Multi-Task Transfer Learning for Weakly-Supervised Relation Extraction

Jing Jiang
School of Information Systems
Singapore Management University
80 Stamford Road, Singapore 178902
jingjiang@smu.edu.sg

Abstract

Creating labeled training data for relation extraction is expensive. In this paper, we study relation extraction in a special weakly-supervised setting when we have only a few seed instances of the target relation type we want to extract but we also have a large amount of labeled instances of other relation types. Observing that different relation types can share certain common structures, we propose to use a multi-task learning method coupled with human guidance to address this weakly-supervised relation extraction problem. The proposed framework models the commonality among different relation types through a shared weight vector, enables knowledge learned from the auxiliary relation types to be transferred to the target relation type, and allows easy control of the tradeoff between precision and recall. Empirical evaluation on the ACE 2004 data set shows that the proposed method substantially improves over two baseline methods.

1 Introduction

Relation extraction is the task of detecting and characterizing semantic relations between entities from free text. Recent work on relation extraction has shown that supervised machine learning coupled with intelligent feature engineering or kernel design provides state-of-the-art solutions to the problem (Culotta and Sorensen, 2004; Zhou et al., 2005; Bunescu and Mooney, 2005; Qian et al., 2008). However, supervised learning heavily relies on a sufficient amount of labeled data for training, which is not always available in practice due to the labor-intensive nature of human annotation. This problem is especially serious for relation extraction because the types of relations to be extracted are highly dependent on the application domain. For example, when working in the financial domain we may be interested in the *employment* relation, but when moving to the terrorism domain we now may be interested in the *ethnic and ideology affiliation* relation, and thus have to create training data for the new relation type.

However, is the old training data really useless? Inspired by recent work on transfer learning and domain adaptation, in this paper, we study how we can leverage labeled data of some old relation types to help the extraction of a new relation type in a weakly-supervised setting, where only a few seed instances of the new relation type are available. While transfer learning was proposed more than a decade ago (Thrun, 1996; Caruana, 1997), its application in natural language processing is still a relatively new territory (Blitzer et al., 2006; Daume III, 2007; Jiang and Zhai, 2007a; Arnold et al., 2008; Dredze and Crammer, 2008), and its application in relation extraction is still unexplored.

Our idea of performing transfer learning is motivated by the observation that different relation types share certain common syntactic structures, which can possibly be transferred from the old types to the new type. We therefore propose to use a general multi-task learning framework in which classification models for a number of related tasks are forced to share a common model component and trained together. By treating classification of different relation types as related tasks, the learning framework can naturally model the common syntactic structures among different relation types in a principled manner. It also allows us to introduce human guidance in separating the common model component from the type-specific components. The framework naturally transfers the knowledge learned from the old relation types to the new relation type and helps improve the recall of the relation extractor. We also exploit ad-

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1012–1020,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

ditional human knowledge about the entity type constraints on the relation arguments, which can usually be derived from the definition of a relation type. Imposing these constraints further improves the precision of the final relation extractor. Empirical evaluation on the ACE 2004 data set shows that our proposed method largely outperforms two baseline methods, improving the average F1 measure from 0.1532 to 0.4132 when only 10 seed instances of the new relation type are used.

2 Related work

Recent work on relation extraction has been dominated by feature-based and kernel-based supervised learning methods. Zhou et al. (2005) and Zhao and Grishman (2005) studied various features and feature combinations for relation extraction. We systematically explored the feature space for relation extraction (Jiang and Zhai, 2007b) . Kernel methods allow a large set of features to be used without being explicitly extracted. A number of relation extraction kernels have been proposed, including dependency tree kernels (Culotta and Sorensen, 2004), shortest dependency path kernels (Bunescu and Mooney, 2005) and more recently convolution tree kernels (Zhang et al., 2006; Qian et al., 2008). However, in both feature-based and kernel-based studies, availability of sufficient labeled training data is always assumed.

