 10% of the labeled data. This gave us the *weakly-supervised* setting. This was repeated five times with different training and test sets. We report the average performance over the five runs.

In our experiments, we set $\mu = 0.8$ in Eq. 1; $\theta = 0.18$ in Eq. 2; and $\gamma = 0.1$ and $\beta = 0.3$ in Eq. 3. For each target domain, we used $k \in \{1,3,5\}$ different source domains chosen randomly from the remaining domains. Thus, the relevance of the source domains to the target domain varies from experiment to experiment.

5.2 Baselines

We compare our framework with several other methods, including state-of-the-art machine learning, relation extraction and common domain adaptation methods. These are described below.

In-domain multiclass classifier This is Support-vector-machine (Fan et al., 2008, SVM) using the one-versus-rest decoding without removing positive labeled data (Jiang and Zhai, 2007b) from the target domain. Its performance can be regarded as an upper bound on the performance of the cross-domain methods.

No-transfer classifier (NT) We only use the few labeled instances of the target relation type together with the negative relation instances to train a binary classifier.

Alternate no-transfer classifier (NT-U) We use the union of the k source-domain labeled data sets D^ss and the small set of target-domain labeled data D_l to train a binary classifier. It is then applied directly to predict on the unlabeled target-domain data D_u without any adaptation.

Laplacian SVM (L-SVM) This is a semi-supervised learning method based on label propagation (Melacci and Belkin, 2011).

Multi-task transfer (MTL) This is a learning method for weakly-supervised relation extraction (Jiang, 2009).

Adaptive domain bootstrapping (DAB) This is an instance-based domain adaptation method for relation extraction (Xu et al., 2010).

Structural correspondence learning (SCL) We use the feature-based domain adaptation approach by Blitzer et al. (2007). We apply SCL on the D^ss and D_l to train a model. The learned model then makes predictions on D_u.

Domain Adaptation Machine (DAM) We use the framework of Duan et al. (2009), which is a multiple-sources domain adaptation method.

For the kernel-based methods above, we use the same composite kernel used in our method. The source codes of L-SVM, MTL, SCL and DAM were obtained from the authors. The others were re-implemented.

5.3 Experimental Results

Tables 2, 3 and 4 present the results on ACE 2004 (corresponding to $k = 1,3,5$), and Tables 5 present those on YAGO (corresponding to $k = 5$).

From Table 3 and Table 5, we see that the proposed method has the best F_1 among all the other methods, except for the supervised upper bound (In-domain). We first notice that NT-U generally does not perform well, and sometimes it performs worse than NT. The reason is that NT-U aims to obtain a consensus among the domains, and this will give a worse label than NT when there are enough irrelevant sources to influence the classification decision wrongly. In fact, one can roughly deduce that a target domain has few relevant source domains by simply comparing columns NT with columns NT-U in the tables: a decrease in F_1 from NT to NT-U suggests that the source domains are mainly irrelevant. For example, for domain BC in ACE 2004, we find that its F_1 decreases from NT to NT-U consistently in Tables 2, 3 and 4, which suggests that BN, NW, CTS, UN and WL are generally irrelevant to it; and similarly for domain CTS. We investigate this further by examining the relevance scores $w_{s,j}$s, and we find that the decreases in F_1 from NT to NT-U happen when there are more regions in the target domain to which source-domains are irrelevant.

We find that MTL, DAB and SCL are better than NT-U when the majority of source domains are relevant. This shows that MTL, DAB and SCL are able to make more effective use of relevant sources than NT-U. Howevever, we find that their perfomances are not stable: for example, MTL for target UN in Table 2. In contrast, we find the performance of L-SVM and DAM to be more stable. The reason is their reduced vulnerability to

Table 2: The F_1 of different methods on ACE 2004 with $k = 1$ source domain. The best performance for each target domain is in bold.

Target	In-domain	NT	NT-U	L-SVM	MTL	DAB	SCL	DAM	RDA
BC	55.74	30.00	20.31	32.42	32.74	32.12	30.41	33.07	**35.43**
BN	67.24	33.43	38.31	35.40	44.81	27.32	45.27	43.26	**47.28**
NW	68.32	41.48	39.35	41.50	42.28	43.27	44.16	41.69	**45.41**
CTS	72.92	36.60	29.90	36.15	**45.06**	37.50	44.68	39.40	44.27
UN	45.16	21.67	17.55	25.10	18.69	18.78	28.77	26.57	**31.07**
WL	46.46	28.53	23.84	29.90	26.13	24.78	23.71	27.01	**30.80**
Average	57.58	31.95	28.21	33.41	35.02	30.46	29.57	33.50	**39.00**

Table 3: The F_1 of different methods on ACE 2004 with $k = 3$ source domains.