Chen et al. (2006) explored semi-supervised learning for relation extraction using label propagation, which makes use of *unlabeled* data. Zhou et al. (2008) proposed a hierarchical learning strategy to address the data sparseness problem in relation extraction. They also considered the commonality among different relation types, but compared with our work, they had a different problem setting and a different way of modeling the commonality. Banko and Etzioni (2008) studied open domain relation extraction, for which they manually identified several common relation patterns. In contrast, our method obtains common patterns through statistical learning. Xu et al. (2008) studied the problem of adapting a rule-based relation extraction system to new domains, but the types of relations to be extracted remain the same.

Transfer learning aims at transferring knowledge learned from one or a number of old tasks to a new task. Domain adaptation is a special case of transfer learning where the learning task remains the same but the distribution of data changes. There has been an increasing amount of work on transfer learning and domain adaptation in natural language processing recently. Blitzer et al. (2006) proposed a structural correspondence learning method for domain adaptation and applied it to part-of-speech tagging. Daume III (2007) proposed a simple feature augmentation method to achieve domain adaptation. Arnold et al. (2008) used a hierarchical prior structure to help transfer learning and domain adaptation for named entity recognition. Dredze and Crammer (2008) proposed an online method for multi-domain learning and adaptation.

Multi-task learning is another learning paradigm in which multiple related tasks are learned simultaneously in order to achieve better performance for each individual task (Caruana, 1997; Evgeniou and Pontil, 2004). Although it was not originally proposed to transfer knowledge to a particular new task, it can be naturally used to achieve this goal because it models the commonality among tasks, which is the knowledge that should be transferred to a new task. In our work, transfer learning is done through a multi-task learning framework similar to Evgeniou and Pontil (2004).

3 Task definition

Our study is conducted using data from the Automatic Content Extraction (ACE) program[1]. We focus on extracting binary relation instances between two relation arguments occurring in the same sentence. Some example relation instances and their corresponding relation types as defined by ACE can be found in Table 1.

We consider the following weakly-supervised problem setting. We are interested in extracting instances of a *target* relation type $\mathcal{T}$, but this relation type is only specified by a small set of seed instances. We may possibly have some additional knowledge about the target type not in the form of labeled instances. For example, we may be given the entity type restrictions on the two relation arguments. In addition to such limited information about the target relation type, we also have a large amount of labeled instances for K *auxiliary* relation types $\mathcal{A}_1, \ldots, \mathcal{A}_K$. Our goal is to learn a relation extractor for $\mathcal{T}$, leveraging all the data and information we have.

[1] http://projects.ldc.upenn.edu/ace/

Syntactic Pattern	Relation Instance	Relation Type (Subtype)
arg-2 *arg-1*	**Arab** *leaders*	OTHER-AFF (Ethnic)
	his *father*	PER-SOC (Family)
	South **Jakarta** Prosecution *Office*	GPE-AFF (Based-In)
arg-1 of **arg-2**	*leader* of a minority **government**	EMP-ORG (Employ-Executive)
	the youngest *son* of ex-director **Suharto**	PER-SOC (Family)
	the *Socialist People's Party* of **Montenegro**	GPE-AFF (Based-In)
arg-1 [verb] **arg-2**	*Yemen* [sent] **planes** to Baghdad	ART (User-or-Owner)
	his *wife* [had] three young **children**	PER-SOC (Family)
	Jody Scheckter [paced] **Ferrari** to both victories	EMP-ORG (Employ-Staff)

Table 1: Examples of similar syntactic structures across different relation types. The head words of the first and the second arguments are shown in italic and bold, respectively.

Before introducing our transfer learning solution, let us first briefly explain our basic classification approach and the features we use, as well as two baseline solutions.

3.1 Feature configuration

We treat relation extraction as a classification problem. Each pair of entities within a single sentence is considered a candidate relation instance, and the task becomes predicting whether or not each candidate is a true instance of $\mathcal{T}$. We use feature-based logistic regression classifiers. Following our previous work (Jiang and Zhai, 2007b), we extract features from a sequence representation and a parse tree representation of each relation instance. Each node in the sequence or the parse tree is augmented by an *argument tag* that indicates whether the node subsumes *arg-1*, *arg-2*, both or neither. Nodes that represent the arguments are also labeled with the entity type, subtype and mention type as defined by ACE. Based on the findings of Qian et al. (2008), we trim the parse tree of a relation instance so that it contains only the most essential components. We extract unigram features (consisting of a single node) and bigram features (consisting of two connected nodes) from the graphic representations. An example of the graphic representation of a relation instance is shown in Figure 1 and some features extracted from this instance are shown in Table 2. This feature configuration gives state-of-the-art performance (F1 = 0.7223) on the ACE 2004 data set in a standard setting with sufficient data for training.

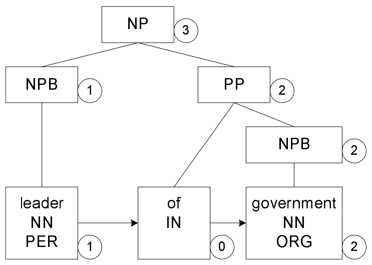

Figure 1: The combined sequence and parse tree representation of the relation instance "*leader of a minority government*." The non-essential nodes for "*a*" and for "*minority*" are removed based on the algorithm from Qian et al. (2008).

Feature	Explanation
ORG^2	*arg-2* is an *ORG* entity.
of^0 government^2	*arg-2* is "government" and follows the word "of."
$\text{NP}^3 \rightarrow \text{PP}^2$	There is a noun phrase containing both arguments, with *arg-2* contained in a prepositional phrase inside the noun phrase.

Table 2: Examples of unigram and bigram features extracted from Figure 1.

3.2 Baseline solutions

We consider two baseline solutions to the weakly-supervised relation extraction problem. In the first baseline, we use only the few seed instances of the target relation type together with labeled *negative* relation instances (i.e. pairs of entities within the same sentence but having no relation) to train a binary classifier. In the second baseline, we take the union of the positive instances of both the target relation type and the auxiliary relation types as our positive training set, and together with the negative instances we train a binary classifier. Note that the second baseline method essentially learns a classifier for *any* relation type.

Another existing solution to weakly-supervised learning problems is semi-supervised learning, e.g. bootstrapping. However, because our proposed transfer learning method can be combined with semi-supervised learning, here we do not include semi-supervised learning as a baseline.

4 A multi-task transfer learning solution

We now present a multi-task transfer learning solution to the weakly-supervised relation extraction problem, which makes use of the labeled data from the auxiliary relation types.

4.1 Syntactic similarity between relation types

To see why the auxiliary relation types may help the identification of the target relation type, let us first look at how different relation types may be related and even similar to each other. Based on our inspection of a sample of the ACE data, we find that instances of different relation types can share certain common syntactic structures. For example, the syntactic pattern "*arg-1* of *arg-2*" strongly indicates that there exists some relation between the two arguments, although the nature of the relation may be well dependent on the semantic meanings of the two arguments. More examples are shown in Table 1. This observation suggests that some of the syntactic patterns learned from the auxiliary relation types may be transferable to the target relation type, making it easier to learn the target relation type and thus alleviating the insufficient training data problem with the target type.

How can we incorporate this desired knowledge transfer process into our learning method? While one can make explicit use of these general syntactic patterns in a rule-based relation extraction system, here we restrict our attention to feature-based linear classifiers. We note that in feature-based linear classifiers, a useful syntactic pattern is translated into large weights for features related to the syntactic pattern. For example, if "*arg-1* of *arg-2*" is a useful pattern, in the learned linear classifier we should have relatively large weights for features such as "the word *of* occurs before *arg-2*" or "a preposition occurs before *arg-2*," or even more complex features such as "there is a prepositional phrase containing *arg-2* attached to *arg-1*." It is the weights of these generally useful features that are transferable from the auxiliary relation types to the target relation type.

4.2 Statistical learning model

As we have discussed, we want to force the linear classifiers for different relation types to share their model weights for those features that are related to the common syntactic patterns. Formally, we consider the following statistical learning model.