Target	In-domain	NT	NT-U	L-SVM	MTL	DAB	SCL	DAM	RDA
BC	55.74	30.00	24.55	32.42	35.26	34.12	37.83	36.08	**39.43**
BN	67.24	33.43	38.31	35.40	49.76	32.15	49.25	45.89	**51.28**
NW	68.32	41.48	43.35	42.50	43.28	43.71	44.16	44.01	**46.41**
CTS	72.92	36.60	30.25	36.15	45.06	37.50	44.68	42.51	**49.27**
UN	45.16	21.67	27.55	25.10	19.72	**35.78**	31.77	33.29	35.07
WL	46.46	28.53	30.72	30.90	33.21	32.81	26.37	32.46	**35.11**
Average	57.58	31.95	32.46	34.20	37.72	36.01	39.01	39.10	**42.76**

Table 4: The F_1 of different methods on ACE 2004 with $k = 5$ source domains.

Target	In-domain	NT	NT-U	L-SVM	MTL	DAB	SCL	DAM	RDA
BC	55.74	30.00	27.32	33.07	37.76	35.08	40.38	38.70	**42.90**
BN	67.24	33.43	40.83	36.42	52.69	42.76.	50.47	48.23	**53.40**
NW	68.32	41.48	44.35	43.69	47.80	44.09	45.50	46.06	**49.13**
CTS	72.92	36.60	34.60	38.90	45.06	38.71	47.35	45.69	**52.63**
UN	45.16	21.67	29.34	26.34	35.47	35.44	33.21	34.13	**36.02**
WL	46.46	28.53	32.41	31.56	34.72	32.81	36.89	32.29	**37.90**
Average	57.58	31.95	34.80	35.0	42.25	38.15	42.30	40.84	**45.33**

negative transfer from irrelevant sources by relying on similarity of feature vectors between source and target domains based on labeled and unlabeled data. Further improvements can still be made, as shown by the better performance of RDA over L-SVM and DAM. This is achieved by further adjusting the relevances between source and target domains according to regions in the target-domain input space.

We analyzed histogram of the relation types to order the domains according to the imbalance of the class distributions. Using this, we observe that MTL, DAB and SCL perform relatively badly when the target-domain distribution is more imbalanced. In constrast, L-SVM, DAM and RDA are more robust.

Comparing with the baselines, RDA achieves the best performance on almost all the experiments. Using the two-phase framework, RDA can successfully transfer useful knowledge even in the pressence of irrelevant sources and imbalanced distributions. For ACE 2004, the improvement in F_1 over the best baseline can be up to 4.0% and is on average 3.6%. Similarly for YAGO, the improvement in F_1 over the best baseline can be up to 5.5% and is on average 4.3%.

Impact of Number of Source Domains Tables 2, 3, 4 and 6 also demonstrate that RDA improves monotonically as the number of source domains increases for both ACE 2004 and YAGO.

Table 5: The F_1 of different methods on YAGO with $k = 5$ source domains.

Target	In-domain	NT	NT-U	L-SVM	MTL	DAB	SCL	DAM	RDA
acquirer_acquiree	58.74	32.12	33.19	43.16	45.28	39.08	44.19	45.07	**51.15**
actedIn	77.36	40.73	44.32	50.45	57.18	49.61	58.23	56.37	**63.40**
bornIn	68.32	42.39	40.35	44.38	49.80	48.36	50.67	48.12	**56.93**
ceo_company	82.92	47.60	51.27	55.27	61.06	58.33	57.41	59.08	**66.71**
company_headquarters	75.16	48.92	52.15	50.13	59.47	61.23	58.36	56.65	**64.36**
created	74.26	46.37	43.58	60.45	60.74	55.08	59.42	57.34	**65.28**
diedIn	81.45	42.78	47.37	57.37	62.69	57.16	65.28	60.44	**71.15**
directed	70.11	44.42	48.29	50.57	54.29	49.09	52.31	50.30	**57.71**
discovered	68.13	37.34	42.51	48.77	53.04	49.82	53.73	51.21	**59.12**
graduatedFrom	69.37	39.28	45.74	51.56	58.22	54.38	56.32	51.17	**60.37**
hasChild	74.56	49.14	50.98	56.07	64.82	53.41	62.38	61.12	**66.83**
hasWonPrize	69.41	38.75	45.72	53.47	57.38	52.76	58.29	54.03	**63.13**
isLeaderOf	79.18	46.31	52.66	58.88	63.49	60.27	63.75	61.51	**70.27**
isMarriedTo	73.33	47.85	48.16	52.31	56.39	50.73	55.35	52.10	**62.58**
livesIn	66.93	36.16	35.15	40.28	50.27	41.72	43.59	48.11	**56.91**
participatedIn	85.38	46.22	48.33	62.48	67.51	61.08	65.38	61.12	**71.72**
person_birthplace	77.62	43.43	45.27	49.66	58.47	59.32	57.55	52.14	**65.80**
person_field	68.32	36.25	37.93	47.69	54.22	50.46	50.47	48.89	**59.47**
politicianOf	79.10	39.17	42.25	53.38	64.56	62.11	60.74	58.82	**68.12**
worksAt	84.29	45.78	49.78	59.34	65.33	65.44	66.53	63.24	**73.31**
Average	74.20	42.55	45.25	52.28	58.21	53.97	56.80	54.84	**63.72**