Let ω^k denote the weight vector of the linear classifier that separates positive instances of auxiliary type $\mathcal{A}_k$ from negative instances, and let $\omega^{\mathcal{T}}$ denote a similar weight vector for the target type $\mathcal{T}$. If different relation types are totally unrelated, these weight vectors should also be independent of each other. But because we observe similar syntactic structures across different relation types, we now assume that these weight vectors are related through a common component ν:

$$\begin{aligned} \omega^{\mathcal{T}} &= \mu^{\mathcal{T}} + \nu, \\ \omega^k &= \mu^k + \nu \quad \text{for} \quad k = 1, \ldots, K. \end{aligned}$$

If we assume that only weights of certain general features can be shared between different relation types, we can force certain dimensions of ν to be 0. We express this constraint by introducing a matrix F and setting $F\nu = 0$. Here F is a square matrix with all entries set to 0 except that $F_{i,i} = 1$ if we want to force $\nu_i = 0$.

Now we can learn these weight vectors in a multi-task learning framework. Let x represent the feature vector of a candidate relation instance, and $y \in \{+1, -1\}$ represent a class label. Let $\mathcal{D}_{\mathcal{T}} = \{(x_i^{\mathcal{T}}, y_i^{\mathcal{T}})\}_{i=1}^{N_{\mathcal{T}}}$ denote the set of labeled instances for the target type $\mathcal{T}$. (Note that the number of *positive* instances in $\mathcal{D}_{\mathcal{T}}$ is very small.) And let $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ denote the labeled instances for the auxiliary type $\mathcal{A}_k$.

We learn the optimal weight vectors $\{\hat{\mu}^k\}_{k=1}^K$, $\hat{\mu}^{\mathcal{T}}$ and $\hat{\nu}$ by optimizing the following objective function:

$$\begin{aligned} &\left(\{\hat{\mu}^k\}_{k=1}^K, \hat{\mu}^{\mathcal{T}}, \hat{\nu}\right) \\ = &\mathop{\arg\min}_{\{\mu^k\}, \mu^{\mathcal{T}}, \nu, F\nu=0} \Bigg[L(\mathcal{D}_{\mathcal{T}}, \mu^{\mathcal{T}} + \nu) \\ &+ \sum_{k=1}^K L(\mathcal{D}_k, \mu^k + \nu) \\ &+ \lambda_\mu^{\mathcal{T}} \|\mu^{\mathcal{T}}\|^2 + \sum_{k=1}^K \lambda_\mu^k \|\mu^k\|^2 + \lambda_\nu \|\nu\|^2 \Bigg]. \quad (1) \end{aligned}$$

The objective function follows standard empirical risk minimization with regularization. Here $L(\mathcal{D}, \omega)$ is the aggregated loss of labeling x with y for all (x, y) in $\mathcal{D}$, using weight vector ω. In logistic regression models, the loss function is the negative log likelihood, that is,

$$\begin{aligned} L(\mathcal{D}, \omega) &= -\sum_{(x,y)\in\mathcal{D}} \log p(y|x, \omega), \\ p(y|x, \omega) &= \frac{\exp(\omega_y \cdot x)}{\sum_{y' \in \{+1,-1\}} \exp(\omega_{y'} \cdot x)}. \end{aligned}$$

$\lambda_\mu^{\mathcal{T}}$, λ_μ^k and λ_ν are regularization parameters. By adjusting their values, we can control the degree of weight sharing among the relation types. The larger the ratio $\lambda_\mu^{\mathcal{T}}/\lambda_\nu$ (or $\lambda_\mu^k/\lambda_\nu$) is, the more we believe that the model for $\mathcal{T}$ (or $\mathcal{A}_k$) should conform to the common model, and the smaller the type-specific weight vector $\mu^{\mathcal{T}}$ (or μ^k) will be.