Performance Gap From Tables 2 to 4, we observe that the smallest performance gap between RDA and the in-domain settings is still high (about 12% with $k = 5$) on ACE 2004. This is because we have used a lot less labeled instances in the target domains: only 10% are used. However, the gaps reduces when the number of source domains increases. Comparing with the in-domain results in Table 5 (which is constant with k), Table 6 also shows a similar trend on YAGO. By exploiting the labeled data in ten source domains in YAGO, our RDA algorithm can reduce the gap between the cross-domain and in-domain settings to 9%.

Table 6: Average F_1 of RDA on YAGO

# source domains	F_1
$k = 1$	53.81
$k = 3$	59.43
$k = 5$	63.72
$k = 10$	65.55

6 Conclusion and Future Work

In this paper, we have proposed a robust domain adaptation (RDA) approach for the relation extraction problem where labeled data is scarce. Existing domain adaptation approaches suffer from negative transfer and under imbalanced distributions. To overcome these, we have proposed a two-phase approach to transfer only relevant information from multiple source domains, and thus derive accurate and robust predictions on the unlabeled target-domain data. Experimental results on ACE 2004 and YAGO have shown that the our domain adaptation method achieves the best performance on F_1 measure compared with the other baselines when only few labeled target instances are used. Because of the practical importance of domain adaptation for relation extraction due to lack of labeled data in new domains, we hope our study and findings will lead to further investigations into this problem.

Acknowledgments

This work is supported by DSO grant DSOCL10021. We thank Jiang for providing the source code for feature extraction and Bollegala for sharing his YAGO dataset.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Michele Banko, Oren Etzioni, and Turing Center. 2008. The tradeoffs between open and traditional relation extraction. *Proceedings of ACL-08: HLT*, pages 28–36.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. ACL.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011. Relation adaptation: learning to extract novel relations with minimum supervision. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2205–2210. AAAI Press.

Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. *Advances in neural information processing systems*, 18:171–178.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 423–429. ACL.

Hal Daume and D Marcu. 2007. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, pages 256–263.

Anhai Doan, Pedro Domingos, and Alon Halevy. 2003. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50(3):279–301.

Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual ICML*, pages 289–296. ACM.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Jing Jiang and ChengXiang Zhai. 2007a. Instance weighting for domain adaptation in nlp. In *Annual Meeting-Association For Computational Linguistics*, pages 264–271.

Jing Jiang and ChengXiang Zhai. 2007b. A systematic exploration of the feature space for relation extraction. In *HLT-NAACL*, pages 113–120.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the 47th Annual Meeting of the ACL: Volume 2-Volume 2*, pages 1012–1020.

Zornitsa Kozareva and Eduard Hovy. 2010. Not all seeds are equal: Measuring the quality of text mining seeds. In *HLT: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 618–626. ACL.

Stefano Melacci and Mikhail Belkin. 2011. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of EMNLP-13*, volume 500, pages 447–457.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of NAACL-HLT*, pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts - step one: The one-million fact extraction challenge. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1400–1405. AAAI Press.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1498–1507.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd Conference on Computational Linguistics*, pages 697–704. ACL.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on AI*, pages 474–479.

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898, pages –.

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the HLT Conference of the North American Chapter of the ACL*, pages 304–311.

Alex J Smola and Bernhard Scholkopf. 1998. *Learning with kernels*. Citeseer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics.

Feiyu Xu, Hans Uszkoreit, Sebastian Krause, and Hong Li. 2010. Boosting relation extraction with limited closed-world knowledge. In *Proceedings of the 23rd Conference on Computational Linguistics*, pages 1354–1362. ACL.

Wei Xu, Raphael Hoffmann Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of EMNLP-13*, pages 665–670.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *NIPS*.

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World wide web*, pages 101–110. ACM.