The model presented above is based on our previous work (Jiang and Zhai, 2007c), which bears the same spirit of some other recent work on multitask learning (Ando and Zhang, 2005; Evgeniou and Pontil, 2004; Daume III, 2007). It is general for any transfer learning problem with auxiliary labeled data from similar tasks. Here we are mostly interested in the model's applicability and effectiveness on the relation extraction problem.

4.3 Feature separation

Recall that we impose a constraint $F\nu = 0$ when optimizing the objective function. This constraint gives us the freedom to force only the weights of a subset of the features to be shared among different relation types. A remaining question is how to set this matrix F, that is, how to determine the set of general features to use. We propose two ways of setting this matrix F.

Automatically setting F

One way is to fix the number of non-zero entries in ν to be a pre-defined number H of general features, and allow F to change during the optimization process. This can be done by repeating the following two steps until F converges:

1. Fix F, and optimize the objective function as in Equation (1).

2. Fix $(\mu^{\mathcal{T}} + \nu)$ and $(\mu^k + \nu)$, and search for $\mu^{\mathcal{T}}$, $\{\mu^k\}$ and ν that minimizes $\left(\lambda_\mu^{\mathcal{T}} \|\mu^{\mathcal{T}}\|^2 + \sum_{k=1}^K \lambda_\mu^k \|\mu^k\|^2 + \lambda_\nu \|\nu\|^2\right)$, subject to the constraint that at most H entries of ν are nonzero.

Human guidance

Another way to select the general features is to follow some guidance from human knowledge. Recall that in Section 4.1 we find that the commonality among different relation types usually lies in the syntactic structures between the two arguments. This observation gives some intuition about how to separate general features from type-specific features. In particular, here we consider two hypotheses regarding the generality of different kinds of features.

Argument word features: We hypothesize that the head words of the relation arguments are more likely to be strong indicators of *specific* relation types rather than *any* relation type. For example, if an argument has the head word "*sister*," it strongly indicates a *family* relation. We refer to the set of features that contain any head word of an argument as "**arg-word**" features.

Entity type features: We hypothesize that the entity types and subtypes of the relation arguments are also more likely to be associated with *specific* relation types. For example, arguments that are *location* entities may be strongly correlated with *physical proximity* relations. We refer to the set of features that contain the entity type or subtype of an argument as "**arg-NE**" features.

We hypothesize that the **arg-word** and **arg-NE** features are type-specific and therefore should be excluded from the set of general features. We can force the weights of these hypothesized type-specific features to be 0 in the shared weight vector ν, i.e. we can set the matrix F to achieve this feature separation.

Combined method

We can also combine the automatic way of setting F with human guidance. Specifically, we still follow the first automatic procedure to choose general features, but we then filter out any hypothesized type-specific feature from the set of general features chosen by the automatic procedure.

4.4 Imposing entity type constraints

Finally, we consider how we can exploit additional human knowledge about the target relation type $\mathcal{T}$ to further improve the classifier. We note that usually when a relation type is defined, we often have strong preferences or even hard constraints on the types of entities that can possibly be the two relation arguments. These type constraints can help us

Target Type $\mathcal{T}$		BL	BL-$\mathcal{A}$	TL-auto	TL-guide	TL-comb	TL-NE
Physical	P	0.0000	0.1692	0.2920	0.2934	0.3325	0.5056
	R	0.0000	0.0848	0.1696	0.1722	0.2383	0.2316
	F	0.0000	0.1130	0.2146	0.2170	0.2777	**0.3176**
Personal /Social	P	1.0000	0.0804	0.1005	0.3069	0.3214	0.6412
	R	0.0386	0.1708	0.1598	0.7245	0.7686	0.7631
	F	0.0743	0.1093	0.1234	0.4311	0.4533	**0.6969**
Employment /Membership /Subsidiary	P	0.9231	0.3561	0.5230	0.5428	0.5973	0.7145
	R	0.0075	0.1850	0.2617	0.2648	0.3632	0.3601
	F	0.0148	0.2435	0.3488	0.3559	0.4518	**0.4789**
Agent-Artifact	P	0.8750	0.0603	0.1813	0.1825	0.1835	0.1967
	R	0.0343	0.2353	0.6471	0.6225	0.6422	0.6373
	F	0.0660	0.0960	0.2833	0.2822	0.2854	**0.3006**
PER/ORG Affiliation	P	0.8889	0.0838	0.1510	0.1592	0.1667	0.1844
	R	0.0567	0.4965	0.6950	0.8369	0.8794	0.8723
	F	0.1067	0.1434	0.2481	0.2676	0.2802	**0.3045**
GPE Affiliation	P	1.0000	0.2530	0.3904	0.3604	0.3560	0.5824
	R	0.0077	0.4509	0.6416	0.5992	0.6166	0.6127
	F	0.0153	0.3241	0.4854	0.4501	0.4513	**0.5972**
Discourse	P	1.0000	0.0298	0.0503	0.0471	0.1370	0.1370
	R	0.0036	0.0789	0.1075	0.1147	0.3477	0.3477
	F	0.0071	0.0433	0.0685	0.0668	0.1966	**0.1966**
Average	P	0.8124	0.1475	0.2412	0.2703	0.2992	0.4231
	R	0.0212	0.2432	0.3832	0.4764	0.5509	0.5464
	F	0.0406	0.1532	0.2532	0.2958	0.3423	**0.4132**

Table 3: Comparison of different methods on ACE 2004 data set. P, R and F stand for precision, recall and F1, respectively.

remove some false positive instances. We therefore manually identify the entity type constraints for each target relation type based on the definition of the relation type given in the ACE annotation guidelines, and impose these type constraints as a final refinement step on top of the predicted positive instances.

5 Experiments

5.1 Data set and experiment setup

We used the ACE 2004 data set to evaluate our proposed methods. There are seven relation types defined in ACE 2004. After data cleaning, we obtained 4290 positive instances among 48614 candidate relation instances. We took each relation type as the target type and used the remaining types as auxiliary types. This gave us seven sets of experiments. In each set of experiments for a single target relation type, we randomly divided all the data into five subsets, and used each subset for testing while using the other four subsets for training, i.e. each experiment was repeated five times with different training and test sets. Each time, we removed most of the positive instances of the target type from the training set except only a small number S of seed instances. This gave us the weakly-supervised setting. We kept all the positive instances of the target type in the test set. In order to concentrate on the classification accuracy for the target relation type, we removed the positive instances of the *auxiliary* relation types from the test set, although in practice we need to extract these auxiliary relation instances using learned classifiers for these relation types.

5.2 Comparison of different methods

We first show the comparison of our proposed multi-task transfer learning methods with the two baseline methods described in Section 3.2. The performance on each target relation type and the average performance across seven types are shown in Table 3. BL refers to the first baseline and BL-$\mathcal{A}$ refers to the second baseline which uses auxil-

$\lambda_\mu^\mathcal{T}$	100	1000	10000
P	0.6265	0.3162	0.2992
R	0.1170	0.3959	0.5509
F	0.1847	0.2983	0.3423

Table 4: The average performance of TL-comb with different $\lambda_\mu^\mathcal{T}$. ($\lambda_\mu^k = 10^4$ and $\lambda_\nu = 1$.)

iary relation instances. The four TL methods are all based on the multi-task transfer learning framework. TL-auto sets F automatically within the optimization problem itself. TL-guide chooses all features except **arg-word** and **arg-NE** features as general features and sets F accordingly. TL-comb combines TL-auto and TL-guide, as described in Section 4.3. Finally, TL-NE builds on top of TL-comb and uses the entity type constraints to refine the predictions. In this set of experiments, the number of seed instances for each target relation type was set to 10. The parameters were set to their optimal values ($\lambda_\mu^\mathcal{T} = 10^4$, $\lambda_\mu^k = 10^4$, $\lambda_\nu = 1$, and $H = 500$).

As we can see from the table, first of all, BL generally has high precision but very low recall. BL-$\mathcal{A}$ performs better than BL in terms of F1 because it gives better recall. However, BL-$\mathcal{A}$ still cannot achieve as high recall as the TL methods. This is probably because the model learned by BL-$\mathcal{A}$ still focuses more on type-specific features for each relation type rather than on the commonly useful general features, and therefore does not help much in classifying the target relation type.

The four TL methods all outperform the two baseline methods. TL-comb performs better than both TL-auto and TL-guide, which shows that while we can either choose general features automatically by the learning algorithm or manually with human knowledge, it is more effective to combine human knowledge with the multi-task learning framework. Not surprisingly, TL-NE improves the precision over TL-comb without hurting the recall much. Ideally, TL-NE should not decrease recall if the type constraints are strictly observed in the data. We find that it is not always the case with the ACE data, leading to the small decrease of recall from TL-comb to TL-NE.

5.3 The effect of $\lambda_\mu^\mathcal{T}$

Let us now take a look at the effect of using different $\lambda_\mu^\mathcal{T}$. As we can see from Table 4, smaller $\lambda_\mu^\mathcal{T}$ gives higher precision while larger $\lambda_\mu^\mathcal{T}$ gives higher recall. These results make sense because the larger $\lambda_\mu^\mathcal{T}$ is, the more we penalize large weights of $\mu^\mathcal{T}$. As a result, the model for the target type is forced to conform to the shared model ν and prevented from overfitting the few seed target instances. $\lambda_\mu^\mathcal{T}$ is therefore a useful parameter to help us control the tradeoff between precision and recall for the target type.

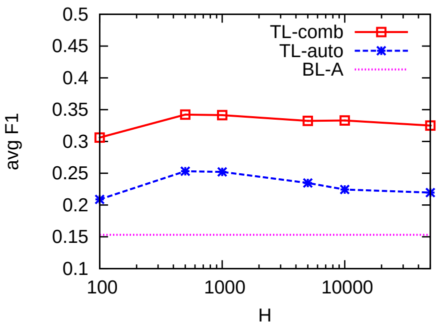

Figure 2: Performance of TL-comb and TL-auto as H changes.

While varying λ_μ^k also gives similar effect for type $\mathcal{A}_k$, we found that setting λ_μ^k to smaller values would not help $\mathcal{T}$ because in this case the auxiliary relation instances would be used more for training the type-specific component μ^k rather than the common component ν.

5.4 Sensitivity of H

Another parameter in the multi-task transfer learning framework is the number of general features H, i.e. the number of non-zero entries in the shared weight vector ν. To see how the performance may vary as H changes, we plot the performance of TL-comb and TL-auto in terms of the average F1 across the seven target relation types, with H ranging from 100 to 50000. As we can see in Figure 2, the performance is relatively stable, and always above BL-$\mathcal{A}$. This suggests that the performance of TL-comb and TL-auto is not very sensitive to the value of H.

5.5 Hypothesized type-specific features

In Section 4.3, we showed two sets of hypothesized type-specific features, namely, **arg-word** features and **arg-NE** features. We also experimented with each set separately to see whether both sets are useful. The comparison is shown in Table 5. As we can see, using either set of type-specific features in either TL-guide or TL-comb can improve the performance over BL-$\mathcal{A}$, but the

	arg-word	arg-NE	union
TL-guide	0.2095	0.2983	0.2958
TL-comb	0.2215	0.3331	0.3423
BL-$\mathcal{A}$	0.1532		

Table 5: Average F1 using different hypothesized type-specific features.

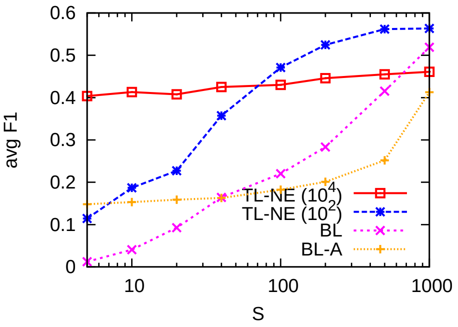

Figure 3: Performance of TL-NE, BL and BL-$\mathcal{A}$ as the number of seed instances S of the target type increases. ($H = 500$. $\lambda_{\mu}^{\mathcal{T}}$ was set to 10^4 and 10^2).

arg-NE features are probably more type-specific than **arg-word** features because they give better performance. Using the union of the two sets is still the best for TL-comb.

5.6 Changing the number of seed instances

Finally, we compare TL-NE with BL and BL-$\mathcal{A}$ when the number of seed instances increases. We set S from 5 up to 1000. When S is large, the problem becomes more like traditional supervised learning, and our setting of $\lambda_{\mu}^{\mathcal{T}} = 10^4$ is no longer optimal because we are now not afraid of overfitting the large set of seed target instances. Therefore we also included another TL-NE experiment with $\lambda_{\mu}^{\mathcal{T}}$ set to 10^2. The comparison of the performance is shown in Figure 3. We see that as S increases, both BL and BL-$\mathcal{A}$ catch up, and BL overtakes BL-$\mathcal{A}$ when S is sufficiently large because BL uses positive training examples only from the target type. Overall, TL-NE still outperforms the two baselines in most of the cases over the wide range of values of S, but the optimal value for $\lambda_{\mu}^{\mathcal{T}}$ decreases as S increases, as we have suspected. The results show that if $\lambda_{\mu}^{\mathcal{T}}$ is set appropriately, our multi-task transfer learning method is robust and advantageous over the baselines under both the weakly-supervised setting and the traditional supervised setting.

6 Conclusions and future work

In this paper, we applied multi-task transfer learning to solve a weakly-supervised relation extraction problem, leveraging both labeled instances of auxiliary relation types and human knowledge including hypotheses on feature generality and entity type constraints. In the multi-task learning framework that we introduced, different relation types are treated as different but related tasks that are learned together, with the common structures among the relation types modeled by a shared weight vector. The shared weight vector corresponds to the general features across different relation types. We proposed to choose the general features either automatically inside the learning algorithm or guided by human knowledge. We also leveraged additional human knowledge about the target relation type in the form of entity type constraints. Experiment results on the ACE 2004 data show that the multi-task transfer learning method achieves the best performance when we combine human guidance with automatic general feature selection, followed by imposing the entity type constraints. The final method substantially outperforms two baseline methods, improving the average F1 measure from 0.1532 to 0.4132 when only 10 seed target instances are used.

Our work is the first to explore transfer learning for relation extraction, and we have achieved very promising results. Because of the practical importance of transfer learning and adaptation for relation extraction due to lack of training data in new domains, we hope our study and findings will lead to further investigation into this problem. There are still many issues that remain unsolved. For example, we have not looked at the degrees of relatedness between different pairs of relation types. Presumably, when adapting to a specific target relation type, we want to choose the most similar auxiliary relation types to use. Our current study is based on ACE relation types. It would also be interesting to study similar problems in other domains, for example, the protein-protein interaction extraction problem in biomedical text mining.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, November.

Andrew Arnold, Ramesh Nallapati, and William W. Cohen. 2008. Exploiting feature hierarchy for transfer learning in named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 245–253.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 28–36.

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

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 724–731.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 129–136.

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

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 689–697.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117.

Jing Jiang and ChengXiang Zhai. 2007a. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 264–271.

Jing Jiang and ChengXiang Zhai. 2007b. A systematic exploration of the feature space for relation extraction. In *Proceedings of the Human Language Technologies Conference*, pages 113–120.

Jing Jiang and ChengXiang Zhai. 2007c. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 401–410.

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 International Conference on Computational Linguistics*, pages 697–704.

Sebastian Thrun. 1996. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems 8*, pages 640–646.

Feiyu Xu, Hans Uszkoreit, Hong Li, and Niko Felger. 2008. Adaptation of relation extraction rules to new domains. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 2446–2450.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference*, pages 288–295.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 419–426.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 427–434.

GuoDong Zhou, Min Zhang, DongHong Ji, and QiaoMing Zhu. 2008. Hierarchical learning strategy in semantic relation extraction. *Information Processing and Management*, 44(3):1008–1021.